Integrating Heterogeneous Databases: Lazy or Eager?

(Position Statement)

Jennifer Widom
Stanford University

widom@db.stanford.edu
http://db.stanford.edu/~widom

Providing integrated access to multiple, distributed, heterogeneous, autonomous databases and other information sources is a topic that has been studied in the database research community for well over a decade. There has been a surge of work in the area recently, due primarily to increased demand from customers ("real" customers as well as funding agencies). Nevertheless, despite the longevity of the subfield and the current large population of researchers working in the area, no winning solution or even consensus of approach has emerged.

In the research community, most approaches to solving the data integration problem are based very roughly on the following two-step process:

1. Accept a query, determine the appropriate set of information sources to answer the query, and generate the appropriate subqueries or commands for each information source.

2. Obtain results from the information sources, perform appropriate translation, filtering, and merging of the information, and return the final answer to the user or application (hereafter called the *client*).

We refer to this process as a *lazy* or *on-demand* approach to data integration, since information is extracted from the sources only when queries are posed. (This process also may be referred to as a *mediated* approach, since the module that decomposes queries and combines results often is referred to as a *mediator*.)

The natural alternative to a lazy approach is an *eager* or *in-advance* approach to data integration. In an eager approach:

1. Information from each source that may be of interest is extracted in advance, translated and filtered as appropriate, merged with relevant information from other sources, and stored in a (logically) centralized repository.

2. When a query is posed, the query is evaluated directly at the repository, without accessing the original information sources.

This approach is referred to as *data warehousing*, since the repository serves as a warehouse storing the data of interest.[1]

[1] Data warehousing as a buzzword has numerous other connotations. In fact, some would consider the integration aspect only a side-effect of data warehousing. Nevertheless, it's our belief that information integration will emerge as one of the most important uses of a data warehouse.

A lazy approach to integration is appropriate for information that changes rapidly, for clients with unpredictable needs, and for queries that operate over vast amounts of data from very large numbers of information sources (e.g., the world-wide web). However, the lazy approach may incur inefficiency and delay in query processing, especially when queries are issued multiple times, when information sources are slow, expensive, or periodically unavailable, and when significant processing is required for the translation, filtering, and merging steps. In cases where information sources do not permit ad-hoc queries, the lazy approach is simply not feasible.

In the warehousing approach, the integrated information is available for immediate querying and analysis by clients. Thus, the warehousing approach is appropriate for: (1) clients requiring specific predictable portions of the available information; (2) clients requiring high query performance but not necessarily over the most recent state of the information; (3) environments in which native applications at the information sources require high performance (large multi-source queries are executed at the warehouse instead); (4) clients wanting access to private copies of the information so that it can be modified, annotated, summarized, and so on; and (5) clients wanting to save information that is not maintained at the sources (such as historical information).

As stated above, in the past the database research community has focused primarily on lazy approaches to integration, although recent work has begun to consider the warehousing approach. What's the answer? Certainly there are scenarios that clearly favor one approach over the other. However, it is our belief that many of the complex, large scale inter-database applications of the future will require *both* approaches. The ideal information integration system to handle such applications will be one in which some data is fetched, processed, and integrated in advance and stored in the system's warehouse, while other data is fetched and processed only in response to user queries. Aside from performance and "freshness" considerations, the difference between pre-fetched (eager) and fetched-on-demand (lazy) data should be fully transparent to the client.

Surprisingly, extremely little research to date has considered the numerous technical problems associated with this flexible and general approach to the problem of integrating heterogeneous databases. Our position is that it's high time for this approach to be pursued.