Patricia Seybold's

UNIX IN THE OFFICE

VOL. 4, NO. 5
ISSN: 0887-3054
MAY 1989

INSIDE

PRODUCTS • TRENDS • ISSUES • ANALYSIS

Sybase

The Next Generation

By Judith R. Davis

SYBASE FROM Sybase, Incorporated, is a hot contender these days in the relational database management system (RDBMS) market. Designed specifically to meet the needs of online transaction processing applications, Sybase has hit a nerve in the industry and caused a lot of people to sit up and take notice.

We still remember our first encounter with Sybase when it was announced two years ago. Robert Epstein, Sybase's executive vice president, forcefully declared that the database was "no longer just a spineless filing cabinet, a wimp" meekly accepting whatever data was *(continued on page 3)*

• E D I T O R I A L •

The Object Management Group

A Move towards Computing in the '90s

By Judith S. Hurwitz

JUST WHEN YOU thought it was safe to come out...another consortium is born. It would be easy to dismiss the Object Management Group (OMG) as just another group, but that would be a serious mistake. The Object Management Group has important implications both for the way we will be doing computing in the '90s and the slow pace of software innovation.

So what is OMG all about? Its core technology is Hewlett-Packard's NewWave, an object-oriented environment for "workplace automation." New applications will be able to be written in this object-oriented model, and older applications can be encapsulated to operate in this new environment. What good is object orientation? From a programming perspective, it allows code to be written in small modules. These modules can be linked together and can point to each other. Thus, one programmer could create a spreadsheet module containing the logic for adding rows of numbers together, and another, a module containing the formula for a regression analysis. The end user is the beneficiary of all this. The accountant, for example, creates a series of numbers and "moves" them inside the spreadsheet regression analysis module. Now these numbers have the benefit of the program that created them. The accountant simply had to know the module existed and could interact with the specific spreadsheet on a representational level.

OMG will take this concept and move it beyond the single accountant working at a PC or workstation. OMG's aim is to create a network version of this process for application-to-application communication based on the distributed computing model that is the underpinning for future networking. With the foundation in place, it will be easier for programmers and end users to take advantage of modules no matter where they exist or who created them. Some in the industry believe that, in time, we will have the equivalent of integrated circuits for software. We might some day see "software IC" factories churning out small, bite-sized chunks of code that can be used over and over again.

Why do we need the Object Management Group? It is simple. We can no longer assume that all the information we need to accomplish our jobs will be stored in our own PCs or even our own corporate databases. We need information from our customers and our suppliers. We are beginning to form more and more consortia and agreements among different organizations and companies across the United States and throughout the world. (Doesn't Chrysler sell Japanese cars under its own label?) Leveraged buyouts create huge companies with different computer systems and a huge variety of information types. Our corporations have given up requiring everyone to use one type of computer or one database or software tool.

To move to the software required for the '90s, we will need distributed object management. And even more importantly, we will need a standard way of communicating between objects. Significantly, all the founding members of the Object Management Group are planning to implement this new foundation technology under Unix. It is yet another indication that innovation may well shift to Unix from proprietary and DOS.

The OMG is just beginning. We applaud as pioneers the founding members: Canon, Data General, Gold Hill Computers, Hewlett-Packard, Phillips, Prime, Soft•Switch, Sun Microsystems, Unisys, 3Com, and American Airlines. We encourage other hardware, software, and end-user organizations to join the effort. While the core technology will be NewWave, each member will submit technologies that can be used by all other members.

These founding members come from both factions of the Unix community: HP and Phillips are key sponsors of OSF, while Prime, Sun, and Unisys are sponsors of Unix International. Efforts that bring different factions together are critical if Unix and open systems are to move into the '90s. ◐

Patricia Seybold's Office Computing Group

Publisher PATRICIA B. SEYBOLD
Editor-in-Chief JUDITH S. HURWITZ
Managing Editor RONNI T. MARSHAK
News Editor DAVID S. MARSHAK
Senior Editors MICHAEL D. MILLIKIN, JUDITH R. DAVIS Telephone: (617) 861-3926
Contributing Editor GARY J. NUTT Telephone: (303) 444-0341
Associate Editors JOHN N. KELLY, JOHN R. RYMER, LAURE BROWN
Sales Director RICHARD ALLSBROOK JR.
Circulation Manager DEBORAH A. HAY
Customer Service Manager ANNE L. WORKMAN

148 State Street, Suite 612, Boston, Massachusetts 02109 Telephone: (617) 742-5200 FAX: (617) 742-1028

Patricia Seybold's UNIX in the Office (ISSN 0887-3054) is published monthly for $495 (US), $507 (Canada), and $519 (Foreign) per year by Patricia Seybold's Office Computing Group, 148 State Street, Suite 612, Boston, MA 02109. Application to mail at second-class postage rates is pending at Boston, MA and additional mailing offices. POSTMASTER: Send address changes to *Patricia Seybold's UNIX in the Office*, 148 State Street, Suite 612, Boston, MA 02109.

•SYBASE•

(continued from page 1) thrown its way. Sybase enabled the database to stand up and say, "No. You can't do that." The objective was to effectively enforce an organization's business rules while providing high performance, round-the-clock availability, and dramatic improvements in developer productivity. The image stuck, and we were impressed with both the design of the product and the appeal of the company's marketing strategy.

Although the competition initially downplayed Sybase's potential impact on the market, saying that the company started too late, etc., the evidence is clear: There was pent-up demand for what Sybase had to offer. Take a look at the financial indicators. Company revenues have grown at a phenomenal rate since Sybase was commercially introduced. Although Sybase is privately held and does not report financial results, estimates from a reliable source look like this: Revenues of $6 million in fiscal year 1987, $24 million in 1988, and projections of $55 million plus in 1989. This means annual revenue increases of 300 percent and 129 percent respectively. While this is certainly a growing market, Sybase is still considered a relatively new kid on the block, competing with established RDBMS vendors such as Oracle, Relational Technology, and Informix Software.

In this issue, we take an in-depth look at the Sybase architecture and provide some user and developer perspective as well.

SOME BACKGROUND. Prior to starting Sybase in 1984 with venture capital backing, the company's founders had been involved in the development and marketing of several leading relational products, including the Ingres research project at the University of California at Berkeley, IBM's DB2, and Britton Lee's IDM. The objective for the Sybase effort was to address the shortcomings of existing RDBMS products for handling online applications.

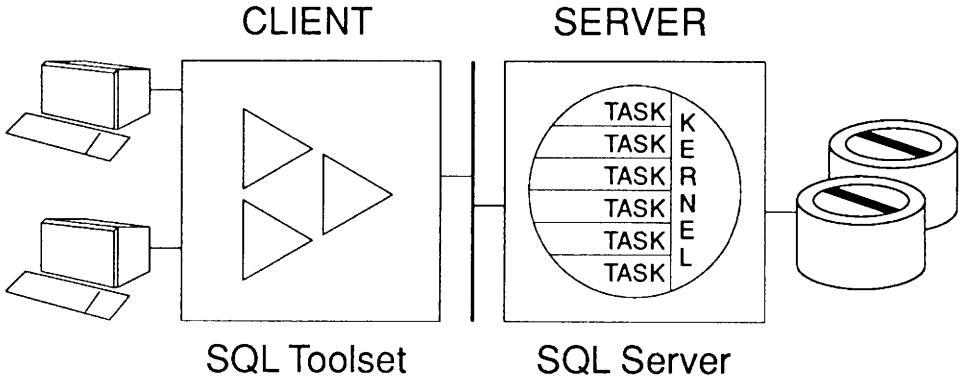
The primary issue was to design a relational database that would provide more than just a decision support system. The underlying problems fell into three critical areas: performance, availability, and database integrity.

The architecture of traditional RDBMSs simply was not adequate for online applications, those mission-critical, transaction-oriented systems supporting many concurrent users and often accessing very large databases. RDBMSs were generally single-threaded products that accessed the database using interpretive rather than compiled syntax. It was also necessary to bring the entire system down in order to backup or modify the database. All of this hampered performance.

In terms of integrity, the traditional RDBMS was, in fact, a spineless filing cabinet. It handled the storage and retrieval of data, but filtering or evaluation to assess data validity was left to the front-end application. This can create problems in a networked environment, a characteristic of many online applications.

The company did its homework well and accurately assessed some important industry trends. As a result, Sybase is designed for both online and decision-support applications. It is relational, SQL-based, and designed from the start as a client/server DBMS. The server, called the SQL Server, is multithreaded and provides integrated support for both networking

The Sybase Architecture and Platforms



Client support *	Network support	Server support
• VAX/VMS	• DECnet	• VAX/VMS (incl. clusters)
• Sun/Unix	• TCP/IP	• Sun/Unix
• Stratus/Vos	• LAN Manager	• Stratus/Vos
• Pyramid/Unix	• PC LANs	• Pyramid/Unix
• OS/2		• OS/2
• DOS		
• IBM RT/AIX		• IBM RT/AIX
• NeXT/Unix		• NeXT/Unix
• HP/Unix **		• HP/Unix **
• AT&T/Unix		• AT&T/Unix
• Apollo/Unix		
• Macintosh/HyperCard		

* APT Workbench is currently available on VAX/VMS and Sun/Unix.

** In beta test

The Sybase client/server architecture is multithreaded and optimized to run on a network. The platforms listed below the dotted lines are in development and not yet available.

and distributed databases. The server is also intelligent, enforcing integrity constraints from within the database itself. As Bob Epstein stated recently, "Networking has been around a long time. But a meaningful client/server architecture requires intelligence in the server itself. This is the difference between having a server and having a mere database on the network."

Sybase employs over 350 people (double the number from last year) and has 18 U.S. sales offices. International operations include both wholly owned subsidiaries (Canada, United Kingdom, France, Germany, and Australia) and distributors.

PRODUCT HISTORY. Sybase originally ran on only three platforms: DEC VAX/VMS, Sun/Unix workstations, and Pyramid 98x/Unix. Recognizing that it could not be all things to all people, Sybase took a sharply focused approach to operating platforms, particularly for the SQL Server.

Sybase currently runs on five server platforms and six front-end platforms (see page 3), with more to come. The company has no immediate plans in the IBM MVS/DB2 environment other than gateways. It will continue to add Unix platforms very carefully, ensuring that it covers a broad price/performance range as well as platforms that are important internationally. Thus, the company is keeping a close eye on its ports and will move ahead as business conditions dictate.

Sybase is no longer just a database engine company. It is investing as much effort in its tools as in its server.

The company acknowledges two things here: strong pressure from customers to port to additional systems, and the truth in Oracle's contention that portability across multiple platforms really does matter. We expect to see Sybase on many more platforms over the next couple of years.

Another initial negative for Sybase was the lack of applications development tools. As Epstein says, "Our image has been one of a server and not a tools company." This, too, has been remedied with the recent introduction of APT Workbench.

The next version of Sybase is in beta test and will be released shortly. We discuss the coming attractions more fully under "Future Directions" below.

MARKETING STRATEGY. The company markets to corporate and government end users, OEMs, and VARs. Its two strongest target industries are financial services and government/aerospace. Right behind are manufacturing/distribution and telecommunications. These four markets comprise about 80 percent of Sybase's business. Other up-and-coming segments include medical/health care, electronic publishing/information services, and transportation.

Sybase has an OEM agreement with Stratus Computer and joint marketing agreements with other platform vendors. The number of VARs working with Sybase is more than 25 and growing.

Since controlled shipments began in October, 1986, Sybase has over 400 customers and over 500 sites worldwide. The company expects to add 600 new customer sites this fiscal year, bringing its sites to over 1,000.

Software Partners. Sybase has also put much effort into developing a successful software partners program. These partners provide a wide variety of additional tools that can access the Sybase SQL Server (see page 14). Sybase views working with these independent vendors as important for two reasons. Again, Sybase recognizes that it cannot provide all things to all customers (Stewart Schuster, vice president of marketing, calls this the company's "religion"). In addition, software partners expand the Sybase environment, allowing the Sybase server to run with other front-end tools as well as providing potential links with other databases.

Sybase is no longer just a database engine company. It is investing as much effort in its tools as in its server. The company is willing to share the tools business with its partners, but is not willing to share the server business without a fight (except with IBM and DB2).

OS/2. Early in 1988, Sybase jointly announced, with Microsoft and Ashton-Tate, the Sybase SQL Server on OS/2. SQL Server will be marketed by all three companies when it becomes available shortly (network developer kits have already been distributed).

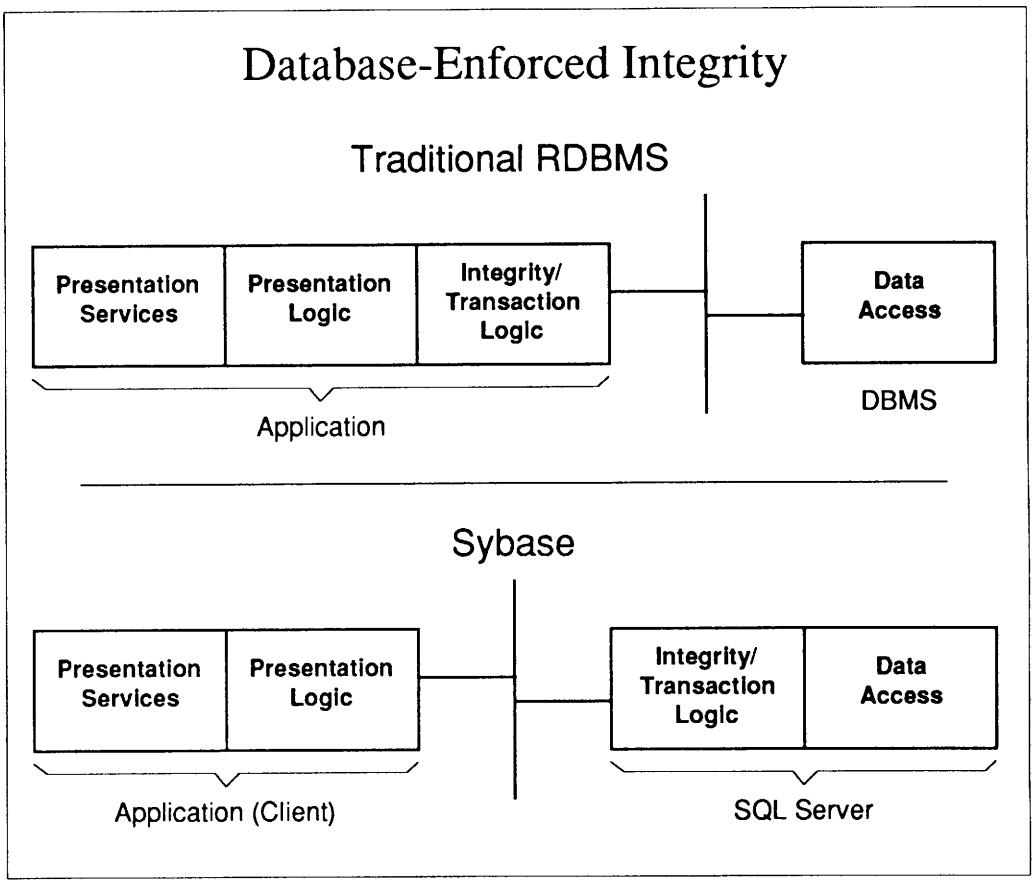
SQL Server will provide a database foundation in the OS/2 environment for distributed services and data management. This is essential for the true implementation of distributed network computing on the PC LAN platform. Developers and users alike are drooling over this opportunity.

NeXT. Another interesting marketing approach is the agreement with Steve Jobs's NeXT, Incorporated, to bundle the Sybase SQL Server with every NeXT computer system. NeXT's marketing agreement with Businessland provides a potential retail outlet for Sybase.

Architecture

Sybase is optimized to run on a network of servers and workstations. Its client/server architecture splits the DBMS into a front end (the SQL Toolset) and a back end (the SQL Server). The server handles data management functions for all users on the system. The SQL Toolset provides the user interface and a set of window-based tools for building and running applications. The two components can run concurrently on a single computer or independently on different machines. Moving from a single machine to a network does not require rewriting applications or

retraining users (Sybase uses network communications between the server and front end even when both are on the same machine), and applications run interchangeably on character terminals and bit-mapped workstations.



Unlike many of its competitors, which enforce database integrity in the front end, Sybase stores and enforces integrity constraints in the server. This implements an important aspect of the Sybase definition of client/server: Business rules and integrity constraints must reside with the data in the server.

SQL SERVER. One of the unique aspects of Sybase is the fact that the SQL Server runs as a single program. Sybase is essentially a "database operating system" on top of, and independent of, the underlying operating system. Eliminating most operating system overhead accounts for a large measure of the increased performance Sybase provides. Sybase itself controls and optimizes such multiuser functions as opening and closing files, scheduling concurrent users against a database, task-switching, disk-caching, indexing, transaction processing, and locking. Sybase only uses the operating system to physically drive the disks and manage the network. A mere 33K of memory per user is required to manage this single SQL Server process (e.g., Sybase can handle 30 users with 1MB of memory).

Sybase is also multithreaded. It can optimize and process several SQL statements at the same time, including transaction locking and input/output.

BUILT-IN DATA INTEGRITY. A key feature of Sybase is the ability to define and store all data integrity rules, including referential integrity, as procedures in the data dictionary. Integrity is enforced centrally in the SQL Server, rather than in each application program, and no application or user can circumvent the rules. Developers can prototype and write applications with less code, and changes to integrity rules only need to be made in one place. The database administrator is no longer dependent on the applications development staff to maintain data integrity.

This issue is particularly important in a networked environment. Here, it is much more difficult to control what version of a program runs on a workstation, or to ensure that every developer understands the rules for protecting the database from corruption and inconsistencies.

HIGH AVAILABILITY. Unlike other relational databases, Sybase allows all backups, recovery, database design changes, and other system maintenance to be performed while applications are running. This is a requirement for many online systems.

On some platforms, Sybase also provides software-based fault tolerance enforced through the RDBMS itself, rather than through the operating system. Copies of the database and transaction log are maintained on two different disks. Sybase can also mirror a single database rather than the contents of an entire disk, which most other mirroring systems require. This provides optimum use of disk space.

DISTRIBUTED DATABASES. Most RDBMS vendors are developing distributed database capabilities, and have initially implemented the ability to retrieve data from multiple databases on multiple machines in a single query. Sybase also provides the ability to update multiple databases in a single transaction through a "two-phase commit" protocol. This ensures that all updates at multiple locations have been completed before committing the entire transaction. If failures or errors occur, the effects of partially completed transactions will be

automatically backed out, leaving the databases in a consistent state.

DATABASE OBJECTS. While Sybase is not an object-oriented database, it implements some of the concepts of the object-oriented model. Objects in Sybase, for example, can have both properties and processing associated with them. Properties and processing can be inherited in some cases. Database objects (databases, tables, views, indexes, data types, defaults, rules, triggers, and stored procedures) are physically stored in the database as part of the data dictionary. Reports are also stored in the database as stored procedures. However, forms are stored externally in operating system files. It would certainly make sense, from the perspective of increased consistency and control, to incorporate forms as database objects in the future. The company plans to do this.

The ability to create a stored procedure is one of the most important and powerful constructs in Sybase. Stored procedures are a key to Sybase's performance.

Transact-SQL. Transact-SQL, the Sybase version of SQL, is the language used to create and manipulate database objects. It includes ANSI-standard SQL statements and extensions to SQL, such as commands to create user-defined data types, rules, defaults, triggers, and stored procedures; control-of-flow language for if/then, do while, go to, and the ability to define local variables; and other functions not found in SQL.

Stored Procedures. The ability to create a stored procedure is one of the most important and powerful constructs in Sybase. Stored procedures are written in Transact-SQL and combine SQL statements with flow-control language.

Stored procedures provide many benefits and are a key to Sybase's performance. Stored procedures are precompiled, minimizing execution time over SQL statements (either alone or in batches) submitted by the front-end application and interpreted each time by the server. The first time a stored procedure is run, the server prepares an execution plan and stores this in the database. Subsequently, the procedure is executed according to the stored plan. The most commonly used stored procedures are also cached in memory in a procedure buffer, eliminating disk I/O.

Because stored procedures are part of the database, they reduce network traffic, which also improves performance. A single "execute stored procedure" statement can be submitted to the server from the front-end application, yet the stored procedure can contain multiple SQL statements. In addition, a single stored procedure can be used by multiple applications. This reduces application development time and eases maintenance since a stored procedure is modified in one place.

Stored procedures also provide extensive flexibility. The developer can define parameters to be supplied at run-time when the stored procedure is executed. A stored procedure can return to the calling program both parameters and status codes to identify the success or failure of its execution (and the reason for failure). Stored procedures can call other procedures. They can also be executed on remote SQL Servers. For example, you could write triggers on a local SQL Server that execute procedures on a remote server whenever certain events take place locally.

System Procedures. Sybase provides an extensive set of predefined stored procedures, called system procedures, that can be used for a variety of functions (e.g., adding users and logins, binding rules and defaults to columns, and defining user data types).

Triggers. A trigger is a special kind of stored procedure often used to protect referential (cross-table) integrity. Triggers can be used to cascade changes through related tables in the database (e.g., making sure primary and foreign keys are consistent), and to disallow changes that would violate referential integrity (e.g., deleting a customer master record if the customer has orders outstanding).

Each table can have one trigger associated with each data manipulation statement: insert, update, and delete. Triggers can also be used beyond just referential integrity to update summary data, perform more complex field validation than that permitted in rules, and other functions.

Triggers can call local or remote stored procedures, can cause other triggers to execute if maintenance operations are performed on other tables (although a trigger cannot cause itself to fire recursively), and can be nested 16 deep.

Sybase is considering adding in the future the ability to define a trigger that fires when a Select statement is executed. In a distributed environment, for example, a Select trigger could be used to retrieve up-to-date data dictionary information necessary to optimize a distributed query. You could also use a Select trigger to generate audit trails, identifying who is accessing data or writing certain kinds of queries.

Rules. Rules are used to validate data entered in a field. You first create the rule using a Transact-SQL statement, and then bind the rule to a column in a table. Rules are applied any time data is inserted or updated in the table. Rules are limited to defining a set of values for a field entry, and cannot reference other column values or information in other database objects. Therefore, rules cannot be used to do cross-table field validation (this is done instead with triggers). A rule can be bound to multiple columns, but each column can only have one rule.

Defaults. Defaults are used to ensure that a value is supplied for every column requiring one. A default is only applied on

insert, and then only if the field is left blank. It provides protection against a user or an application failing to enter a value in a field defined as "not null." A default can be linked to a specific field or to a data type. When linked to a data type, any field defined with that data type inherits the default. Defining a default is very similar to defining a rule.

SQL TOOLSET. The Sybase SQL Toolset is the front-end client software, and it includes the Data Workbench, APT Workbench, and DB-Library. (For those who are wondering, APT stands for "application productivity tools.") The Data Workbench and APT Workbench are visually oriented tools for decision support and applications development. They use a visual "point and pick" style of interface with overlapping windows and pull-down menus.

Data Workbench. The Data Workbench is designed to accommodate all levels of users—developer, administrator, and end user. It includes facilities for creating tables, queries, and reports, and for doing some data entry. Queries can be generated using the SQL editor or a point-and-pick Visual Query Language (VQL). VQL allows the user to compose and run SQL queries without any knowledge of SQL syntax (we like that—no end user really wants to write SQL queries). The Data Workbench also includes database administration utilities and a clipboard function to transfer queries among tools.

The Data Workbench is used by developers as an interface for developing, testing, and debugging Transact-SQL statements, and for creating stored procedures, rules, triggers, etc. These can then be incorporated into applications designed with APT Workbench.

APT Workbench. APT Workbench is a new component that extends APT-Edit, the original Sybase module for designing forms, into a full-fledged applications development environment. APT Workbench adds a 4GL applications development language (APT-SQL) and an applications generator (APT-Build). Previously, Sybase developers could design forms with APT-Edit, but the rest of the application had to be written in a 3GL. This lack of a 4GL was hurting Sybase competitively. Announced in late 1988, this product has just started shipping for Sun and Digital's VMS. Other platforms will follow.

Sybase's tools are primarily targeted at the applications developer, not the end user. According to Bob Epstein, database end-user development tools should and will be offered by vendors such as Microsoft and Lotus, since the user may want to view the database in a spreadsheet or some other familiar format.

APT-SQL. Like Transact-SQL, APT-SQL also deals with objects, but they are application objects such as forms and menus rather than database objects. Both languages are similar in construction and syntax. It is not so much that they are different languages, but that they are designed and optimized to do different things. Another way to think about it is to associate APT-SQL with the context around the database for a particular application, and Transact-SQL with the actual content of the database.

Like stored procedures, APT-SQL procedures are also precompiled, but they are stored on the front end. Eventually, the company plans to store APT procedures in the data dictionary as well. This would make the management and tracking of APT procedures much easier.

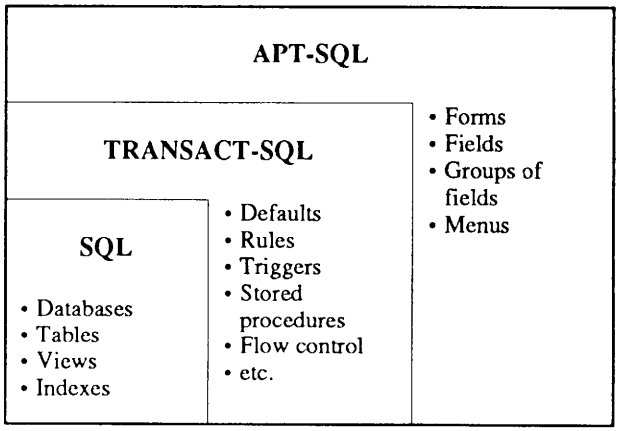
A Sample Rule

```
create rule credit_rule
as
@credit_limit <= 5000

sp_bindrule credit_rule, account.credit_limit
```

The first statement creates a rule, called credit_rule, which only allows values in the credit_limit field that are less than or equal to $5,000. The second statement uses a system stored procedure (sp_bindrule) to associate the credit_rule with the credit_limit field in the account table.

The Sybase Application Development Language



Transact-SQL includes ANSI-standard SQL statements plus Sybase extensions to SQL. It deals with database objects, and it executes in the back end (server). APT-SQL is a further extension of the language to create and manipulate application objects such as forms and menus. APT-SQL executes in the front end. APT-SQL procedures can include Transact-SQL statements or calls to stored procedures (which are submitted to the server for processing).

DB-Library. DB-Library consists of a host language interface for C, Cobol, Pascal, Fortran, and Ada, and networking support for DECnet, TCP/IP, and LAN Manager.

APT-Library. Like DB-Library, APT-Library is a call-level interface allowing 3GL programs to access APT-SQL procedures and forms.

FUTURE DIRECTIONS. Version 4.0 of the Sybase SQL Server, soon to be announced, will introduce a number of major enhancements to the product. The bottom line is to further implement the Sybase goal of an open, adaptable client/server architecture.

Server-To-Server Communications. Version 4.0 will extend the Sybase networking to communications between servers where servers can act as clients to other servers. This server-to-server remote procedure call (RPC) will provide more transparency for the user than the current system. Currently, the client must be logged on to a remote SQL Server in order to access data and stored procedures there (stored procedures always reside with the data they access), and must include the name of the remote server in the command. Now the local SQL Server will take care of the communications with the remote server.

Open Server Technology. Sybase will be providing a toolkit to enable developers to build their own servers, or gateways, to both data services and applications services. According to Bob Epstein, having an open system means "supplying enough technology to your competitors so they can implement your system." Sybase will provide both specs and library routines for its Open Server. The Sybase Open Server will be marketed as open software, licensing both source and object code, and allowing third parties to resell these licenses. The company also plans to offer open access to DB-Library, enabling users to tailor the front end as well.

This strategy is fundamentally different from that undertaken by competitors such as Relational Technology. The customer/developer will be able to integrate Sybase with other databases and with non-SQL systems, such as electronic mail, and not have to wait for Sybase itself to develop the integration. The Open Server is the foundation on which Sybase will build its own gateways in the future.

New Functions and Data Types. New text and image data types will be added, as well as many new mathematical and string functions.

PRESENTATION INDEPENDENCE. Sybase has been working on developing independence at the presentation level for the past 18 months. This facility will debut later this year as the Adaptable Windowing Environment (AWE). A Sybase application run on a particular platform will automatically emulate the look and feel of the interface native to that platform.

MULTIPROCESSORS. By the end of this year, Sybase will introduce the ability to take advantage of all processors in an SMP (symmetric multiprocessor) architecture. This capability is already written and demonstrable, and it will be attractive for customers with very large, high-performance applications.

THE END USER'S PERSPECTIVE

We reviewed a beta version of Sybase Release 4.0 running on a Sun 3/60 workstation under Unix.

User Interface

Sybase provides a consistent, menu-driven user interface, displaying on most screens a set of standard icons for scrolling, exit, tools, and help.

The Tools menu provides access to the data dictionary, a history listing (the last 20 SQL statements executed during the current session), the clipboard, and options for customizing the Data Workbench. We found the data dictionary browser very helpful.

NAVIGATION. It is very easy to move around in Sybase using the mouse on the Sun workstation. Navigating with the keyboard is a little trickier. Keyboard users move through menu options (and the "tools" icon) in a circular fashion with the left and right arrow keys. The Help and Exit options are accessed with function keys.

Sybase provides a consistent, menu-driven user interface, displaying on most screens a set of standard icons for scrolling, exit, tools, and help.

On the keyboard, the up and down arrow keys don't work in the familiar way. Instead of moving the cursor or scrolling information, they select the appropriate symbol in the group of scrolling icons, and the Return key actually invokes the scrolling by screen. This is confusing and requires extra keystrokes to accomplish what should be a very simple navigational task. In Command mode, the arrow keys do operate as expected. However, Command mode only applies for a few operations, requiring the user to turn it on again. We would like the option to keep Command mode on.

Sybase does not currently support the use of standard function keys. However, Sybase defines three permanent keys on the keyboard (Command mode, Help, and Exit) regardless of what window you are in, and one window-dependent soft key. We would like to see active function keys displayed on the screen (with an option to toggle off and on). We would also like the option to define more function keys (coming in a future release).

Sybase supports a wide variety of control key sequences, but they do not always work where you might intuitively expect them to. After retrieving a set of records using query-by-example, we tried to browse through the records using "^N" and "^P" for next/previous record. No such luck. We had to use the menu for this, a two- (or more) step process.

Sybase was designed and developed on a bit-mapped workstation. It has perhaps suffered from a fascination with the graphical and a lack of attention to the needs of the many character-based terminal users out there.

HELP. Help is context sensitive, and related topics are highlighted in reverse video within the Help text. The user can also get an index listing. This is not an alpha list of help topics, unfortunately, but a hierarchical list that follows the menu structure (Sybase calls this "in functional order"). To find a topic, you must browse the list if you don't know where it fits in the menu structure. Other options are a list of control/function keys and printing the Help text.

You can only scroll Help text by screen, but the text does overlap by one line when scrolling between screens.

DOCUMENTATION. The Sybase documentation is excellent, and the new three-ring-binder format is attractively designed. The manuals are clearly written, with adequate white space and tons of examples and screen illustrations, making it easy to follow the text on the system. A refreshing sense of humor also comes through here and there.

There are several manuals, but this is not particularly intimidating since all are clearly targeted. One thing missing in most of the manuals is tabs to identify sections for fast reference. Other potential enhancements include more use of color, heavier stock paper so the reverse page doesn't show through, and higher quality printing.

Creating a Database

Sybase provides three system databases. The Master database keeps track of the entire Sybase installation, including all other databases, valid logins, and physical devices. This information is stored in data dictionary tables. In the future, the Master database will also contain information on other servers with which its SQL Server can communicate.

The Model database provides an initial template for a new database. The Temporary database stores any temporary tables that a user or application creates.

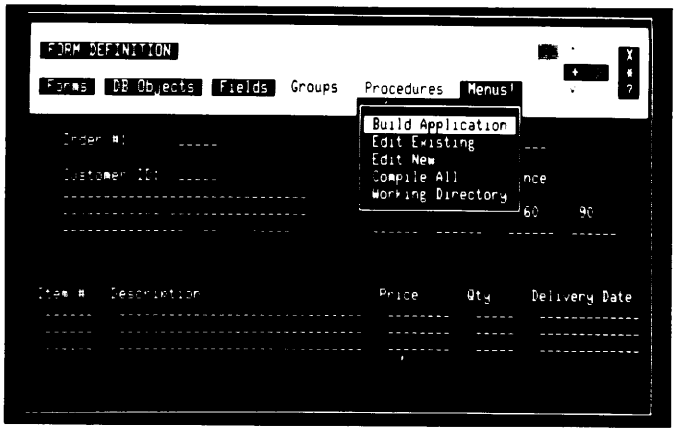
CREATING TABLES. Tables are created using the standard SQL "create table" command in the Data Workbench SQL editor. There is no menu-driven "fill in the form" way to define the fields in a table, which would be a nice feature for the end user. Using SQL means remembering not only the correct syntax, but also all of the options and parameters available for data types.

DATA TYPES. Sybase currently supports 11 different data types, including three types of integers, float, char and varchar (fixed- or variable-length character field up to 255 characters), binary and varbinary (fixed- or variable-length binary field up to 255 characters), bit (0 or 1, used for true/false or yes/no types of data), money, datetime, and timestamp.

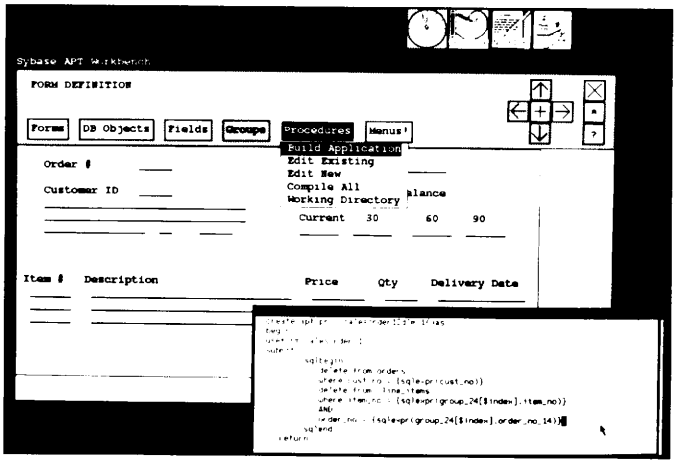
Coming in the next version of Sybase are variable-length text and image data types, each able to hold up to 2 billion bytes.

User Data Types. Sybase also allows the creation of user-defined data types, which are defined in terms of existing system data types. For example, you cannot define a latitude/longitude data type composed of degrees and radians because Sybase doesn't understand these concepts.

User-defined data types can be helpful for field definitions that are used repeatedly, such as a part number. Defining a new data type can save steps when creating a new table and provide



This is how Sybase looks on a bit-mapped Sun workstation.



This is the same screen on a character-based terminal. You can see how closely Sybase has mapped the two interfaces so that the look-and-feel is similar despite the terminal's functional limitations.

Sybase Features Chart

Architecture	Client/server
Underlying file structure	Proprietary
Database parameters	
Databases/server	32,767
Tables/database	2 billion
Records/database	Unlimited
Fields/record	250
Record size	2K
User interface	
Menu bypass	No
Contextual help	Yes
Tutorial	No; demo database followed in documentation
Ability to customize standard menus	No
Data types	11 supported currently
Character	Yes; char for fixed length and varchar for variable length; 255 character max.
Integer	Yes; int, smallint, tinyint
Float	Yes
Currency	Yes (money)
Date/time	Yes
Binary	Yes; binary for fixed length and varbinary for variable length; 255 character max.
Long text	Coming with Version 4.0; 2 billion char max.
Image	Coming with Version 4.0; 2 billion char max.
Support for arrays	No
Screen forms	Built in APT Workbench
Default form generator	Yes
Customized	Yes
Multiple tables/form	Yes
Multiple screens/form	Yes
Embedded processing (if-then-else, display aggregates)	Yes
Field attributes on forms	
Case conversion	Yes
Default value	Yes
Required field	Yes
Acceptable values	Yes
Verification (enter data twice)	No

consistency across multiple tables using the same field definition.

Sybase supports the concept of nulls, or unknown values, and any field can be defined as accepting nulls or not. The default is "not null."

MODIFYING TABLES. It is not possible to change the structure of a table after it is defined except to add fields with the "alter table" command. This is standard for most of the RDBMSs we have reviewed, although some provide additional options.

In Sybase, if you want to change a field's data type or length, delete a field, or change whether nulls are allowed, you must unload the data, drop the table, redefine it, and load the data back in. Reasons for this are: the data type relates to the physical storage of the data, deleting columns can invalidate applications, and deleting data in a large database can require significant overhead.

However, our frustration level is high here, since, as anyone who has ever tried to design a database knows, you never get it right the first time. And end users do not want to deal with loading and unloading data. Attracting end users as future application developers requires an easier way to modify table structure. Sybase does ease the pain somewhat with a stored procedure for renaming database objects, including tables and fields.

DATA ENTRY. The Data Workbench has a "quick and dirty" data entry facility, using a typical default format (fields listed vertically in the order defined, with field names as prompts). Nothing fancy. Just the basics. You can only enter data into one table at a time. However, triggers and rules still enforce data integrity. You can also edit, find records with a query-by-example approach, and print records.

The data entry facility is primarily designed for use by developers and prototypers. For end users, custom forms and menus for entering and updating data can be defined with a wide range of functionality and flexibility using APT Workbench.

Sybase Features Chart

Field attributes on forms (continued)	
Formatting of data	Yes
Calculated fields	Yes
Display only (no entry/update)	Yes (read only)
Hidden	Yes
Prompt (for data entry)	Yes
Error message	Yes
Customized help	Yes
Video display	Yes (reverse video)
Query-By-Forms	
Exact match	Yes
Relational operators	Yes
Ranges	Yes
List of values	No
Wildcards	Yes
Maximum/minimum values	No
Print query results	Yes
Pass results to report writer	Yes
SQL	
Standard SQL statements	
Data definition language (DDL)	Yes
Data manipulation language (DML)	Yes
Query language	Yes
Extensions to SQL	TRANSACT-SQL
Commit/rollback transactions	Yes
Execute operating system commands	No (yes in APT-SQL)
Load/unload data to/from ASCII file	Yes
Additional data definition statements	Yes
Control-of-flow logic	Yes
Can be embedded in C/Cobol programs	Yes
Can create new table with query results	Yes
Stored queries	Yes
Case-insensitive (e.g., field names)	No
Optimizer	Yes
Can call C routines	No (yes in APT-SQL)
How create SQL queries/statements	Interactive SQL editor
Report writer	Included in Data Workbench
Nonprocedural	Yes
Default report generator	Yes
Interactive report generator using screen forms	Yes
Interactive debugging	Yes
Input source	Query only (SQL or VQL)
Multiple tables	Yes
Page formatting	Yes
Headers and footers	Yes

INDEXING. Sybase supports B-tree indexing only. One special flavor of a B-tree index, called a clustered index, physically stores the data in sorted order to improve performance. A table can have one clustered index and up to 250 nonclustered indexes. Indexes can be defined as unique and can be dynamically created and dropped.

QUERIES. Queries are generated using VQL (the point-and-pick approach) or the SQL editor (to write SQL statements directly or recall stored queries). A third option is building a custom form with query-by-example option.

VQL. The Visual Query Language facility is well implemented, menu driven, and easy to use. To query a table, the user selects the table name, and a list of the fields is then displayed in the top window on the screen. In the lower window, Sybase automatically creates the actual SQL syntax for the query and dynamically edits it as the user changes it with menu options. This is a nice touch for training users in SQL in a relatively painless fashion. Menu options exist for creating expressions, aggregates, and type conversions; and for defining joins, field lists, sort orders, and group-by clauses.

VQL is used for simple Select statements (queries cannot be nested). It cannot be used to insert, update, or delete data, nor to save the results of a query to another table. VQL is designed as an ad hoc query tool, not as a way to create database objects.

When you select two or more tables for a query, Sybase can identify the appropriate join conditions if the names of the fields for the join are identical in each table. Otherwise, you tell the system how to perform the join. Sybase supports an outer join.

You cannot save a query as constructed in VQL. You can only save the SQL equivalent. When recalling the query, you edit it using the SQL editor.

With VQL's extensive formatting options, it is easy to add or delete columns, change the order of the columns, and change the properties associated

Sybase Features Chart

Report writer (continued)	
Data formatting	Yes
Sort data	Yes, in query
Aggregate functions	Yes
Logical processing (if-then-else logic)	No (yes in APT-SQL)
User variables	Yes
Prompt for input variables at run-time	Yes
B-tree indexing	Yes
Maximum number of indexes	251/table
Maximum number of fields/index	16
Maximum size of index key	256 characters
Order options	Ascending only (default)
Unique index	Yes
Clustered index	Yes (one/table)
Other file access methods (hash, etc.)	None
Database security	
Login password	Yes
Database-level access	Yes
Table-level access	Yes
Record-level access	Yes
Field-level access	Yes
Access by time of day	No
Access by location (workstation)	No
Application generator	APT Workbench/APT-SQL
Ability to design application menus	Yes
Default menu generator	Yes
Custom help	Yes
Ability to create views	Yes
Transactions	
Logging	Yes
Commit/rollback transaction	Yes
Roll forward	Yes

with both the heading and the data in each column. This formatting is reflected in the SQL equivalent as well.

SQL. The SQL editor is accessed directly from the operating system or from within Data Workbench. The menus allow you to create, run, and debug SQL statements interactively.

REPORTS. Like VQL, the report writer uses a visual approach and is very easy to use. A query provides input for the report. You can create a query within the report writer, using VQL or SQL, if necessary. One report can include multiple queries. Capabilities include defining report properties, report and page headers/footers, control breaks, and extensive properties for each variable on the report.

You can specify embedded parameters in a report query, and the user will be prompted at run-time for parameter values.

FORMS AND APPLICATIONS. APT Workbench, the Sybase application development environment, is forms based and event driven. Events include entering/exiting a field or form, selecting a menu item, etc. Each event can call another form, an APT-SQL routine, or a 3GL program. Each APT Workbench object (a form, field, group of fields, or menu option) has both properties and processing associated with it. Processing is also broken down into pre- and post-processing.

The environment is similar to that of the Data Workbench, with menus to lead you through the design process. It is quite easy to generate a default application. After you select the tables and fields you want on a form, APT-Build generates a default form and a default application complete with menus. You then can customize the form and menus, adding presentation formatting and embedded processing logic. Here are a few samples of the things you can do:

- Design multitable forms using joins.
- Group multiple fields (e.g., the detail records on a master-detail form) and operate on them as a group.
- Create multilevel menus with pull-down and slide-off submenus.
- Define function keys on a form and attach an APT-SQL routine to each; forms called from a form can inherit its function keys.
- Decide whether or not to run rules on the form, depending on whether it is important for the user to get immediate feedback if a rule is not met when data is entered.
- Define custom field help.
- Create a values list from which the user can select a value and

Sybase Features Chart

Integrity	Part of data dictionary
Referential integrity	Yes; triggers
Field validation	Yes; rules/stored procedures
Stored procedures	Yes; part of data dictionary; pre-compiled
Concurrency control—locking levels	
Database	No
Table	Rarely (table lock must be specifically requested)
Record	No
Page	Yes; default locking level
Raw input/output	Yes; optional
Database can span multiple physical devices (disks)	Yes
Network support	DECnet and TCP/IP; integral part of the product
Distributed database capability	Included with Sybase RDBMS
Location transparency	No
Distributed query processing	Yes
Distributed query optimizer	No
Distributed transaction processing (multisite updates in a transaction)	Yes; not automatic; built into application on front end
Support for data replication	No
Access to heterogeneous databases	None; possible using Open Server

transfer it to the field on the form.

- Dynamically change properties on forms (e.g., prompts, display characteristics) at run-time in APT-SQL procedures.

You can also use APT-SQL procedures to further refine the application. The ability to execute a form while in design mode is extremely helpful for testing and debugging.

In the forms design process, we would like to see the form name displayed on screen, the ability to draw boxes, and simpler notation for defining a field as all upper/lower case or all numeric.

There is no direct access to the interactive SQL editor in APT Workbench, which disappointed us. We were thrown into the Unix system editor (in our case, vi) instead. While Sybase has explained that the system editor is more powerful and probably more familiar to the professional developer, we end users would like the option to use the friendly Sybase editor with menus attached.

Sybase recommends that, in developing applications, you first complete the necessary SQL work: creating all of the tables, triggers, rules, stored procedures, and defaults. Then, when you build the application, you invoke the database objects as required. This allows one group of developers to work on SQL and others to design the application without knowing SQL at all. Thus, the Sybase architecture makes it possible to extensively modularize the development effort. Applications are designed the same way for character terminals and bit-mapped workstations.

APT Workbench is easy to learn and use, up to a certain point. Writing complex APT-SQL procedures is not for the end user. Once you start creating more complex applications with lots of processing attached to fields, forms, and menus, it gets confusing to keep it all straight—what procedure goes with what object. The extensive set of properties offered for each object can get a little overwhelming as well. However, for the professional developer, the rich functionality and extensive flexibility of the product facilitate the design of very complex custom applications.

SECURITY. Sybase has extensive security options. It uses the standard "grant" and "revoke" commands to enforce database security at multiple levels. Permissions can be applied to both database objects and commands.

THE DEVELOPER'S PERSPECTIVE

We interviewed four Sybase developers about their experiences with the product. Three are VARs and one is a developer in an end-user organization. Length of time working with Sybase ranges from six months to over two years. All have experience with other DBMSs, such as Oracle, Informix, and MDBS, and tools like Focus.

STRENGTHS. Several strong points were mentioned by all of the developers: performance, integrity, flexibility, and the applications development environment.

Sybase Software Partners

Development Tools

Ashton-Tate
Borland International
Data Ease International
D&N Systems
Fortex
Information Builders
Inside Automation
National Information Systems
Progress Software Corporation
ServioLogic
Unify

Case Tools

DEFT
IDE

Artificial Intelligence

BIM
Gold Hill Computers
Natural Language Inc.
Neuron Data

Connectivity Tools

DB/ACCESS
Network Innovations

Decision Support

Access Technology
SPSS
Verity
V.I. Corporation

Performance. Performance is a key strength of Sybase. In fact, when the end-user organization tested its first applications with Sybase, the system went so fast they thought it wasn't working. Two developers have integrated Sybase with real-time data feeds. In one case, the Sybase database has to poll a process control system as often as every second while also processing user queries. Another developer stated that, although Sybase is not always faster, it is generally 1.5 to 2 times faster than Oracle on similar applications. One developer described Sybase as up to 400 percent faster than other products in some cases.

Architectural Integrity. All are impressed with the robustness of the database engine and the ability to build integrity into the back-end database. They stressed the major importance and benefits of having both stored procedures and triggers in the Sybase architecture.

Flexibility. Sybase offers the developer extensive options in structuring applications. By providing rich functionality and different ways to do things, Sybase allows the developer to optimize Sybase for a particular application's requirements. For example, triggers often do things that a normal application would do anyway. Using both application logic and triggers can result in a "double hit" on some integrity checks, which might adversely affect performance. Here, the developer has the choice to put the integrity checks in the back end or the front end, depending on the application.

Applications Development Environment. All of the developers mentioned the impact of the Sybase architecture on designing applications. While designing against the architecture, with its separation of functionality between the front and back ends, may at first look difficult, the Sybase architecture is a major strength of the product.

One benefit is the ability to create a database server, separating the physical aspects of the database (database administration) from the functional aspects (applications development).

One developer states that he can build an entire application with stored procedures and integrity mechanisms. "The Sybase architecture allows you to build a very strong design methodology. You can build a functional prototype quickly and attach either the front or back end later, building both separately."

As a caution, one developer described the need to proceed carefully in learning how to take advantage of the Sybase architecture in structuring an application. In most cases, the developer is faced with learning three new things: a product, an architecture, and an application. This person recommended that new developers use the "old" architecture more at first until they achieve a level of confidence with the new architecture. Using triggers is a somewhat radical way of structuring an application, and there is risk involved in using any new system for critical applications.

Developers in the end-user organization we spoke to took a similar approach. They redeveloped some of the more complicated pieces of existing systems, pushing Sybase to the limits of their own experience before trying it on a new application. They were able to build these modules in much less time, with much less code, yet get more functionality with Sybase. And these were the difficult applications that couldn't be done with products such as Oracle or Focus.

APT-SQL Language. APT-SQL language is a well-structured programming environment with the ability to manipulate objects at a high level. Commands are implemented at a high level of abstraction or expression, yet the developer has control at a low level in terms of manipulating the primitives of the product. One example here is the ability to change field properties dynamically. Oracle, for example, doesn't allow any manipula-

tion of field properties at run-time.

One developer described APT-SQL as a "small, elegant, and succinct language, with very little redundance." Another developer described it as "very clean."

C Interface. The developers also praised Sybase's programming interface, which provides the ability to shift effortlessly between APT-SQL and 3GL code. In many competing products (e.g., Focus), the developer must exit the 4GL environment to write a C program.

Support. Two developers cited support as a strength. The end-user organization stated that Sybase's technical support is "excellent, just superb. They assign you a technical rep on your first call. This person gets to know you, your company, and your applications." This developer also praised the Sybase documentation (it is easy to read, with many helpful examples) and the Sybase training courses.

Sybase has done a lot of things right. It has forged its own path to success, choosing not to be just another RDBMS.

Other. One developer stressed the importance of high availability. Two indicated that the company's efforts to port only to selected platforms was impressive, resulting in a tighter product and the ability to maintain high quality standards.

One developer is impressed with the overall quality of the Sybase staff, particularly in engineering. The company brings a great deal of experience to its product, boding well for continued improvements that will enable the company to maintain its state-of-the-art position.

WEAKNESSES. Sybase has two primary weaknesses: the current limitations of the front-end tools for presentation management, and the report writer.

Presentation. All of the developers mentioned the inflexibility of the Sybase front-end tools as an area requiring improvement. The end-user company had assumed Sybase would have more of a PC-like front end than it did. Problems include: When you tab into a field in Sybase, you are put at the end of the field entry in Insert mode, though you expect to be at the beginning of the field in Overstrike mode; no support for color; no facility to draw boxes on the screen; menus can only be a certain size at the top of the screen, taking up a lot of unnecessary space, particularly on terminal screens; the scrolling icons can be awkward; the arrow keys work differently than expected.

Report Writer. From the developer's perspective, Sybase has underestimated the need for a sophisticated report writer. The report writer is fine for simple ad hoc reports, but not for complex reports. The developer wants more of a programming language, with more functions and flexibility. One solution is to extend the 4GL code for reports. Two of the developers are using a product called SQR from D&N Systems (Burlington, Massachusetts) as an alternative. Sybase indicates that we will see enhancements to the report writer this year.

Other. One developer stated that Sybase needs to offer tools for performance monitoring. Since Sybase is a single process to the operating system, you cannot tell which application is causing a performance problem on the system. Sybase also needs better management tools for keeping track of source code (APT Workbench procedures).

Conclusion

Sybase has done a lot of things right. It has forged its own path to success, choosing not to be just another RDBMS. The company clearly defined its objectives and took a focused approach to both product architecture and marketing. As a result, the RDBMS has moved to a higher level of functionality, with the ability to meet the requirements of a new class of applications.

The company correctly anticipated important industry trends and emerging market needs: the move to distributed network computing, the need for performance and availability, and the benefits of database-enforced integrity. The Sybase RDBMS is designed to meet these needs, and its success is pushing the competition to meet the challenge.

The edge for Sybase is that its product is up and running, and the necessary functionality is an integral part of its architecture. Some of its competitors, on the other hand, are faced with the issue of redesigning their products. Oracle, Relational Technology, and Informix have all announced or implemented OLTP versions of their database engines. They are also now talking about the need for stored procedures and referential integrity. Perhaps Sybase has created some de facto standards of its own in the industry.

Sybase faces some challenges in the near future. The introduction of APT Workbench was long overdue and critical to the company's continued success. While the initial reaction to it is very positive, the front-end tools need to grow and mature. It is also important that the company not rest on its laurels, but continue to improve and enhance Sybase. The company plans to double in size again over the next year. It must manage its own growth with the same care and attention it has directed to its platform strategy.

We always like to see innovative newcomers like Sybase push the industry ahead and keep it on its toes. And from what we have seen so far, the company has the potential for a very bright future.

Patricia Seybold's Office Computing Group
Special Research Reports

Available
April, 1989

Compound Documents

VENDOR STRATEGIES, ARCHITECTURES, AND APPLICATIONS

By Michael D. Millikin

This special report explores the emergence of what will be a major application enabler in the coming years: compound documents.

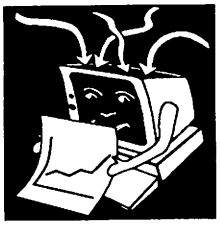
Compound documents are documents consisting of a variety of data types: text, image, and graphics, for example. In the context of a distributed network computing environment, compound documents become critical as a method of providing users with a transparent means to access, integrate, manipulate, and disseminate a variety of information stored on heterogeneous processors.

This report examines the four major defined document architectures:

- Digital's Compound Document Architecture (CDA)
- IBM's Mixed Object: Document Content Architecture (MO:DCA)
- ISO's Office Document Architecture (ODA)
- Wang's Wang Information Transfer Architecture (WITA)

We explore the functionality defined in each specification, and, where possible, give examples of sample applications.

Additionally, we will touch on the strategy for document architectures of two other vendors: Hewlett-Packard and Microsoft.



Available
June, 1989

Executive Support Systems

By Mickey Williamson

An Executive Support System is an important business tool that can integrate and summarize key information about a corporation and its competitive environment.

Unlike most other computer applications, these systems are adapted for justifiably impatient non-users who demand fast answers to questions about their companies and won't accept technical barriers to gathering those answers from different computers and databases.

This special report:

- Examines the opportunities and constraints faced by companies deciding to implement an ESS
- Compares in-house development, vendor-supplied software and consulting, and mixed development options
- Outlines the costs, time frames, and technologies, and the people who tend to be involved in successful ESS implementations
- Examines the leading ESS products and the technology trends that will determine the form and evolution of Executive Support Systems in the 1990s

OSF And UNIX International

SETTING THE OPEN SYSTEMS AGENDA

By Judith S. Hurwitz

The inception of the Open Software Foundation (OSF) and Unix International (UI) has changed the dynamics of the open systems industry for the future. These organizations are being influenced by the standards-setting work of X/Open.

This report will look at the origins, developments, and structure of these organizations as well as their plans for the future development of their environments. It will also examine the implications of each organization and the impact they will have on Unix and the standards movement as we move into the 1990s.

Among the highlights of this in-depth report are:

- User Interface: A close look at OSF's Motif and AT&T's Open Look
- Operating System Kernels: Design, implementation, and technologies
- Communications and Networking Foundations: How each group is building its networking underpinnings
- Applications Binary Interfaces: How OSF and UI will implement the ABIs and the implications for shrink-wrapped applications
- The Development Environment: Tools that will emerge for developing applications in these environments
- Administration, Backup, and Security: How these vital areas are being addressed
- X/Open and the Common Applications Environment: The impact that X/Open's portability guides will have on OSF and UI

Available May, 1989

ORDER FORM

☑ Please send me the research reports indicated below!

Qty:	Report:	Price:
____	Compound Documents	$495
____	Executive Support Systems	$495
____	OSF and Unix International	$495

Total Amount Due: _____

Form of Payment:

- ❑ Bill me P.O. # _____
- ❑ Check enclosed
- ❑ Charge to my: VISA/MasterCard/AMEX (circle one)

_____
Signature

_____
Card Number Exp. Date

Please print or type. Checks should be made payable to Patricia Seybold's Office Computing Group.

_____
Name

_____
Title

_____
Company/Dept.

_____
Address

_____
City, State, Zip

_____
Business Phone

Simply fill out the order form and send it to:

Patricia Seybold's
Office Computing Group
148 State St., Ste. 612
Boston, MA 02109

For information on multiple order discounts, contact Richard Allsbrook.

For *faster* service, call 1-800-826-2424 (in Mass. 617-742-5200) or fax 617-742-1028.

NEWS ANALYSIS

PRODUCTS • TRENDS • ISSUES • ANALYSIS

• INSIDE •

• HEWLETT-PACKARD •

HP Swallows Apollo

It was the best kept secret in the computer industry: Hewlett-Packard announced that it would acquire Apollo, and the industry was stunned. HP is not a company that typically makes acquisitions.

So why the switch? First, the two companies, at least on the research and development level, are a good cultural match. Both are technology driven, and both have a large stake in the workstation market. In fact, acting as a single company, they lead the workstation marketplace.

What might the new venture mean? Apollo can bring Hewlett-Packard some key technology that it would probably not be able to develop on its own in a timely fashion. First, Apollo gives HP direct access to its NCS (Network Computing System) technology. Second, Apollo's successful state-of-the-art object-management technology, which allows objects to be distributed across the network, could help HP accelerate the development of distributed objects for NewWave. Distributed objects are necessary to allow applications that live on multiple hardware platforms to communicate with each other. Apollo also has an object-oriented, user-interface development system called Open Dialog that will benefit HP.

Apollo will go down in history as an important pioneer. Its founders had the vision and insight to invent the engineering workstation. In its early, heady days, these powerful and then relatively inexpensive powerhouses sold themselves. Apollo's sales force could afford to sit back and let their workstations and unique brand of networking do the selling. Apollo's executives, enamored with the company's position in the marketplace, felt that justifying their products was beneath their status. At the same time, they watched Sun Microsystems out-maneuver and out-market them. Apollo engineers scoffed at the notion of using an unadorned version of Unix. The company believed that its highly specialized version of Unix, Aegis, was superior and that Unix was, in fact, a handicap for Sun. It was sure that Sun would trip when it freely licensed NFS to the industry. Sun continued to amass strength and even surpass Apollo. Apollo never managed to recover, and thereafter found it impossible to keep from lambasting Sun at every turn.

While there was no doubt that Apollo had good technology (HP is even considering implementing Apollo's Prism RISC technology), it had serious problems marketing its wares. Whether HP will be able to solve this problem is uncertain. Like Apollo, HP has never been a flashy marketer. HP's challenge will be to combine the technological strengths of both companies and get the message out.

CONCLUSION. We think that the match between Apollo and Hewlett-Packard is a good one. If HP uses Apollo as a technology R&D organization, it will gain some excellent technology that it sorely needs. Apollo's organization could benefit from the savvy and market presence that HP has gained in recent years. It will be an interesting marriage to watch.

—J. Hurwitz

• OMG •

Object Management Group Formed

In an attempt to set standards for the way in which multivendor applications will communicate and interoperate across heterogeneous platforms and operating systems, eleven companies have formed the Object Management Group (OMG). The founding members are

Canon, Data General, Gold Hill Computers, Hewlett-Packard, Philips International B.V., Prime Computer, Soft•Switch, Sun Microsystems, 3Com, Unisys, and American Airlines.

The goal of the OMG, according to Christopher Stone, executive director of the consortium and group manager of software for DG, "is to achieve a common, worldwide, object-oriented, applications-integration environment which will allow users to mix and match independently-developed applications across various operating systems." These include Unix, DOS, and OS/2.

OMG OBJECTIVES. The specific objectives of the OMG include:

- Promote an object-oriented, applications-integration environment based on appropriate industry standards.
- Promote a framework for compatible and independent development of applications.
- Enable coordination among applications across heterogeneous networked systems in a multinational, multilingual environment.
- Adopt a core of commercially available implementations of the framework and promote their international market acceptance and use.
- Actively influence the future direction and development of these core products and technologies.
- Foster the development of tools and applications that conform to and extend this framework, and provide a mechanism for certifying compliance with the core technology.
- Adopt Hewlett-Packard's NewWave object management facility, independent of user interface, as a working example of this framework.

This last point has drawn the most interest and has led to the comment that the OMG seems to be more of a NewWave user's group than an independent entity. The founding members have tried to make it clear that HP's object management facility (OMF) is but one of the technologies that will eventually be encompassed in the object management framework. Each of the companies is actively developing technologies and products in this area, and each is committed to submitting those that are compatible with and that will extend the core technology. The group is also seeking additional members from the vendor and end-user communities.

NEWWAVE AS AN EXAMPLE. NewWave is, however, very important to the OMG, both as a core technology and as a demonstration of how the object-oriented model can enable applications to communicate and work together without being specifically written to know about each other. Thus, under NewWave, a word processor will not have to have the capability of creating and manipulating graphics or of spell-checking. Rather, the developer of the word processor would be able to concentrate on the quality of his or her product, which would call on a graphics or spell-checking program to handle the tasks that each does best.

Bill Crow, research and development project manager for HP, gives an example of the value and coming necessity for this. He notes that there are hundreds of commercial applications and thousands of homegrown programs that can take data from Lotus 1-2-3 files because they know how to read a .WK1 file format. When Lotus releases its next version, it is likely to change the file format, as it did from Release 1A to Release 2. All of these applications will have to be rewritten to accommodate the new format, and then rewritten again and again for each change in each application that Lotus supports. If, however, your application knew how to ask 1-2-3 for its data or chart, rather than trying to take the data from 1-2-3's files, the file format changes would not cause a problem. According to Crow—who quotes SmallTalk pioneer and founder of ParcPlace Systems Adele Goldberg—"Applications should not touch—they should ask." What the OMG is all about is standardizing the way to ask.

Another example is provided by a NewWave ISV that is developing a voice product which provides the ability to record, edit, and play back voice messages. The same object can be called by an electronic mail system to provide voice mail, or by a word processor or spreadsheet to provide voice annotation, in a particular document or worksheet cell.

One problem with NewWave is that it is currently only available for the DOS platform. Neither Unix nor OS/2 versions are ready, and the product is still limited to a single workstation. However, HP is now demonstrating a version of NewWave that has distributed capabilities, allowing a user on one workstation to access a NewWave object on another. The demonstration shows multiple users viewing the same object, with their views changing to reflect changes occuring in the application. These changes can come from any of the permitted users, independent of location.

Other companies will add to NewWave and produce their own product sets. Data General, for example, is planning to extend the object-management capabilities across the network. Thus, network services—mail, filing, printing, and also project manager, expert system, etc.—can be defined as objects and can be invoked by any other object anywhere on the network.

As DG and the other members bring out their applications, adherence to the OMG specifications should assure that they will be able to work together.

POSTSCRIPT. The April issue of our audio publication—*P. S. postscript on information technology*—focuses on object orientation, object management, and the Object Management Group. In it, Patricia Seybold discusses these issues with Chris Stone, Bill Crow, Adele Goldberg, and Joyce Wren, American Airlines vice president of information systems technology. The tape also includes a lively discussion between Bill Crow and Adele Goldberg on whether the time is ripe for an object management consortium. ◐

—D. Marshak

Patricia Seybold's Computer Industry Reports

ORDER FORM

Please start my subscription to:

		U.S.A.	Canada	Foreign
☐ *Patricia Seybold's Office Computing Report*	12 issues per year	$385	$397	$409
☐ *Patricia Seybold's UNIX in the Office*	12 issues per year	$495	$507	$519
☐ *Patricia Seybold's Network Monitor*	12 issues per year	$495	$507	$519
☐ *P.S. postscript on information technology*	12 issues & tapes per year	$395	$407	$419
☐ *P.S. postscript on information technology*	12 issues per year	$ 95	$107	$119

Please send me a sample of: ☐ *Office Computing Report* ☐ *UNIX in the Office* ☐ *Network Monitor* ☐ *P.S. postscript on information technology*

Please send me information on: ☐ Consulting ☐ Special Reports ☐ Conferences

☐ **My check for $_____ is enclosed.** ☐ **Please bill me.** ☐ **Please charge my subscription to: Mastercard/Visa/American Express** (circle one)

Name: _____ Title: _____

Company Name: _____ Dept.: _____

Address: _____

City, State, Zip code: _____

Country: _____ Bus. Tel. No.: _____

Card #: _____

Exp. Date: _____

Signature: _____

Checks from Canada and elsewhere outside the United States should be made payable in U.S. dollars. You may transfer funds directly to our bank: Shawmut Bank of Boston, State Street Branch, Boston, MA 02109, into the account of Patricia Seybold's Office Computing Group, account number 20-093-118-6. Please be sure to identify the name of the subscriber and nature of the order if funds are transferred bank-to-bank.

IU-0589

Send to: Patricia Seybold's Office Computing Group: 148 State Street, Boston MA 02109; FAX: 1-617-742-1028; MCI Mail: PSOCG
To order by phone: call (617) 742-5200

Topics covered in Patricia Seybold's Computer Industry Reports in 1988/1989:

Back issues are available, call (617) 742-5200 for more information.

Office Computing Report

1988—Volume 11

#	Date	Title
7	July	AI Update—Expert Systems Flourish While the AI Industry Disappears
8	Aug.	Metaphor Computer Systems—A Quiet Revolution
9	Sep.	Technology That Supports Meetings
10	Oct.	Document Processing
11	Nov.	Upper CASE Tools—Getting Users to Help Design Systems
12	Dec.	Executive Information Systems

1989—Volume 12

#	Date	Title
1	Jan.	Digital's Office Strategy—Going After the Desktop
2	Feb.	Compound Documents Take the Stage
3	Mar.	Wang's Freestyle—Not the Paperless Office, But an Office with Less Paper
4	Apr.	Focus on Training Technologies—Office Systems Training from DG, Digital, HP, IBM, and Wang

UNIX in the Office

1988—Volume 3

#	Date	Title
7	July	Olivetti—Global Bridges on a Standard Foundation
8	Aug.	BBN's Slate System—Delivering Compound Documents
9	Sep.	Distributed Databases—Where Are We?
10	Oct.	DECwindows
11	Nov.	Data General's Unix—A New Beginnning
12	Dec.	Document Processing with Interleaf *Second Feature:* Apollo and Sun Service and Support

1989—Volume 4

#	Date	Title
1	Jan.	Unisys Ofis Ensemble—In Search of Cohesion
2	Feb.	A Tale of Two Operating Systems—System V.4 and OSF1
3	Mar.	FrameMaker 2.0—The Art and Science of Document Processing
4	Apr.	CMU's Andrew Project—A Prototype for Distributed, Object-Oriented Computing

Network Monitor

1988—Volume 3

#	Date	Title
7	July	Transparent File Systems—The Locus Model for DNC
8	Aug.	Novell—The Juggernaut Rolls On
9	Sep.	OSI Standards—A Promise Fullfilled?
10	Oct.	ISDN: We Still Don't Know
11	Nov.	Data General's PC Networking Strategy
12	Dec.	3+Open—3Com Makes Its Move

1989—Volume 4

#	Date	Title
1	Jan.	APPN—Peer-to-Peer Networking Arrives at IBM
2	Feb.	Bridges & Routers—Dividing to Conquer
3	Mar.	BBN Builds Big Networks—Designing Private Packet-Switched Networks
4	Apr.	Northern Telecom DNS—Trying to Build Intelligent Networks