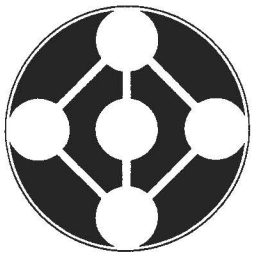

Alternative Technologies

The Importance of Enterprise Database Systems

David McGoveran
Alternative Technologies
13130 Highway 9, Suite 123
Boulder Creek, CA 95006
Telephone: 408/425-1859
FAX: 408/338-3113

I. Introduction

Few companies can afford to engage in wholesale replacement of information systems as new technologies develop, but neither can the new technologies be ignored. The situation poses a difficult problem: if the wrong strategy is chosen, IS costs can quickly outweigh any benefits. The RDBMS industry has been promising delivery of products which would significantly relieve this problem.

In this paper we explain why this database technology is essential for enterprise IS. RDBMS requirements of today's enterprise information systems are discussed in the broader context of enterprise IS goals. The key issues being addressed by RDBMS vendors are examined, including reliability, scalability, costs, and extensibility through open systems support.

II. Enterprise IS

Enterprise IS sometimes appears to be the Holy Grail of corporate computing. The goal is to manage business information system requirements for the entire enterprise. This is particularly difficult when requirements change faster than systems can be built. The cost of building and maintaining smaller systems that meet particular needs has steadily increased over the years. At the same time, managing large-scale projects has become increasingly difficult and the resulting systems more rigid. The costs continue to increase for two key reasons; first, new systems are often subtlely redundant with respect to existing ones and, second, this lead to disparities in shared data that must be rectified. Over the years, enterprise computing environments have changed considerably. Early IS implementers had few of the problems we see today because they usually were involved in automating some specific labor-intensive and mission critical application. During the 1960's, many larger corporations moved to computer-based financial systems. Financial applications such as

accounting and payroll were so much a part of business computing that programmers were often required to understand basic accounting practices. Due to standard accounting practices, the tax laws, the high cost of mainframes, and the relatively slow evolution of computing hardware, business applications were extremely stable. The ideas of legacy applications and disparate or corrupt data were hardly conceivable in such an environment. Neither was it necessary to consider integrating computing systems, since costs dictated that IS be centralized.

Through the 1970's, the charter of corporate computing changed radically. Inventory control and manufacturing systems became more prevalent. Minicomputers became available for divisional and departmental computing. UNIX was introduced and became popular, in part, as one of the first operating systems that were not tied to any particular computer manufacturer. Special purpose systems for word processing became available, and, late in the decade, the personal computer brought computing within the realm of non-IS managers.

With the advent of commercial relational DBMSs in the early 1980's, IS was faced with the idea of databases that could be designed to support shared data among as-yet-unspecified applications. For the first time, end users had not only the computing power, but also the technical ability to access corporate data directly. New applications were conceived to address every conceivable aspect of the enterprise, each of which could create and maintain its own database. Decentralized computing was sometimes achieved, but usually at a high cost; integration, consistency, and control were effectively lost. Applications often were designed to meet specific, immediate needs, without planning for longer term maintainability or flexibility.

Client/server database applications, introduced in the mid-1980's, provided some degree of control over the database portion of applications through a new form of centralization. It also had an enabling effect on open systems and interoperability. With the introduction of graphical user interfaces, the number of applications managed -- and the complexity of operations requested -- by end-users compounded the problems. IS had to take into account the needs of many types of hardware and software, as well as a user-community with a broad range of computer skills, training, and expectations.

These problems have been compounded by the speed with which new technology has been made available. With lower cost hardware and the client/server approach, the incremental cost to support an additional user has steadily decreased. Unfortunately, the recently touted hidden maintenance costs due to indiscriminant downsizing can easily outweigh the savings. It is unclear that other savings (such as those due to increased productivity, stress reduction, better local autonomy, etc.) do not compensate, but the effects are difficult to quantify.

The solution appears to be an enterprise-wide framework for designing smaller information systems. Several of the key companies in the computer industry are promoting such frameworks, among them DEC's NAS, IBM's SAA, and Apple's VITAL. However, just as the availability of commercial technologies has enabled the demand for enterprise frameworks, none of these frameworks will succeed unless commercial technologies support them.

Among these, enterprise database technology is the most essential. To understand why this is true, consider the following points:

- If an enterprise RDBMS fails or is unavailable, damage affects all applications using data in that RDBMS
- If an enterprise RDBMS is unable to provide adequate performance or is not scalable, the throughput of the entire enterprise IS is affected
- The cost per transaction of the RDBMS is typically the determining factor in the cost of the enterprise system
- If the RDBMS provides limited support of open systems and connectivity standards, all applications and their supporting hardware are constrained and islands of data develop

III. Enterprise RDBMSs

Enterprise DBMSs should be reliable, scalable, cost effective, and open. These characteristics are directly related to today's essential enterprise IS goals.

Reliability is one of the primary goals of enterprise IS. There are three key reasons for this. First, without reliability, the very definition of enterprise IS cannot be met because general and standard operational procedures cannot be developed. This means that technical specialists are required, raising the costs of training and operations, and increasing the complexity of recovery after failures. Second, a variety of processing types needs to be supported by an enterprise IS approach including OLTP, OLCP, batch, ad-hoc, and decision support. This situation places importance on reliable (i.e., predictable) behavior of the system with respect to error conditions, performance, and load. Third, some of these applications will undoubtedly be mission critical. For mission critical applications, unavailability due to failures or maintenance requirements quickly becomes more than an inconvenience. Clearly, an enterprise RDBMS must be reliable.

Scalability is an important goal of enterprise IS for two main reasons. First, because business requirements change so rapidly, it is no longer possible to perform the kind of rigid and time-consuming long-term capacity planning that was once promoted in MIS. Certainly, capacity planning is still needed, but the most

common enterprise IS approach is to provide for incremental, evolutionary increases or decreases in capacity rather than the costly stepwise, revolutionary changes. Second, technology changes rapidly, resulting in new capabilities and lower hardware costs. Without an incremental approach, the enterprise cannot afford to take advantage of new technology without depreciating older technology. In today's business environment, this means that older businesses cannot compete with newer ones due to outdated capital investments. These facts require that both the hardware and software, especially the RDBMS, be scalable.

Cost effectiveness insures that the benefits of the enterprise computing strategy will outweigh the costs of ownership and operation. There are two main reasons why cost effectiveness is a key concern. The first is obvious: Budgetary constraints and profitability are always important to a business. The second reason is a little more complex and requires some detailed explanation: The assumed benefits of distributed computing have proven sensitive to the strategy used and hidden costs have been discovered.

Most enterprise IS frameworks support some kind of distributed computing, typically client/server or peer-to-peer. Migrating to distributed computing environment from a centralized or even remote computing environment usually involves careful *downsizing* or *rightsizing* choices. Experience has shown that important benefits from downsizing via client/server are achieved in terms of development time, ease of use, and, when properly designed, performance. It is difficult to anticipate and measure the quantitative benefits of enterprise IS strategies. This can be done through "before and after comparisons such as comparing application backlogs (often it is the rate at which the backlog is increasing that is reduced rather than the actual backlog), performing user satisfaction studies, turnaround time for satisfying new application requests, etc.

Measuring costs of ownership is less difficult than measuring benefits. It should include the obvious costs associated with capacity requirements (such as $/user or $/TPS) as well as other costs. While most of the industry was convinced of the benefits of downsizing and client/server, the significant costs associated with code maintenance, software licensing, distribution, configuration management, and system administration came as a surprise to many. Perhaps the most significant costs are determined by the choice of RDBMS. One approach to cost effectiveness is to lower the cost of RDBMS ownership.

Enterprise IS strategies commit to proprietary systems less and less often in an effort to build *extensible* systems. Instead, the goal is to implement enterprise IS with *open systems*. There are three reasons for this goal. First, no vendor can offer all the technologies needed by enterprise IS. Second, even the largest and best established vendor can have difficulty meeting delivery schedules at times, may suffer an economic setback, or may not offer the best prices. A well-planned enterprise IS

strategy will not risk dependence on a single vendor. Third, given the rapidly changing technology, it is unlikely that the best technological solution will be developed by any single company.

In the distributed computing environments common to enterprise IS, open systems are expected to reduce the costs associated with systems connectivity and interoperability. Open systems help the enterprise extend the life of existing systems, preserve the investment in systems under development, and minimize the costs of migrating to new technologies in the future. For client/server implementations, open systems support is especially important. The ability to connect various client platforms to the RDBMS server and of various RDBMS server platforms to interoperate are key goals of the client/server computing model.

IV. Conclusions

Not all RDBMS products meet the needs of enterprise IS. If an RDBMS is to be the key component of an enterprise IS strategy, it must be reliable, scalable, cost effective, and extensible. Great care must be taken in evaluating RDBMS products.

Contrary to the predictions of non-technical industry analysts, extensive experience in evaluating RDBMS products shows they are not becoming commodity items. On the one hand, the differences between these products is becoming more subtle and difficult to understand. On the other, these subtleties make the difference between success and failure. If all RDBMS products were identical, the cost of replacing one with the other would be limited to the license fees and DBA training costs.

Unfortunately, applications, database designs, optimization, transaction profiles, and even SQL syntax must be changed when an enterprise RDBMS product is changed. Given these costs, enterprise IS cannot afford to select the wrong RDBMS. And the database is arguably the single-most important factor in successful enterprise IS efforts today.

References:

D. McGoverna, "ORACLE7", Database Product Evaluation Report Series, c. 1992, Alternative Technologies.

