Providing OLAP to User-Analysts: An IT Mandate

Introduction

Overview
Recently, there has been a great deal of discussion in the trade press and elsewhere regarding the coexistence of so-called transaction databases with decision support systems. These discussions usually revolve around the argument that the physical design required for acceptable performance of each is incompatible and that therefore, data should be stored redundantly in multiple enterprise databases: one for transaction processing, and the other for decision support type activities. Also, these same arguments usually confuse physical schema with logical and conceptual schema.

These arguments are fuzzy and imprecise. These arguments ignore the fundamental requirements of the types of analytical data models required for efficient information synthesis and also ignore the fact that the majority of enterprises have numerous, diverse data stores from which information needs to be synthesized. This paper defines a category of database processing: Online Analytical Processing (abbreviated OLAP). This paper defines the OLAP category, describes an enabling architecture for OLAP, and identifies the fundamental components and criteria for evaluating a given product's efficacy in its support of the OLAP category. Finally, a commercially available product is evaluated according to the rules for OLAP. In this paper the symbol DBMS denotes a database management system.

E.F. Codd, S.B. Codd and C.T. Salley

E.F. Codd Associates

The Evolution of OLAP

The Relational Model

The Relational Model was developed to address a multitude of shortcomings that existed in the fields of database management and application development. Prior to the publication, acceptance, and development of systems based on E. F. Codd's work, the database management systems in the marketplace were home grown, ad hoc collections of ideas formulated into systems. One interesting characteristic of these systems was that each was originally designed to solve a particular type of problem and then later was extended to become a more general purpose solution.

> *The original objectives in developing the Relational Model were to address each and every one of the shortcomings that plagued those systems that existed at the end of the 1960s decade.*

The resulting systems represented a collection of products that were needlessly complex. Each also suffered from the concomitant problems of being difficult to understand, install, maintain and use. Moreover, these products required significantly large budgets and supporting staffs of people with significantly difficult-to-attain skills that were in short supply. Access to these systems required prior preparation by staffs of database administrators and application developers. Unanticipated end-user access to the data was rarely provided by the DBMS. A separate query-only system was available from some DBMS vendors, but not from others. At that time, no DBMS product supported the maintenance of logical integrity of the data as a DBMS responsibility.

The original objectives in developing the abstract model known as the Relational Model were to address each and every one of the shortcomings that plagued those systems that existed at the end of the 1960s decade, and make DBMS products more widely appealing to all kinds of users. With an abstract model for database management based on mathematical principles and predicate logic, the developers of future database management systems would have a blueprint to follow in systematically creating a variety of products and tools that could be built and enhanced over time.

> *In every industry, relational systems are being used for applications requiring storing, updating and/or retrieval of single as well as multiple shared data elements, for operational, transaction, and complex processing.*

Today, although falling short of the capabilities offered by the Relational Model itself, existing relational database management systems offer powerful, yet simple solutions for a wide variety of commercial and scientific application problems. In every industry, relational systems are being used for applications requiring storing, updating and/or retrieval of single as well as multiple shared data elements, for operational, transaction, and complex processing as well as decision support systems, including query and reporting; in short, for every conceivable type of application. Recent versions of these systems provide system-managed concurrency for multiple application programs and multiple interactions with the database by end users.

The Evolution of OLAP

Expanding Data Bases and Need for Analysis

Corporate data has grown consistently and rapidly during the last decade. During the 1980's, businesses and governments worked with data in the megabytes and gigabyte range. Contemporary enterprises are having to manipulate data in the range of terabytes and pedabytes. Concurrently, the need for more sophisticated analysis and faster synthesis of better quality information has grown.

Today's markets are much more competitive and dynamic than those in the past. Business enterprises prosper or fail according to the sophistication and speed of their information systems, and their ability to analyze and synthesize information using those systems. The numbers of individuals within an enterprise who have a need to perform more sophisticated analysis is growing.

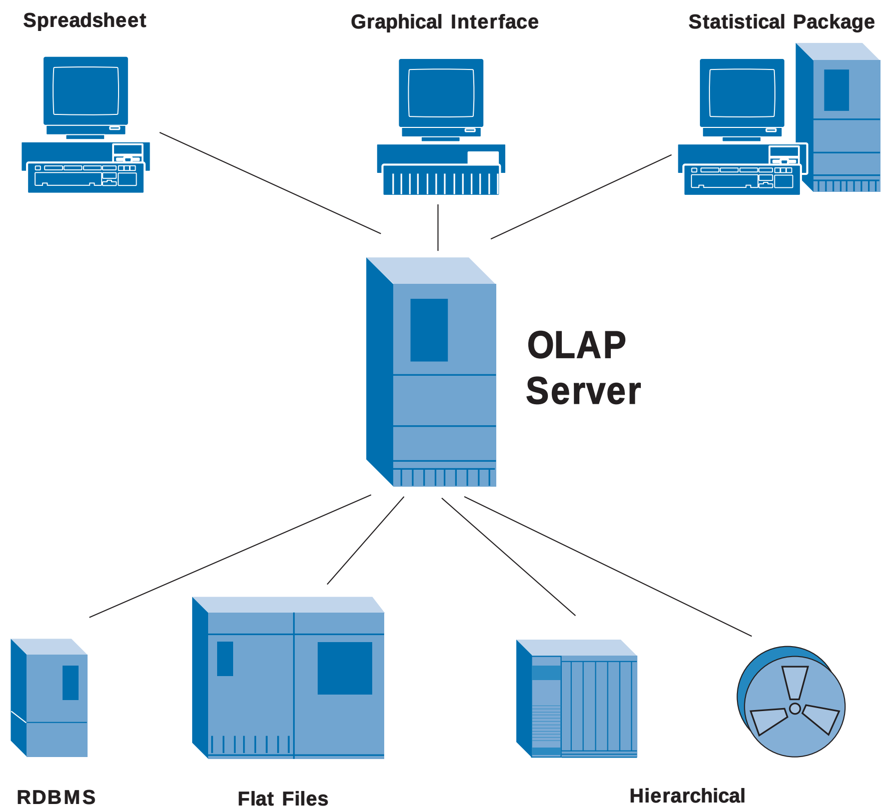
> *Business enterprises prosper or fail according to the sophistication and speed of their information systems, and their ability to analyze and synthesize information using those systems. The numbers of individuals within an enterprise who have a need to perform more sophisticated analysis is growing.*

Data in relational systems is also being accessed by a wide variety of non-data processing professionals through the use of many different types of tools and interfaces. These include general purpose query products, spreadsheets, graphics packages, off the shelf application packages for human resource management, accounting, banking, and other disciplines. Moreover, as the emphasis upon interoperability becomes more pronounced, these products are finding their way to users with every conceivable type of hardware architecture.

The following diagram illustrates the mediating role that an OLAP Server provides with respect to the various types of databases and files in which data may be stored and the numerous types of front-end packages that the end users may need. These front-end packages (only 3 are shown) are placed at the top of the diagram, while the data organization types (only 4 are shown) are placed at the bottom. The OLAP Server is in the center of the diagram. This mediating role is a very important property that an OLAP Server should have.

With relational technology, the system complexity of pre-relational systems has been replaced by ease of learning, ease of use, and support for *ad hoc* query and manipulation. Moreover, a relational DBMS includes a more powerful means of preserving the logical integrity of the data than any pre-relational DBMS. This major feature permits enterprises to acquire confidence in the accuracy of the data. At every turn, the relational database management system has become the gateway to enterprise data. Additionally, the relational language of the system has become the interface to that data store for all of the end-user products in all of these different environments and architectures.

The Evolution of OLAP



Mediating Role of the OLAP Server

Of the wide variety of business applications that have been afforded faster, cheaper and better solutions in the relational DBMS world, perhaps none are more dramatic than query/report processing. Once handled almost exclusively by COBOL application programmers, the combination of the powerful relational DBMS coupled with the easy to learn, easy to use query/spreadsheet product has enabled end-users to develop and execute these query/report applications themselves.

Thus empowered, end-users now to a large extent satisfy their own requirements. Not only are they able to experiment with various data formats and aggregations, they are able to improve the information content of their reports. Moreover, they can do this on de-mand, all the while avoiding the long delays waiting for support from database administration and application development that characterized the pre-relational days.

> *However, as enabling for end-users as these new relational DBMS products and associated tools and interfaces have been, there are still significant limitations to their efficacy… commercial DBMS products do have boundaries with respect to providing function to support user views of data.*

However, as enabling for end-users as these new relational DBMS products and associated tools and interfaces have been, there are still significant limitations to their efficacy.

The Evolution of OLAP

The Relational Model dictates relational DBMS system design that provides unprecedented power in storing, updating and retrieving data. The power of any one specific relational DBMS when compared to the power of the Relational Model is dependent on the extent to which the DBMS is faithful to the Relational Model.

However, commercial DBMS products do have boundaries with respect to providing function to support user views of data. The DBMS products of today rely on front-end products to embellish their support of possible ways in which business analysts might wish to consolidate and view different kinds of enterprise data. Also, because of the limited support in existing DBMS products for dynamic physical representations of the data, and representations which adjust to provide optimum performance in accordance with the way the data is actually used, static physical designs often impede certain data analysis activities.

Most notably lacking has been the ability to consolidate, view, and analyze data according to multiple dimensions, in ways that make sense to one or more specific enterprise analysts at any given point in time. This requirement is called "multidimensional data analysis."

Until recently, the end-user products that had been developed as front-ends to the relational DBMS provided very straightforward simplistic functionality. The query/report writers and spreadsheets have been extremely limited in the ways in which data (having already been retrieved from the DBMS) can be aggregated, summarized, consolidated, summed, viewed, and analyzed.

Most notably lacking has been the ability to consolidate, view, and analyze data according to multiple dimensions, in ways that make sense to one or more specific enterprise analysts at any given point in time. This requirement is called "multidimensional data analysis." Perhaps a better and more generic name for this type of functionality is online analytical processing (OLAP), wherein multidimensional data analysis is but one of its characteristics.

The term "OLAP" is defined, its fundamental characteristics including multidimensional data analysis are examined, the business requirement for OLAP is discussed, and the types of users who are likely to benefit most from OLAP are identified. Additionally, twelve rules specifying key functionality for OLAP tools are presented and one commercially available product that appears to be defining this marketplace is evaluated according to the twelve OLAP rules.

Data Analysis Models

In his seminal paper *Extending the Database Relational Model to Capture More Meaning*, E. F. Codd has pointed out that when discussing semantic data modeling, "there is a strong emphasis on structural aspects, sometimes to the detriment of manipulative aspects. Structure without corresponding operators or inferencing techniques is rather like anatomy without physiology." Likewise, having a large enterprise-wide database is of little value if the end-users are unable to synthesize necessary information readily from those data stores. All too often, this is precisely the case.

Attempting to force one technology or tool to satisfy a particular need for which another tool is more effective and efficient is like attempting to drive a screw into a wall with a hammer when a screwdriver is at hand: the screw may eventually enter the wall but at what cost?

IT should never forget that technology is a means to an end, and not an end in itself. Technologies must be evaluated individually in terms of their ability to satisfy the needs of their respective users. IT should never be reluctant to use the most appropriate interface to satisfy users' requirements. Attempting to force one technology or tool to satisfy a particular need for which another tool is more effective and efficient is like attempting to drive a screw into a wall with a hammer when a screwdriver is at hand: the screw may eventually enter the wall but at what cost?

OLAP Concepts

Flexible Information Synthesis

The synthesis of information from large databases is a task performed by all database end-users and business analysts. The primary approach to synthesis of information from data is analysis. Historically, the majority of research and investigation in the field of data analysis has centered upon the comparison of one static data value with another as is commonplace in most online transaction processing systems. Increasingly, more and more users are being required to perform more sophisticated, dynamic analysis of historical data according to consolidation approaches such as those previously defined by E. F. Codd. As individual businesses seek to meld their information systems with those of their business partners for additional strategic advantage, the number of users who are required to perform this more sophisticated data analysis grows, as does the number of disparate data stores from which to synthesize information, and the number of diverse dimensions across which analysis must be performed.

As individual businesses seek to meld their information systems with those of their business partners for additional strategic advantage, the number of users who are required to perform this more sophisticated data analysis grows, as does the number of disparate data stores from which to synthesize information, and the number of diverse dimensions across which analysis must be performed.

One of the fundamental flaws of traditional data analysis and database design is the pervasive reliance upon the so-called "Entity-Relationship" (E-R) approach to data modeling. Because of its denial of the domain concept, E-R is ignorant of all of the intra-database relationships based upon common domains and all of the inter-tabular relationships, except for those manifested by primary key-foreign key relationships. For this reason, many users and some technicians, jump to the erroneous conclusion that relational databases are inappropriate for concurrent support of online transaction processing and decision support. This is not true.

OLAP Online Analytical Processsing

> *The need which exists is NOT for yet another database technology, but rather for robust OLAP enterprise data analysis tools which complement the enterprise's existing data management system and which are rigorous enough to anticipate and facilitate the types of sophisticated business data analysis inherent in OLAP.*

Relational databases have been, are today, and will continue to be the most appropriate technology for enterprise databases. The need which exists is NOT for yet another database technology, but rather for robust OLAP enterprise data analysis tools which complement the enterprise's existing data management system and which are rigorous enough to anticipate and facilitate the types of sophisticated business data analysis inherent in OLAP.

The activities of these analytic tools against the enterprise database constitute a transaction. However, the duration of this transaction may be many times longer than its traditional OLTP counterpart (days or weeks versus seconds or perhaps minutes). Another difference between these types of transactions is that OLTP applications of necessity must work with instantaneously accurate data, whereas what the authors call "online analytic processing (OLAP)" works almost exclusively with historical data deemed accurate as of a given point in time, specifically the beginning of the OLAP transaction.

> *OLAP is made up of numerous, speculative "what-if" and/or "why" data model scenarios executed within the context of some specific historical basis and perspective.*

OLAP is made up of numerous, speculative "what-if" and/or "why" data model scenarios executed within the context of some specific historical basis and perspective. Within these scenarios, the values of key variables or parameters are changed, often repeatedly, to reflect potential variances in supply, production, the economy, sales, marketplace, costs, and/or other environmental and internal factors. Information is then synthesized through animation of the data model which often includes the consolidation of projected enterprise data according to more than one data consolidation path or dimension. Results of online analytic processing are normally displayed on terminals, but may have to be recorded in some of the data organizations that are supported. When these results are stored in a database, it is necessary to be sure that speculative data is kept separate from and not confused with data that represents the actual state of the enterprise.

Multiple Data Dimensions/Consolidation Paths

Data consolidation is the process of synthesizing pieces of information into single blocks of essential knowledge. The highest level in a data consolidation path is referred to as that data's dimension. A given data dimension represents a specific perspective of the data included in its associated consolidation path. There are typically a number of different dimensions from which a given pool of data can be analyzed. This plural perspective, or Multidimensional Conceptual View appears to be the way most business persons naturally view their enterprise. Each of these perspectives is considered to be a complementary data dimension. Simultaneous analysis of multiple data dimensions is referred to as multidimensional data analysis.

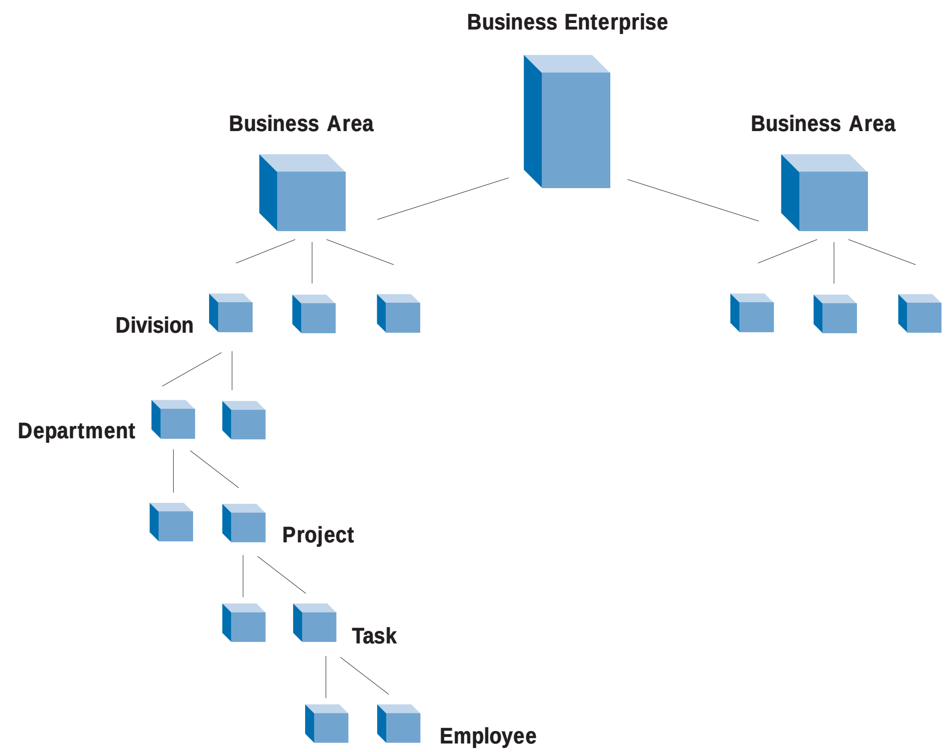
There are typically a number of different dimensions from which a given pool of data can be analyzed. This plural perspective, or Multidimensional Conceptual View appears to be the way most business persons naturally view their enterprise.

Historical data sets, regardless of content or origin, can be consolidated in a vast number of different ways. Each data consolidation path reflects the perspective and intent of its creator and consequently, emphasis is placed upon the factors of interest to that individual at that particular point in time, effectively de-emphasizing factors that may be of greater interest to other users or at some point in the future.

Furthermore, considerations leading to the specification of consolidation paths change over time. When an enterprise database is designed, it is designed for a specific purpose just as a data consolidation path is constructed to portray a particular perspective of data. The database designer has implied in the database design a particular consolidation path which usually has been dictated by the priorities of critical traffic at the time the database was designed.

Data consolidation paths consist of series of consolidation levels or steps that are defined in terms of multi-level parameters. These parameters apply to values from any variable where each successive level represents a higher degree of data consolidation. Thus for example, "business enterprise" could serve as a consolidation path for the variable "worker." "Business enterprise" might include as a consolidation path the levels "business area," "division," "department," "project," "task," and "employee." Data from the "employee" level could then be aggregated from the "task" level to each more aggregated business organization level.

If "task" were the sum of a variety of specially-skilled workers such as plumbers, electricians, and lathe operators, then "skill" would serve as a variable consolidation path. When variables such as these are consolidated, the consolidation takes place relative to some parameter location, e.g., project Alpha, April. Consolidation variables can be evaluated wherever their component variables are evaluated. Changing the consolidation level of a variable does not affect the consolidation level of the data.



Sample Data Consolidation Paths

Once data has been consolidated according to one or more consolidation paths, drilling down to greater levels of detail is possible. Drilling down refers to the movement from higher to lower levels of consolidation. Conversely, rolling up refers to the movement from lower to higher levels of consolidation.

Once data has been consolidated according to one or more consolidation paths, drilling down to greater levels of detail is possible.

Consolidation paths determine which details are visible to the end-user when the user drills down. Since consolidation paths are vertical by nature, concurrent horizontal relationships which may themselves be of interest to a user are not available without creating an additional data consolidation path or dimension. This represents significant effort in technologies not specifically designed to facilitate this type of dynamic data analysis.

Consolidations may involve simple roll-ups or more complex relationships or equations and computations which span multiple consolidation paths or dimensions.

OLAP is the name given to the dynamic enterprise analysis required to create, manipulate, animate, and synthesize information from exegetical, contemplative, and formulaic data analysis models (see the later section headed “Enterprise Data Models”). This includes the ability to discern new or unanticipated relationships between variables, the ability to identify the parameters necessary to handle large amounts of data, to create an unlimited number of dimensions (consolidation paths), and to specify cross-dimensional conditions and expressions.

OLAP Characteristics

Dynamic Data Analysis

Once data has been captured in a database, the analytical process of synthesizing the data into information can begin. Effective data analysis requires the use of different types of data analysis tools and data models. Data analysis which examines historical data without the need for much manipulation is referred to as static data analysis. Tools in support of this type of analysis are common, and perform primarily comparisons of data values. Databases designed and tuned for OLTP tend to support this type of analysis fairly well.

Dynamic data analysis can provide an understanding of the changes occurring within a business enterprise, and may be used to identify candidate solutions to specific business challenges as they are uncovered, and to facilitate the development of future strategic and tactical formulae.

Data analysis in which historical data must be manipulated extensively is referred to as dynamic data analysis. Dynamic data analysis involves multiple data dimensions and is the more challenging of the two types of analysis to perform in the traditional OLTP or OLCP environment. Dynamic data analysis is concerned primarily with the creation and manipulation of enterprise data models which access and usually update the subject of the analysis many times, across multiple dimensions. Dynamic data analysis can provide an understanding of the changes occurring within a business enterprise, and may be used to identify candidate solutions to specific business challenges as they are uncovered, and to facilitate the development of future strategic and tactical formulae.

Four Enterprise Data Models

Both of these types of data analysis employ data models as their principal tool. The data models used in both static and dynamic data analysis fall into four categories: the categorical model; the exegetical model; the contemplative model; and the formulaic model.

The construction and manipulation of each of these four model types is different. These data models and their use follows a continuum from actual historic data to future anticipated data values and behaviors; the requirement for analysis of the data across multiple dimensions, the degree of analyst interaction and involvement required to construct and manipulate each of these models increases along this same axis.

Categorical, Exegetical, Contemplative, and Formulaic Models

At the starting point of the continuum is the categorical data model. This model is employed in static data analysis to describe what has gone on before by comparing historical values or behaviors which have typically been stored in the enterprise database. Because this is a static data model, little if any user interaction is required and the data is viewed according to the previously defined consolidation path inherent in the original database design. A simple query can be executed against a database and value comparisons can be made with ease. Tools supporting this type of analysis are plentiful. Common database query facilities, report writers, and spreadsheets are examples of tools available for creation and analysis of this type of static model.

Moving along the continuum, the exegetical model reflects what has previously occurred to bring about the state which the categorical model reflects. This tool is primarily a static data analysis model with which the user peels back one or more layers of the onion through subsequent simple queries. If the analyst is interested in any motivation other than that inherent in the original database design, multiple dimensions must be created on the fly. Few tools today support this multidimensional view of the analytical model.

The third model, the contemplative model, indicates what outcomes might result from the introduction of a specific set of parameters or variances across one or more dimensions of the data model. This type of analysis is significantly more dynamic and requires a higher level of interaction on the part of the data analyst. Numerous consolidation paths may have to be created and examined as the business analyst specifies certain critical variables and equations across multiple data dimensions, animates the model, and gains insight into what the possible outcomes of these actions may be. A few spreadsheets offer a small subset of this capability within a single data dimension, but none provides this functionality across multiple dimensions.

The fourth data model, the formulaic model, is the most dynamic and requires the highest degree of user interaction and associated variable data consolidation. This data model indicates which values or behaviors across multiple dimensions must be introduced into the model to effect a specific outcome. Within this dynamic model, the analyst specifies an outcome, and a starting point, animates the model, and gains insight as to what variables of which dimensions must be manipulated and to what extent, if the specified outcome is to be attained. No product of which we are aware provides this capability today.

Vendors have for the most part taken the path of least resistance and have concentrated upon the development of tools in support of static data analysis and the categorical model.

Vendors have for the most part taken the path of least resistance and have concentrated upon the development of tools in support of static data analysis and the categorical model. They have, with the exception of delivery of a few cumbersome, mainframe and mini-computer based, batch facilities, ignored users' requirements for robust workstation tools supporting the creation and manipulation of the last three types of dynamic data analysis and strategic enterprise support models.

Common Enterprise Data

The data required for OLTP systems is the same data which is required for OLAP. The nature of the transactions differs, as does the need for the data to be strictly up-to-date, but both types of processing take place against the same data stores. Frequently, especially in the use of exegetical analysis, the analyst must "drill down" within a particular data consolidation path to discern why a particular situation is as it is. Referring back to the actual transaction data is often necessary. This data MUST be readily available to the enterprise user-analyst.

A Complementary Approach

Synergistic Implementation

While relational systems provide functions to select, compare, scope and aggregate collections of data according to different use views or needs, they are limited in their ability to present data in different formats, under different headings and according to diverse dimensions. In fact, relational DBMS were never intended to provide the very powerful functions for data synthesis, analysis, and consolidation that is being defined as multidimensional data analysis. These types of functions were always intended to be provided by separate, end-user tools that were outside and complementary to the relational DBMS products.

In fact, relational DBMS were never intended to provide the very powerful functions for data synthesis, analysis, and consolidation that is being defined as multi-dimensional data analysis. These types of functions were always intended to be provided by separate, end-user tools that were outside and complementary to the relational DBMS products.

Unfortunately, during the several years that relational DBMS and the concomitant end-user tools have been available, the existing versions of these query-report tools and spreadsheets remained completely ignorant of the requirement for OLAP, both with respect to the wide variety of uses for the function, and the large numbers of diverse end-user analysts who could significantly improve their productivity were this function available. It is only very recently that the requirement for OLAP has become evident and understood.

The challenge has become one of how to get this function out into the marketplace, into the hands of user-analysts, with a minimum of new education and without sacrificing their existing levels of productivity. Since the end-user has become very comfortable with the interface to the spreadsheet, the most obvious approach would have been to simply add the function to the spreadsheet product.

Examination of the collection of functions requiring implementation in order to support OLAP suggests the inclusion of the following:

- access to the data in the DBMS or access method files;
- definitions of the data and its dimensions required by the user;
- the variety of ways and contexts in which the user might wish to view, manipulate, and animate the data model; and
- accessibility to these functions via the end-user's customary interface.

With the exception of retrieving the data in question, all the other functions listed are really enhancements of the kinds of capabilities one would expect in a spreadsheet product. Unfortunately, the spreadsheet vendors have not shown much interest in supporting online analytical processing to the required degree of robustness as defined by the OLAP evaluation rules.

Consequently, an architectural framework is required within which the function could be made to appear as if it were part of the user analyst's customary spreadsheet product. This framework has become a significant measure of the efficacy of the multidimensional data analysis product itself, and, in fact, represents the first criterion for evaluating OLAP products.

OLAP Product Evaluation Rules
The twelve rules for evaluating OLAP products are:

1. Multidimensional Conceptual View
2. Transparency
3. Accessibility
4. Consistent Reporting Performance
5. Client-Server Architecture
6. Generic Dimensionality
7. Dynamic Sparse Matrix Handling
8. Multi-User Support
9. Unrestricted Cross-dimensional Operations
10. Intuitive Data Manipulation
11. Flexible Reporting
12. Unlimited Dimensions and Aggregation Levels

Multidimensional Conceptual View
A user-analyst's view of the enterprise's universe is multidimensional in nature. Accordingly, the user-analyst's conceptual view of OLAP models should be multidimensional in nature. This multidimensional conceptual schema or user view facilitates model design and analysis, as well as inter and intra dimensional calculations through a more intuitive analytical model. Accordingly user-analysts are able to manipulate such multidimensional data models more easily and intuitively than is the case with single dimensional models. For instance, the need to "slice and dice," or pivot and rotate consolidation paths within a model is common. Multidimensional models make these manipulations easily, whereas achieving a like result with older approaches requires significantly more time and effort.

Transparency
Whether OLAP is or is not part of the user's customary front-end (e. g., spreadsheet or graphics package) product, that fact should be transparent to the user. If OLAP is provided within the context of a client server architecture, then this fact should be transparent to the user-analyst as well. OLAP should be provided within the context of a true open systems architecture, allowing the analytical tool to be embedded anywhere the user-analyst desires, without adversely impacting the functionality of the host tool.

OLAP should be provided within the context of a true open systems architecture, allowing the analytical tool to be embedded anywhere the user-analyst desires, without adversely impacting the functionality of the host tool.

Transparency is crucial to preserving the user's existing productivity and proficiency with the customary front-end, providing the appropriate level of function, and assuring that needless complexity is in no way introduced or otherwise increased.

Additionally, it should be transparent to the user as to whether or not the enterprise data input to the OLAP tool comes from a homogenous or heterogeneous database environment.

Accessibility
The OLAP user-analyst must be able to perform analysis based upon a common conceptual schema composed of enterprise data in relational DBMS, as well as data under control of the old legacy DBMS, access methods, and other non-relational data stores at the same time as the basis of a common analytical model. That is to say that the OLAP tool must map its own logical schema to heterogeneous physical data stores, access the data, and perform any conversions necessary to present a single, coherent and consistent user view. Moreover, the tool and not the end-user analyst must be concerned about where or from which type of systems the physical data is actually coming. The OLAP system should access only the data actually required to perform the indicated analysis and not take the common “kitchen sink” approach which brings in unnecessary input.

The OLAP system should access only the data actually required to perform the indicated analysis and not take the common "kitchen sink" approach which brings in unnecessary input.

Consistent Reporting Performance
As the number of dimensions or the size of the database increases, the OLAP user-analyst should not perceive any significant degradation in reporting performance. Consistent reporting performance is critical to maintaining the ease-of-use and lack of complexity required in bringing OLAP to the end-user.

If the user-analyst were able to perceive any significant difference in reporting performance relating to the number of dimensions requested, there would very likely be compensating strategies developed, such as asking for information to be presented in ways other than those really desired. Spending one’s time in devising ways of circumventing the system in order to compensate for its inadequacies is not what end-user products are about.

It is imperative that the server component of OLAP tools be sufficiently intelligent such that various clients can be attached with minimum effort and integration programming.

Client-Server Architecture
Most data currently requiring on-line analytical processing is stored on mainframe systems and accessed via personal computers. It is therefore mandatory that the OLAP products be capable of operating in a client-server environment. To this end, it is imperative that the server component of OLAP tools be sufficiently intelligent such that various clients can be attached with minimum effort and integration programming.

The intelligent server must be capable of performing the mapping and consolidation between disparate logical and physical enterprise database schema necessary to effect transparency and to build a common conceptual, logical and physical schema.

Generic Dimensionality
Every data dimension must be equivalent in both its structure and operational capabilities. Additional operational capabilities may be granted to selected dimensions, but since dimensions are symmetric, a given additional function may be granted to any dimension. The basic data structure, formulae, and reporting formats should not be biased toward any one data dimension.

Dynamic Sparse Matrix Handling
The OLAP tools' physical schema must adapt fully to the specific analytical model being created to provide optimal sparse matrix handling. For any given sparse matrix, there exists one and only one optimum physical schema. This optimal schema provides both maximum memory efficiency and matrix operability unless of course, the entire data set can be cached in memory. The OLAP tool's basic physical data unit must be configurable to any subset of the available dimensions, in any order, for practical operations within large analytical models. The physical access methods must also be dynamically changeable and should contain different types of mechanisms such as:

1. direct calculation;
2. B-trees and derivatives,
3. hashing;
4. the ability to combine these techniques where advantageous.

Sparseness (missing cells as a percentage of possible cells) is but one of the characteristics of data distribution. The inability to adjust (morph) to the data set's data distribution can make fast, efficient operation unobtainable. If the OLAP tool cannot adjust according to the distribution of values of the data to be analyzed, models which appear to be practical, based upon the number of consolidation paths and dimensions, or the size of the enterprise source data, may be needlessly large and/or hopelessly slow in actuality. Access speed should be consistent regardless of the order of cell access and should remain fairly constant across models containing different numbers of data dimensions or varying sizes of data sets.

If the OLAP tool cannot adjust according to the distribution of values of the data to be analyzed, models which appear to be practical, based upon the number of consolidation paths and dimensions, or the size of the enterprise source data, may be needlessly large and/or hopelessly slow in actuality.

For example, given a set of input data from the enterprise database which is perfectly dense (every possible input combination contains a value, no nulls), it is possible to predict the size of the resulting data set after consolidation across all modeled data dimensions.

For example, in a particular five-dimensional analytical model, let us suppose that the physical schema size after model consolidation is two-and-one-half times the size of the input data from the enterprise database.

However, if the enterprise data is sparse, and has certain distribution characteristics, then the resulting physical schema might be one-hundred times the size of the enterprise data input. But, given the same size data set, and the same degree of sparseness, but with different data distribution, the size of the resulting physical schema might be only two-and-one-half times the size of the enterprise data input as in the case of the perfectly dense example. Or, we could experience anything in between these two extremes. "Eyeballing" the data in an attempt to form an educated guess is as hopeless as using conventional statistical analysis tools to obtain crosstabs of the data.

Because conventional statistical analysis tools always compare only one dimension against one other dimension, without regard for the other, perhaps numerous, data dimensions, they are unsuitable to multi-dimensional data analysis. Even if such tools could compare all dimensions at once (which they can't), the resulting crosstab would be the size of the product of all the data dimensions, which would be the maximum size of the physical schema itself.

> *OLAP tools can empower user-analysts to easily perform types of analysis which previously have been avoided because of their perceived complexity.*

By adapting its physical data schema to the specific analytical model, OLAP tools can empower user-analysts to easily perform types of analysis which previously have been avoided because of their perceived complexity. The extreme unpredictability and volatility in the behavior of multidimensional data models precludes the successful use of tools which rely upon a static physical schema and whose basic unit of data storage has fixed dimensionality (e.g., cell, record or two-dimensional sheet). A fixed, physical schema which is optimal for one analytical model, will typically be impractical for most others. Rather than basing a physical schema upon cells, records, two dimensional sheets, or some other similar structure, OLAP tools must dynamically adapt the model's physical schema to the indicated dimensionality and especially to the data distribution of each specific model.

Multi-User Support

Oftentimes, several user-analyst's have a requirement to work concurrently with either the same analytical model or to create different models from the same enterprise data. To be regarded as strategic, OLAP tools must provide concurrent access (retrieval and update), integrity, and security.

> *To be regarded as strategic, OLAP tools must provide concurrent access (retrieval and update), integrity, and security.*

Unrestricted Cross-Dimensional Operations

The various roll-up levels within consolidation paths, due to their inherent hierarchical nature, represent in outline form, the majority of 1:1, 1:M, and dependent relationships in an OLAP model or application. Accordingly, the tool itself should infer the associated calculations and not require the user-analyst to explicitly define these inherent calculations. Calculations not resulting from these inherent relationships require the definition of various formulae according to some language which of course must be computationally complete.

Such a language must allow calculation and data manipulation across any number of data dimensions and must not restrict or inhibit any relationship between data cells regardless of the number of common data attributes each cell contains.

For example, consider the difference between a single dimensional calculation and a cross-dimensional calculation. The single dimensional calculation: Contribution = Revenue -Variable Cost defines a relationship between attributes in only one data dimension, which we shall call *D_ACCOUNTS*. Upon calculation, what occurs is that the relationship is calculated for all cells of all data dimensions in the data model which possess the attribute *Contribution*.

A Complementary Approach

A cross-dimensional relationship and the associated calculations provide additional challenges. For example, given the following simple five-dimensional outline:

D [1] _Accounts

- Sales
- Overhead
- Interest Rate
- et cetera

D_Corporate

- United Kingdom
  - London
  - York
  - et cetera
- France
  - Paris
  - Cannes
  - et cetera

D_Fiscal Year

- Quarter1
  - January
  - February
  - March
- Quarter2
  - April
  - May
  - June
- et cetera

D_Products

- Audio
- Video
- et cetera

D_Scenario

- Budgeted
- Actual
- Variance
- et cetera

Sample Five-Dimensional Outline Structure

The formula to allocate corporate overhead to parts of the organization such as local offices (Paris, Cannes, et cetera) based upon their respective contributions to overall company sales might appear thus: *Overhead equals the percentage of total sales represented by the sales of each individual local office multiplied by total corporate overhead*

1 "D_" is used to indicate that this top most aggregation level is the dimension.

Here is another example of necessary cross-dimensional calculations. Suppose that the user-analyst desires to specify that for all French cities, the variable *Interest Rate* which is used in subsequent calculations, should be set to the value of the *BUDGETED MARCH INTEREST RATE* for the city of Paris for all months, across all data dimensions. Had the user-analyst not specified the city, month and scenario, the attributes would alter and stay consistent with the month attributes of the data cell being calculated when the analytical model is animated. The described calculation could be expressed as:

If *the value within the designated cell appears within the consolidation path D_Corporate, beneath the consolidation level France,* ***then*** *the global interest rate becomes the value of the interest rate for the month of March which is budgeted for the city of Paris*

Intuitive Data Manipulation
Consolidation path re-orientation, drilling down across columns or rows, zooming out, and other manipulation inherent in the consolidation path outlines should be accomplished via direct action upon the cells of the analytical model, and should neither require the use of a menu nor multiple trips across the user interface. The user-analyst's view of the dimensions defined in the analytical model should contain all information necessary to effect these inherent actions.

Flexible Reporting
Analysis and presentation of data is simpler when rows, columns, and cells of data which are to be visually compared are arranged in proximity or by some logical grouping occurring naturally in the enterprise. Reporting must be capable of presenting data to be synthesized, or information resulting from animation of the data model according to any possible orientation. This means that the rows, columns, or page headings must each be capable of containing/displaying from 0 to N dimensions each, where N is the number of dimensions in the entire analytical model.

Additionally, each dimension contained/displayed in one of these rows, columns, or page headings must itself be capable of containing/displaying any subset of the members, in any order, and provide a means of showing the inter-consolidation path relationships between the members of that subset such as indentation.

Unlimited Dimensions and Aggregation Levels
Research into the number of dimensions possibly required by analytical models indicates that as many as nineteen concurrent data dimensions (this was an actuarial model) may be needed. Thus the strong recommendation that any serious OLAP tool should be able to accommodate at least fifteen and preferably twenty data dimensions within a common analytical model.

Of the several products in the marketplace, the one that appears to be the most robust and, at the same time, able to withstand the tests against these prescribed criteria is Hyperion Essbase from Hyperion Solutions of Sunnyvale, California.

Furthermore, each of these generic dimensions must allow an essentially unlimited number of user-analyst defined aggregation levels within any given consolidation path.

Of the several products in the marketplace, the one that appears to be the most robust and, at the same time, able to withstand the tests against these prescribed criteria is Hyperion Essbase from Hyperion Solutions of Sunnyvale, California. In the next section Hyperion Essbase is described and evaluated according to the twelve OLAP rules.

A Product Evaluation

Hyperion Essbase
Hyperion's Essbase is a new approach to OLAP which complements database and spreadsheet technologies by providing the ability to discern relationships not expressed explicitly in the underlying database design, and by providing for the specification of additional consolidation paths as specified during the analysis session by the user-analyst.

Experienced users describe Hyperion Essbase as a very powerful tool, that is easy to use.

Hyperion Essbase addresses the majority of the features specified in this paper, especially complex data consolidation functions.

Multidimensional Conceptual View
The Hyperion Essbase product provides a multidimensional conceptual view to the user-analyst. This multidimensional conceptual view facilitates OLAP model design and analysis, as well as inter and intra dimensional calculations. The overwhelming majority of the type of intuitive data manipulation and calculation functions such as "slice and dice," or pivot and rotate consolidation paths are provided and work against both single and multiple dimensions.

Transparency
The look, feel, and functionality of the user-analyst's chosen interface to Hyperion Essbase is not adversely impacted by the presence of the Hyperion Essbase product. Hyperion Essbase provides additional menu items which are integrated with their customary human interface (e. g., spread-sheet, graphics tool, et cetera) and additional functionality is made available through the mouse or other pointing device.

Accessibility
Hyperion Essbase is able to map to, access, create and load a common conceptual, logical and physical schema from heterogeneous data stores within the enterprise while keeping this fact transparent to users.

Consistent-Reporting Performance
Hyperion Essbase reporting performed well (better than expected) across multiple models, with varying formats, numbers of dimensions and differing data distribution. Reports having essentially any layout are created with great ease. While there was some variance in the reporting performance, the perceived impact upon the user analyst was considered insignificant.

Client-Server Architecture
Hyperion Essbase appears to be a robust, true client-server facility which enables user-analysts to perform OLAP functions using common spreadsheets as interfaces to the multidimen-sional server. Limited Hyperion Essbase code is installed upon the client which intercepts requests for added functionality from the client interface and communicates with the server.

Generic Dimensionality
All dimensions in Hyperion Essbase are created equal at the primitive level. Manipulations and calculations which can be performed in any one dimension can be performed likewise within the context of any other dimension. User-analysts may elect to associate additional functionality with any dimension. User-analysts are free to create any dimensions bearing any desired characteristics which may participate in any formula or calculation.

Dynamic Sparse Matrix Handling
This is made available through a powerful server technology which can be tightly coupled with the enterprise database. Hyperion Essbase can adjust its physical schema for maximum performance based upon the density of the enterprise input data and the actual distribution of the data values.

Multi-User Support
This feature is provided by the software product but may be constrained by the number of clients which can be supported by the server hardware platform. Additionally, Hyperion Essbase provides multi-user access to a common analytical model and controls access and permissions down to the individual cell level through a menu-driven “grant“ and “revoke“ scheme.

Unrestricted Cross-Dimensional Operations
Any two (or more) cells, regardless of the dimension in which they appear within a given analytical model can be used in any formula. Formulas are not restricted to one dimension at a time. Any term in a formula can be further modified by including attributes from any dimension. In other words, any data cell or range in the data model can be accessed for calculation at any time.

Intuitive Data Manipulation
The majority of all model creation, manipulation, and animation tasks occur via mouse clicks directly associated with the data under analysis.

Flexible Reporting
Hyperion Essbase reporting capability goes significantly beyond that of the front-ends available for commercial RDBMS products. Queries and reports which would normally require significant and sophisticated SQL (and perhaps associated procedural coding effort) can be created easily with Hyperion Essbase and processed significantly faster. User-analysts are able to present multidimensional data in any format, according to any consolidation path, encompassing any number of data dimensions.

Unlimited Dimensions and Aggregation Levels
Hyperion Essbase allows the specification and manipulation of what is essentially an unlimited number of data dimensions within a given model.

Within a given dimension, Hyperion Essbase supports what is essentially an unlimited number of aggregation levels. However; some spreadsheet products which may serve as a front-end to the Hyperion Essbase product have a maximum number of cells which may appear in any spreadsheet, effectively limiting the amount of data which may be viewed within a given model at any one time.

Summary
While no currently available product provides all of the functionality desired for comprehensive OLAP support according to the four types of analytical models presented, Hyperion Essbase is immediately usable, intuitive, and very powerful; this product provides significant value today and shows great potential for the future.

Conclusion

Closing

As more and more organizations recognize the need and significant benefit of OLAP, the number of user-analysts will increase. Historically, a small number of experts in operations research have been responsible for performing this type of sophisticated analysis for business enterprises.

As organizations grow, and as the emphasis upon flexibility and competition is disbursed throughout the entire organizational structure, more and more individuals within business enterprises will be required to perform OLAP.

Whereas a small number of highly trained individuals were dedicated to the performance and communication of the results of business and strategic data analysis during the last decade, it is expected that the majority of managers within an enterprise will come to rely upon OLAP during the latter part of the 90's. OLAP has begun and will continue to permeate organizations at all levels, empowering managers to provide more timely strategic and tactical direction in accordance with the increasing number of internal and external factors impacting contemporary business enterprises. The quality of strategic business decisions made as a result of OLAP is significantly higher and more timely than those made traditionally.

Ultimately, an enterprise's ability to compete successfully and to grow and prosper will be in direct correlation to the quality, efficiency, effectiveness and pervasiveness of its OLAP capability. It is, therefore, incumbent upon IT organizations within enterprises of all sizes, to prepare for and to provide rigorous OLAP support for their organizations.

Hyperion Solutions Corporate Headquarters

Hyperion Solutions Corporation
1344 Crossman Avenue
Sunnyvale, CA 94089
408-744-9500
Fax 408-744-0400
info@hyperion.com
www.hyperion.com

European Headquarters

Hyperion Solutions Europe
Enterprise House
Greencourts Business Park
333 Styal Road
Manchester M22 5HY
United Kingdom
44-161-498-2200
Fax 44-161-498-2210

Asia Pacific Headquarters

Hyperion Solutions Asia Pte. Ltd.
#24-01 IBM Towers
80 Anson Road
Singapore 079907
65-323-3485
Fax 65-323-3486

131 9811