University of Wollongong
Research Online

Faculty of Commerce - Papers (Archive)

Faculty of Business

2001

Using OLAP and multidimensional data for decision making

Helen M. Hasan
University of Wollongong, hasan@uow.edu.au

P. Hyland
University of Wollongong, phyland@uow.edu.au

Publication Details

This article was published as: Hasan, H and Hyland, P, Using OLAP and multidimensional data for decision making, IT Professional, September-October 2001, 3(5), 44-50. Copyright IEEE 2001.

Research Online is the open access institutional repository for the University of Wollongong. For further information contact the UOW Library: research-pubs@uow.edu.au

Using OLAP and multidimensional data for decision making

Abstract
Managers see information as a critical resource and require systems that let them exploit it for competitive advantage. One way to better use organizational information is via online analytical processing and multidimensional databases (MDDBs). OLAP and MDDBs present summarized information from company databases. They use multidimensional structures that let managers slice and dice views of company performance data and drill down into trouble spots. For over a decade, proponents have touted these tools as the ultimate executive information system, but most of the hype comes from product vendors themselves. Based on our experience with several OLAP tools, we have developed a more pragmatic approach to the design of multidimensional information systems that lets managers make the most of their companies' information assets.

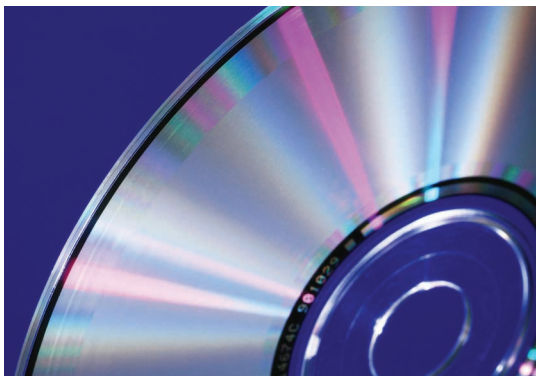
Disciplines
Business | Social and Behavioral Sciences

Publication Details
This article was published as: Hasan, H and Hyland, P, Using OLAP and multidimensional data for decision making, IT Professional, September-October 2001, 3(5), 44-50. Copyright IEEE 2001.

This journal article is available at Research Online: http://ro.uow.edu.au/commpapers/4

When well constructed, multidimensional databases work with online analytical processing to provide critical information.

Helen Hasan and Peter Hyland



Using OLAP and Multidimensional Data for Decision Making

Managers see information as a critical resource and require systems that let them exploit it for competitive advantage. One way to better use organizational information is via online analytical processing and multidimensional databases. OLAP and MDDBs present summarized information from company databases. They use multidimensional structures that let managers slice and dice views of company performance data and drill down into trouble spots. For over a decade, proponents have touted these tools as the ultimate executive information system, but most of the hype comes from product vendors themselves. Based on our experience with several OLAP tools, we have developed a more pragmatic approach to the design of multidimensional information systems that lets managers make the most of their companies' information assets.

VIEW FROM THE TOP

Managers often have an ambivalent attitude toward information systems. On the one hand, the well-known productivity paradox suggests that there is no correlation between IT investment and an organization's performance. Despite this fact, organizations continue to spend a large proportion of their budgets on IT systems.

On the other hand, some managers see information systems as an essential part of the successful, modern business, improving the efficiency of business processes and supporting innovative ways of doing business. The increasing popularity of data warehousing suggests that many managers now see the huge volumes of data stored in organizational databases as a potential company asset. Used effectively, this data provides information for management decision making and strategic planning. Indeed, we believe that the productivity paradox does not imply that organizations are misdirecting their IT investment. Instead, the paradox just doesn't account for the value of the information assets in company databases.

In providing these OLAP- and MDDB-based systems, the challenge is to distinguish between systems suitable for online transaction processing (OLTP)—which emphasizes everyday business operations—and those suitable for OLAP, the form that supports managerial decision making. For example, managers rarely read canned reports—those typically produced by traditional management information systems—with much enthusiasm. They prefer an easy-to-use, interactive system for both intentional information seeking or for browsing to get a general sense of what is going on in and around their organization. Such a system is an ideal application of OLAP and MDDB. Understanding how managers gather and use information, and then building systems to support them, raises complex technical and nontechnical issues. The multidimensional structure of data and presentation of management information is a conceptually simple approach for which technical systems are now available. However, managers must know how to get the

1520-9202/01/$10.00 © 2001 IEEE

most from these systems, and IT professionals need workable methods for designing multidimensional databases to match management needs.

HOW DO MULTIDIMENSIONAL DATABASES PROVIDE OLAP?

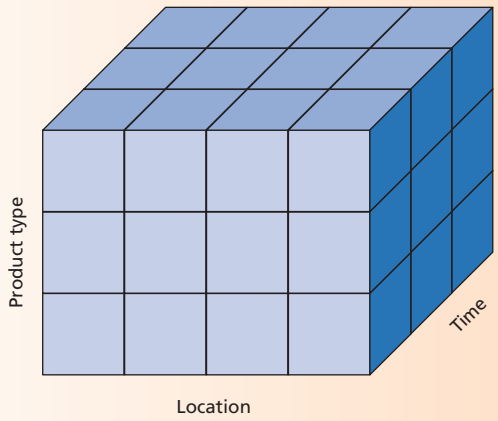
MDDB developers and users consider the dimensional view of organizational data to provide managers with a better means of understanding the current state and future possibilities of their business. MDDBs present data to users as a *hypercube* or multidimensional array, where each core data value (typically key performance indicators) occupies a cell indexed by a unique set of dimension values. In its simplest form you can easily visualize this structure using a value, such as number of products sold, and show how it varies along the three most common dimensions (time, location, and product type). Figure 1 illustrates this hypercube structure.

Hypercubes, however, are not limited to only one set of values or to three reference dimensions, as they can be extended to include any number of values and dimensions. This representation stands in sharp contrast to the set of tables used to represent data in the well-known relational database model, shown in Figure 2. Organizations have adopted relational database management systems (RDBMSs) for their OLTP applications when database tables—those that the information system records in two dimensions—make it quick and simple to record business transactions. Managers can extract information from the database via reports generated by the system or retrieve it ad hoc by using structured query language (SQL). However, the huge volume and fragmentation of normalized data require an incredible number of joins to satisfy even moderately complex queries that an executive might ask.

The data most commonly stored in an MDDB is an organization's historical performance figures. Organizations populate an MDDB with data from several organizational databases via a process that cleanses, summarizes, and processes the data for management consumption. The concept of a unified, logical, corporate data model was popular in the 1980s; in the 1990s, organizations physically implemented the concept as data warehouses. So MDDBs are often used in conjunction with a data warehouse or for data exploitation in an executive information system. Managers can manipulate the resulting MDDB hypercube of information by

- *slicing and dicing* the cube, which, for example, allows you to view sales by time, location, product type, or any combination of these dimensions;
- *drilling down* and *rolling up*, which, for example, allow you to view sales for a year, a quarter, a month or even a single day; and
- *filtering* the data to specify a single item of interest, such as a specific product type.

Figure 1. Hypercube that shows the three most common dimensions—time, location, and product type—for the data.



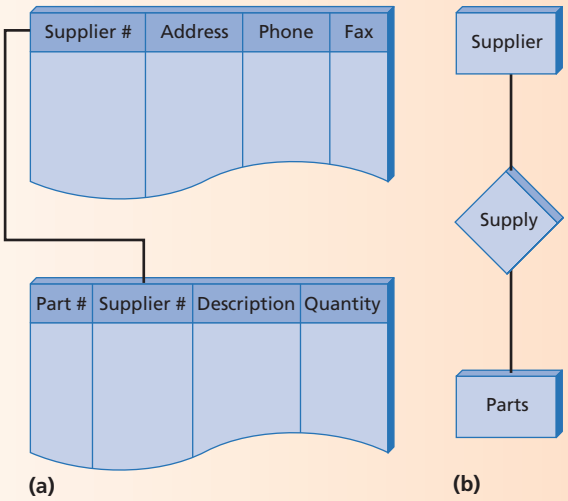
This information is available via an interactive GUI, and the literature uses the term "OLAP" to describe the set of tools that support these manipulations. OLAP supports business analysis at the executive and managerial levels, just as online OLTP automates business processes at the operational level.

Several variants of OLAP have evolved over the past few years, the most important being multidimensional OLAP (MOLAP) and relational OLAP (ROLAP). The main differences between the two technologies concern data storage processing capability and the currency of data.

The data in MOLAP systems is periodically uploaded from an organization's relational databases into a semipermanent hypercube. When uploading such data, it is necessary to perform multiple aggregations of the source data and to "clean" the data by eliminating duplicate or incomplete records that may exist in the source data. Doing so provides greater flexibility and better performance, achieving optimal performance and flexibility for the MDDB's users. One common criticism leveled at MOLAP is the lack of standardization among the proprietary MDDB products that incorporate it. However, the fact that most MOLAP databases are read only and can automate the process of pulling data from all standard RDBMSs minimizes this drawback.

ROLAP, on the other hand, analyzes the original data in the current organizational database or a relational data warehouse. Using the source data directly lets users (in theory) drill down to the unit data level, usually by means of SQL extensions. However, the processing power to perform such analysis on the fly is enormous, requiring expensive, high-performance hardware. Additionally, ROLAP

Figure 2. The relational database model stores data in (a) two-dimensional tables and uses (b) entity-relationship diagrams to describe the relationships among data.



systems tend to use only current organizational databases and so lack historical data. ROLAP does have the advantage of being tied to the open systems standards of the underlying RDBMS.

GETTING STARTED WITH AN MDDB

Several reports have discussed high failure rates for OLAP projects; most projects expend too much effort for too little benefit. We agree with the observation—often called the 80-20 rule—that organizations gain 80 percent of the benefits for about 20 percent of the effort. The remaining 20 percent gain in functionality will require 80 percent of the effort and is almost certainly not worth it.

We also favor incremental development, which leads to greater acceptance by all stakeholders in the organization because it encourages the growth of a mutual understanding among stakeholders. Incremental development also lets an organization reach a balance between short- and long-term benefits.

TWO POPULAR APPROACHES

We began our research into MDDB by employing two different approaches to MDDB development:

- the top-down approach, which focuses on business planning, and
- the bottom-up approach, which focuses on existing systems and databases.

There is a general belief that the bottom-up approach is more common but more likely to lead to failure. This is known as the *data availability syndrome*; that is, just because you've got a lot of data is no reason to bring it all together and make it available to everyone. Two case studies highlight each approach's benefits and shortcomings.

Case 1: Top-down approach

Case 1 concerned the development of an MDDB system to register, track, and monitor problems handled by an IT help desk in a large, multinational company. The business manager provided requirements, from which the developer determined the measures and dimensions for the multidimensional modeling. The developer incorporated this information into a simple dimensional model and generated a prototype MDDB in a common OLAP product. This top-down approach ensured that the MDDB's measures and dimensions matched the business needs of the manager who would eventually use the information provided by the system. This approach builds an MDDB without concern for how or from where the system obtains data. The prototype—produced from sample data—won an enthusiastic reception from the manager, who readily understood its multidimensional form.

There are, however, at least two problems with this approach. First, the MDDB is not fully integrated with the underlying organizational databases, so the MDDB does not immediately reflect changes in the structure of those databases. For example, a business unit might add or amalgamate geographical regions in its databases, but the MDDB would not reflect these differences. Such a change would mean that the business unit's databases were no longer compatible with the multidimensional data model in the MDDB. Managers would have to decide how to anticipate and deal with potential changes to source data.

Second, it was clear that the developer tailored this system to the needs of one particular manager. The MDDB was not necessarily suitable, in the form it had, for more general use.

Case 2: Bottom-up approach

Case 2 concerned the development of an MDDB system to analyze existing student data at a large university. This data included both demographic information about students and student course records. The university stored the source data in an Oracle database with a relational design based on government reporting requirements. This implementation began with an analysis of data in the existing relational database to identify measures and dimensions

that related to a subject of interest. These measures and dimensions normally form what is known as the *star schema*, as shown in Figure 3.

This framework's key concepts are two types of tables:

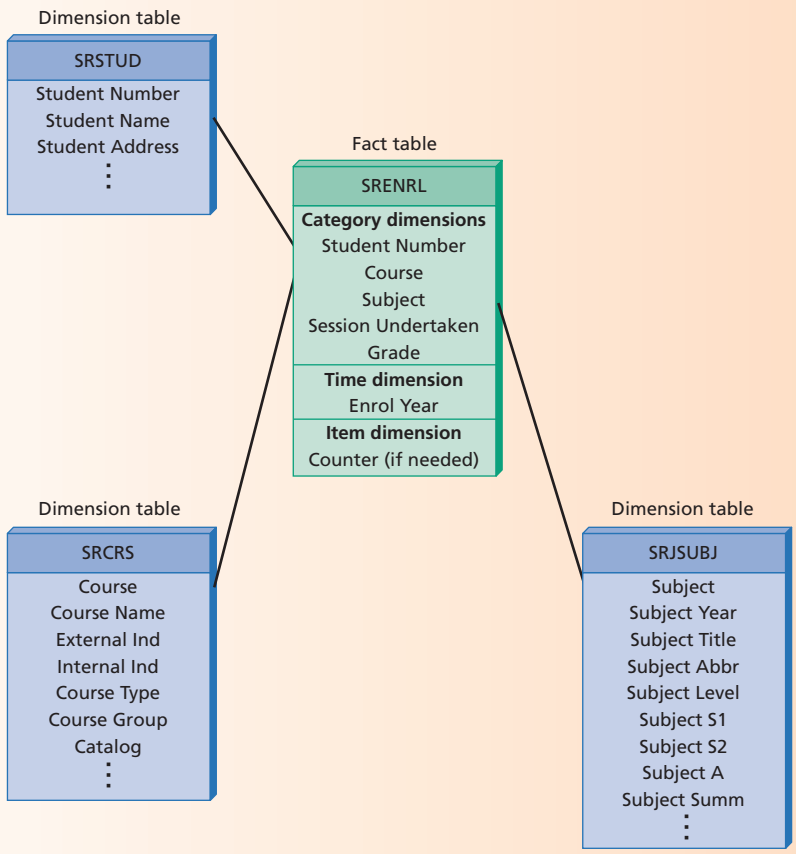
- a fact table, which consists of the numerical measurements that exist within the database, and
- dimension tables, which are more descriptive data items that map to the natural dimensions within the business.

The fact table consists of multipart keys that link back to the dimension tables, producing a star or snowflake schema.

To successfully translate data from a relational database into an MDDB it is necessary to identify the star schema within the relational database that relates to subjects that interest business analysts. The translation process is essentially one of denormalization and hence, simplification. It should, in principle, be possible using an automated procedure.

Such a procedure was available in the software product used in this case. However, extracting meaningful, multidimensional data from a large organizational, relational database was extremely difficult, even for highly qualified personnel. This was particularly true when (as is usually the case) the source data was not designed for the MDDB's purpose. In this case, we decided to convert only part of the whole data set into multidimensional form. The resulting system was of limited interest to management, whose main concern was to summarize information across the whole data set. Given the problems with the underlying data, management decided to cancel the project.

Figure 3. Dimensional model (star schema) for case 2, depicting the subject enrollments fact table and the surrounding lists of dimensions.



A MIDDLE-OUT APPROACH

We have used the lessons learned from cases 1 and 2 to create a composite approach to MDDB development. Our middle-out approach consists of the following guidelines for a fresh, pragmatic approach to MDDB development:

- Enlist a suitable manager to act as project sponsor. Interview this manager to determine an initial working set of measures and dimensions, focusing on the information needs of all managers.
- Locate data sources for as many of these measures and dimensions as possible, identifying a set of sample data that is relatively easy to extract.
- Identify any data available in these sources that the project sponsor has not identified. Have the sponsor evaluate the usefulness of such data to other managers.
- Eliminate measures or dimensions that are not essential to meeting managers' needs or are difficult to extract. This optimization applies the 80-20 rule.
- Build a prototype using an MDDB development tool and have both the project sponsor and database manager evaluate it.
- If necessary, have the database manager modify the source data to facilitate the extraction of MDDB data.

Case 3

We tested this approach on two new case studies. Case 3 concerned the development of an MDDB system to help a university's managers collect and track research publica-

tion data. The manager of the university's research office indicated which measures and dimensions would be useful for planning and management. The available on-site data collections were analyzed to identify existing data sources that might provide the required measures and dimensions. Each year's publication data was stored in a single flat file that could easily be exported to a comma-delimited text file. We used an OLAP tool to upload the comma-delimited file into a multidimensional model of this data.

Although the OLAP system worked well and met all its design requirements, senior managers have not used it, as was originally intended. However, the organization uploads each year's publication data into the system and several middle managers (unofficially) use the data. The reasons behind its use by middle managers rather than senior managers appear to be political, rather than technical. Despite the increased functionality offered by the OLAP system, the official publication records are still stored in flat files that do not provide senior managers with an effective view of the data.

In a middle-out approach, the bottom-up phases define source data quality and availability.

Case 4

Case 4 concerned the design and construction of a generic MDDB system to provide statistical data from external sources to company executives and market analysts. To determine user requirements, several typical users of external statistical data were interviewed. Reverse engineering of existing systems for disseminating this data provided additional requirements.

At this point, the requirements were exclusively top down. Although the system could store and analyze a variety of data—health information, trade figures, land values, and so on—this case study used Australia's 1991 census as the data source. The data contains the unit (record-level) data for each of 63,000 dwellings, 67,000 families, and 168,000 people. It contained over 60 dimensions, and the number of members in each dimension varies from two (sex of individual) to 58 (field of highest qualification). MDDBs were developed for several subsets of the available dimensions for each entity because MDDBs containing all the dimensions for any one entity were prohibitively large. Thus, the data itself determined the structure of each model, a bottom-up approach.

A group of 40 users, most of whom had no previous experience with OLAP tools, tested the system. After only a brief period of training, the majority of users could complete a set of 18 complex information retrieval tasks in less than an hour. Users also evaluated the system for ease of use and effectiveness. The results were very positive, suggesting that users found the OLAP interface easy to use and generally very effective.

COMPARISONS

The four case studies show that the multidimensional view of data can provide users with an effective means of making sense of large quantities of data. The simplified yet meaningful data structure in an MDDB matches the information needs of decision makers. Most notable of these are senior managers who regularly want summarized and comparative information on performance toward business goals.

In contrast, the underlying data from which an MDDB must extract useful information is, in most cases, stored in relational structures that facilitate the processing of day-to-day business transactions. The business data accumulated in organizational databases is often extremely complex. Case 2 demonstrated that this complexity is difficult to avoid if you use a bottom-up methodology to design an MDDB. In this case, the process of matching the data in relational OLTP tables to a multidimensional OLAP system was unnecessarily difficult, even with the use of a sophisticated tool.

The top-down approach of case 1 focused more on the meaning of the MDDB-delivered information than on the source data. This approach produces a user-oriented system, because information arranged by subject rather than by operational applications better supports management decision making. However, the complexities of identifying the appropriate source data and transferring it into the MDDB still remain a problem for IT professionals.

Cases 3 and 4 show the value of a combined or middle-out approach. Developers determine the measures and dimensions to be included in the MDDB by using an iterative approach to gathering specifications from both users and developers. At the same time, the availability and quality of source data tempers the selection of measures and dimensions.

Dealing with dimensions that change over time can pose a problem for MDDBs.

In a middle-out approach, the bottom-up phases define source data quality and availability. The top-down phases define the MDDB information's business meaning. A pragmatic approach using the 80-20 rule can ensure that the resulting system balances both bottom-up and top-down aspects to obtain the best advantage at reasonable cost.

Also, in contrast to an OLTP system, an MDDB does not require real-time data transfer because system administrators only repopulate an MDDB at infrequent intervals. So, rather than using a star schema as a direct link between a relational database and an MDDB, it is often easier to use the relational database's resources to denormalize data (via SQL queries) into a delimited text file for uploading into an MDDB. Although inefficient in terms of time and space, the repopulation process need only execute periodically, when resources are available.

ONGOING ISSUES

Although the middle-out approach seems to be both effective and efficient, unresolved issues remain.

MOLAP versus ROLAP

The decision about whether to use MOLAP or ROLAP is an instance of the space-time dilemma. MOLAP occupies more space by creating a physical cube, which duplicates data but greatly speeds up OLAP. On the other hand, ROLAP uses far less space but is inherently slower. From our perspective, a MOLAP application is more likely to be the result of a top-down approach, starting with a multidimensional model based on business needs. A ROLAP application is more likely to result from a bottom-up approach, extracting data from existing relational databases on the fly. Further research on the relative merits of MOLAP and ROLAP systems would be useful.

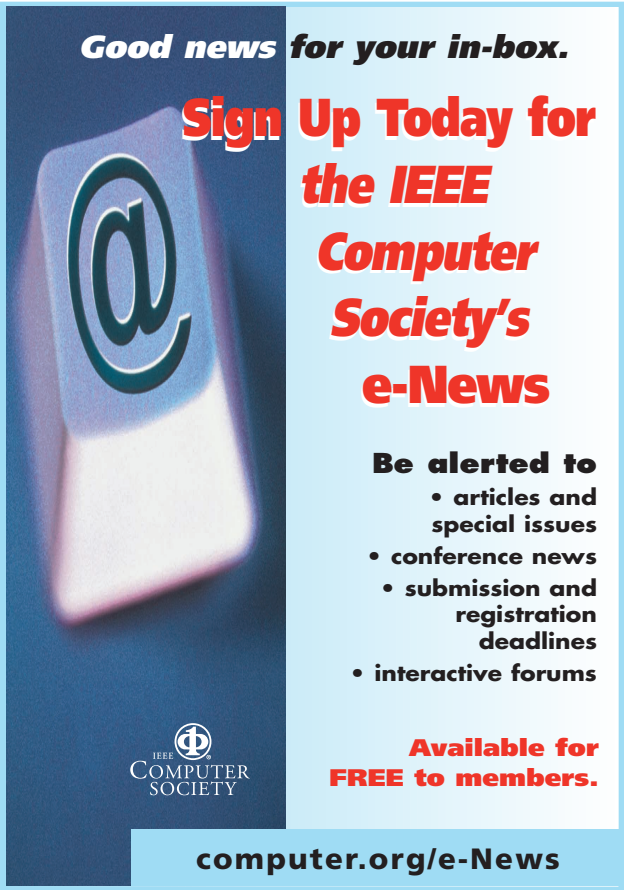
Identifying and fixing dirty data

In the second case study, data cleansing was the most significant problem encountered in setting up an MDDB based on source data located in a large organizational database. The bottom-up approach revealed that although the source data's organization was suitable for OLTP needs, it was unsuitable for analyses involving an overall business perspective.

For example, the system that produced the source data often calculated key performance indicators from several data values collected at significantly different times. This inconsistency meant that managers could not use these key performance indicators for meaningful analysis across the time dimension. The development of MDDBs often reveals such problems, and managers need practical ways of both anticipating and correcting problems when they emerge.

Changes to structure over time

A fundamental problem with an MDDB where time is a dimension is how to deal with dimensions whose structure changes over time. For example, a rapidly growing retail chain might want to report on sales at a regional



level, where previously they had only reported on sales at a state level. To accommodate such reporting, the company would need to redefine the location dimension and upload a new copy of the data. For small systems, like the prototypes described earlier, doing so was not a serious problem. However, some systems hold gigabytes of data, and this problem can become a logistical nightmare for which we have found no immediate solution.

Tailoring MDDB to individual managers

The literature examined during the first case study seemed divided on whether an MDDB's requirements should be based on the critical success factors and key performance indicators of an individual manager or those of the whole organization. From a pragmatic point of view, it is probably better to build systems to support an individual manager because of the difficulties involved in identifying all the measures and dimensions needed to create a dimensional data model for the whole organization.

Developing standard MDDB tools

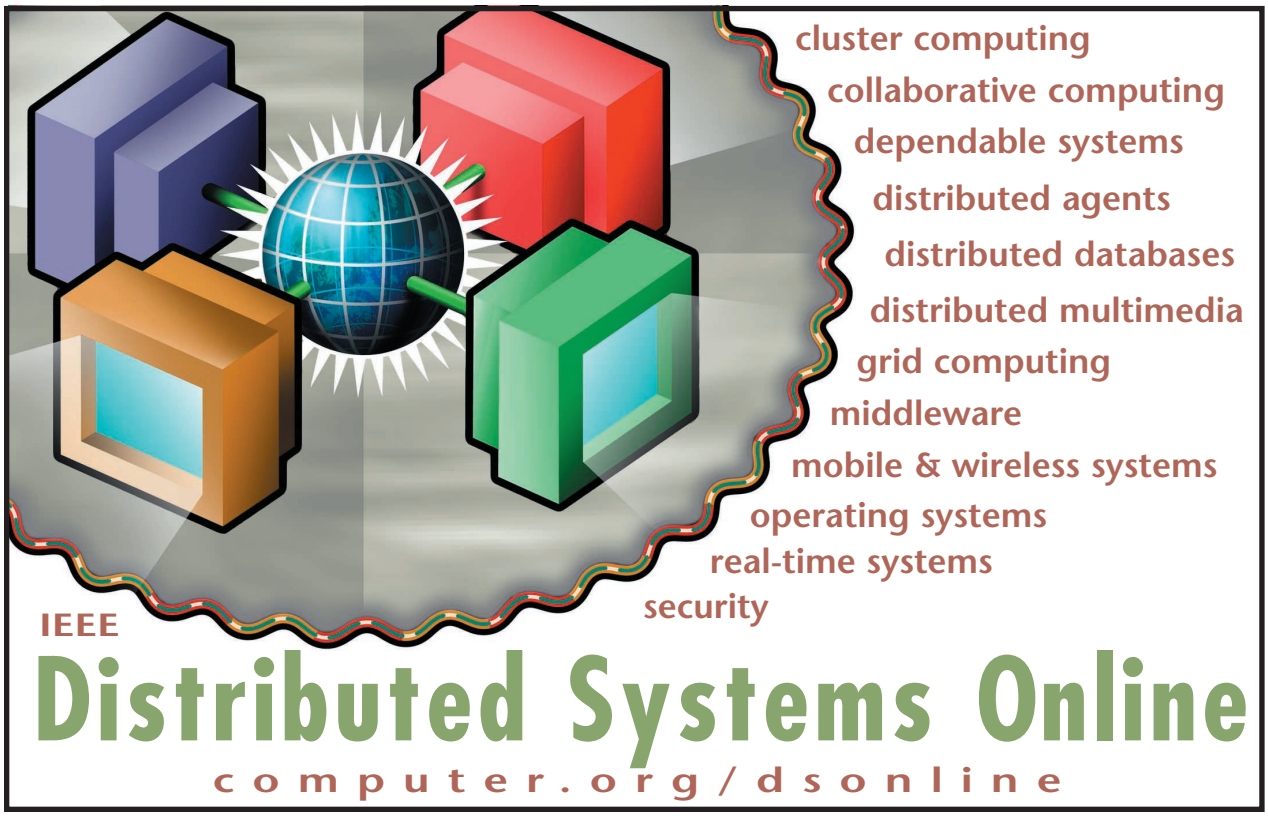
In our studies, the proprietary nature of OLAP tools made the development process more difficult than it should have been. Helpful reference guides and training packages for these products are rare. Developers also did not believe that the time or cost of vendor training was warranted because the product-specific skills are not transferable. Standard tools—similar to tools like entity-relationship modeling and SQL in the relational database sphere—should be developed for MDDB.

Skills for multidimensional modeling

All the IT professionals involved in the case studies reported difficulties with the basic concepts of measures and dimensions, and with identifying them in data. On the other hand, it was surprising how quickly the business managers grasped the dimension concepts. This suggests that familiarity with the data, together with the information needs of the business problem, is more significant than traditional database expertise in top-down, multidimensional modeling. Further research should test this hypothesis.

Manager education

Finally, we believe that management courses should place more emphasis on understanding information and knowledge as business assets for organizational planning, decision making, and innovation. ■

Helen Hasan *is head of the Department of Information Systems at the University of Wollongong, Australia. Contact her at helen_hasan@uow.edu.au.*

Peter Hyland *is a lecturer in information systems at the University of Wollongong, Australia. Contact him at peter_ hyland@uow.edu.au.*