Patricia Seybold Group

Editor-in-Chief
Michael A. Goulde

OPEN INFORMATION SYSTEMS

Guide to Unix and Other Open Systems

Vol. 8, No. 10 • ISSN: 1058-4161 • October 1993

INSIDE

Progress Software's Version 7

Solving Client/Server Migration Issues

By Judith R. Davis

IN BRIEF: In the face of intense competition in the relational database market, Progress Corporation has taken a tack away from emphasizing its database engine to drawing attention to its data access capabilities and application development tools. Application developers familiar with Progress tools have always rated them very highly. With Version 7, Progress has introduced a robust and integrated cross-platform development and deployment environment that combines the best of both 4GL and visual approaches. Its database-independent Data Server architecture allows applications to be deployed across the many supported DBMSs without modification. Progress is facing the challenge of maintaining continued growth and prosperity with a solid product line and an open system strategy. It hopes that Version 7 will help the company extend its reach beyond its traditional markets and achieve more widespread acceptance.

Report begins on page 3.

Transferring the Unix Trademark

Who Really Benefits?

NOVELL RECENTLY ANNOUNCED that it has agreed to transfer the Unix trademark to X/Open, which, in turn, will use it as its brand for products that comply with the Unix API specification. The announcement came after intense negotiations and one false start.

Novell will transfer ownership and management of the Unix brand name to X/Open in exchange for becoming an X/Open shareholder and board member, membership fee waivers, and licensing fee waivers for three years. X/Open will use the Unix trademark as the brand for products which comply with the final version of the Spec 1170 standard. On an interim basis—until the specification is completed—products which are XPG3- or XPG4-compliant, are SVID-compliant, and whose vendors have Unix licenses with USL, can use the trademark. Contrary to the fears of many vendors, Novell's UnixWare was not involved in the final agreement—just Spec 1170.

Companies will be able to have the X/Open Unix trademark on their products without paying royalty fees to Novell. Instead, they will only have to pay fees for any code licensed from Novell. The Unix trademark will be licensed from X/Open, with fees going to that consortium, and only if the product is to carry the Unix brand and meets X/Open certification. Specific licensing terms have not yet been determined, but X/Open's objectives are to break even and to have licensing be usage-based.

X/Open is now in the position of promoting an operating system standard for open systems that is clearly biased toward Unix. This reveals that many of the open systems standardization efforts the industry has engaged in over the past ten years have really been aimed at standardizing Unix. Apparently we are supposed to forget past statements about creating implementation-neutral standards that can be supported by a wide range of technologies.

Microsoft's assertions that the open systems movement has just been a code word for Unix gain credibility as a result of this action. Should Microsoft have to support Spec 1170 and carry the Unix brand in order to support open systems? Granted, Microsoft's interest in open systems is primarily focused on minimal compliance with procurement requirements. However, although limiting open systems to Unix is not in Microsoft's best interests, the real problem is that limiting open systems to one operating system is not in the best interests of customers either.

The beneficiaries here are the Unix vendors. The losers are the users who will potentially find their choices limited. Not that Unix is not good technology. It has matured, improved, and become competitive and commercially viable. But users should be able to choose the operating system and hardware platform that offers optimum performance for a particular application. The goals of portability and interoperability cannot be achieved by enforced sameness, since opportunities for innovation and advancement of technology would disappear.

Particularly frustrating is the fact that the Unix brand won't guarantee absolute portability or interoperability anyway. Vendors such as Digital, SCO, and SunSoft are free to add superset extensions beyond the X/Open standard and still call their product Unix. The trademark refers to the subset of common APIs. Proprietary extensions are not excluded and will not affect the ability of a vendor to label a product Unix. This may spell even worse confusion for users, who will now expect that X/Open branding has resulted in one and only one Unix. Although this process will create a consistent base-level Unix, it will lead to a narrow definition of open systems.

OPEN INFORMATION SYSTEMS

Editor-in-Chief
Michael A. Goulde

MCI:
MGoulde
Internet:
mgoulde@mcimail.com

Publisher
PATRICIA B. SEYBOLD

Analysts and Editors
JUDITH R. DAVIS
STANLEY H. DOLBERG
MITCHELL I. KRAMER
DAVID S. MARSHAK
RONNI T. MARSHAK
JOHN R. RYMER
ANDREW D. WOLFE, JR.

Copy Editors
ALBERT C. D'AMATO
MIRIAM F. D'AMATO

Art Director
LAURINDA P. O'CONNOR

Sales Director
PHYLLIS GUILIANO

Circulation Manager
DEBORAH A. HAY

Customer Relations Manager
DONALD K. BAILLARGEON

Patricia Seybold Group
148 State Street, 7th Floor,
Boston, Massachusetts 02109
Telephone: (617) 742-5200 or
(800) 826-2424
Fax: (617) 742-1028
MCI: PSOCG
Internet: psocg@mcimail.com
TELEX: 6503122583

Open Information Systems (ISSN 0890-4685) is published monthly for $495 (US), $507 (Canada), and $519 (Foreign) per year by Patricia Seybold Group, 148 State Street, 7th Floor, Boston, MA 02109. Second-class postage permit at Boston, MA and additional mailing offices.

POSTMASTER: Send address changes to *Open Information Systems*, 148 State Street, 7th Floor, Boston, MA 02109.

Progress Software's Version 7

Solving Client/Server Migration Issues

The Move to Mission-Critical Client/Server

A Tough Set of Requirements for Tools Vendors

As more and more organizations move beyond simply prototyping client/server applications and begin to deploy real mission-critical versions, two trends are occurring. The more obvious one is that the market for client/server application development tools is growing fast and gets more crowded every day. Everyone, it seems, wants to grab a share of the huge projected revenue. The other trend is that customers are beginning to take a harder look at what a tool offers beyond the ability to create a snazzy graphical interface. How complex can the underlying application really be, and how hard does the developer have to work at it? What tools are available to deploy the application in multiple locations and across multiple platforms? How difficult is it to maintain the application over time? There's more to most applications than just a pretty interface and fast development of screens.

Progress Enters the Database-Independent Tools Market

The clash of these trends will most likely force a shake-up in the industry, exacerbated by the entry of large client/server companies, such as Sybase Incorporated (Emeryville, California) and Oracle Corporation (Redwood Shores, California), and of vendors with proven application development toolsets, such as Progress Software Corporation (Bedford, Massachusetts). In rolling out Version 7 (V7) of its respected Progress Application Development Environment (ADE) and relational database management system (RDBMS), Progress is, for the first time, positioning itself to compete directly with third-party client/server tools vendors such as Powersoft Corporation (Burlington, Massachusetts), Uniface Corporation (Alameda, California), JYACC (New York City), and Gupta Corporation (Menlo Park, California). Progress Version 7 addresses the major issues facing all client/server tools vendors with the following:

- A cross-platform graphical development environment. Version 7 implements a long-awaited graphical user interface (GUI) for the Progress toolset and provides the ability to generate event-driven applications in addition to the more traditional procedural applications. Development options are Windows and Motif, with a subset of tools also available for character terminals.

- A cross-platform deployment environment. Version 7 applications can be deployed across multiple GUIs (Windows and Motif) as well as character-based terminals without modification. One of the tenets of V7 is the separation of development and deployment decisions. (See Illustration 1.) Customers can choose a development platform independent of where the application will run. V7 also offers a smooth migration path for existing Progress customers since it can run Version 6 applications unchanged. A cross-platform deployment environment helps Progress meet the needs of customers who are downsizing and migrating applications from host-based to client/server.

- A robust development environment. A primary strength for Progress has always been its robust, flexible, and complete 4GL and integrated development environment. Experienced developers consistently praise the Progress language for its ability to support complex, mission-critical application development without requiring any 3GL

code. The 4GL has been extended in V7 to support graphical objects and event-driven programming. The V7 User Interface Builder, the Data Dictionary, and the 4GL are all tightly integrated and offer great flexibility in creating applications.

- Database independence. With the Progress DataServer Architecture, Progress applications can be deployed across supported DBMSs without modification. These include Oracle, Sybase, Rdb, RMS, C-ISAM, and the IBM OS/400 Database Manager. Coming is support for DB2 (and other IBM RDBMS products) via Distributed Relational Database Architecture (DRDA), Microsoft Corporation's (Redmond, Washington) Open Database Connectivity (ODBC) API, Hewlett-Packard Company's (HP, Palo Alto, California) Allbase, and Object Design Incorporated's (Burlington, MA) ObjectStore object-oriented DBMS. (See Illustration 2.) In V7, Progress is taking a superset approach with plans to offer the same functionality across all data managers, whether the data manager supports it or not. The company also has an interesting twist in its new support for database triggers. Progress triggers are executed in the client and are, therefore, portable across non-Progress DBMSs. Progress is unbundling its tools and database server products to better support customers who want to use Progress applications against a non-Progress data manager.

- Tools to manage the application development environment. V7 does not yet offer tools here but will in the future.

- Tools to facilitate deployment of applications, not just development. This is another area where Progress has traditionally been strong. V7 adds a new Translation Manager to help the developer deploy an application in multiple languages.

- Integration with other applications on the desktop and with multiple data sources on the back end, including access to legacy data. The Progress DataServers include support for distributed access and heterogeneous joins today. In its overall architecture, Progress plans to add API support for Windows on the front end, ODBC on both the front and back ends, and DataServers for additional data sources such as DB2 and HP's Allbase. The Windows version of V7 supports both DDE and DLLs but not yet OLE.

Progress V7 Product Architecture

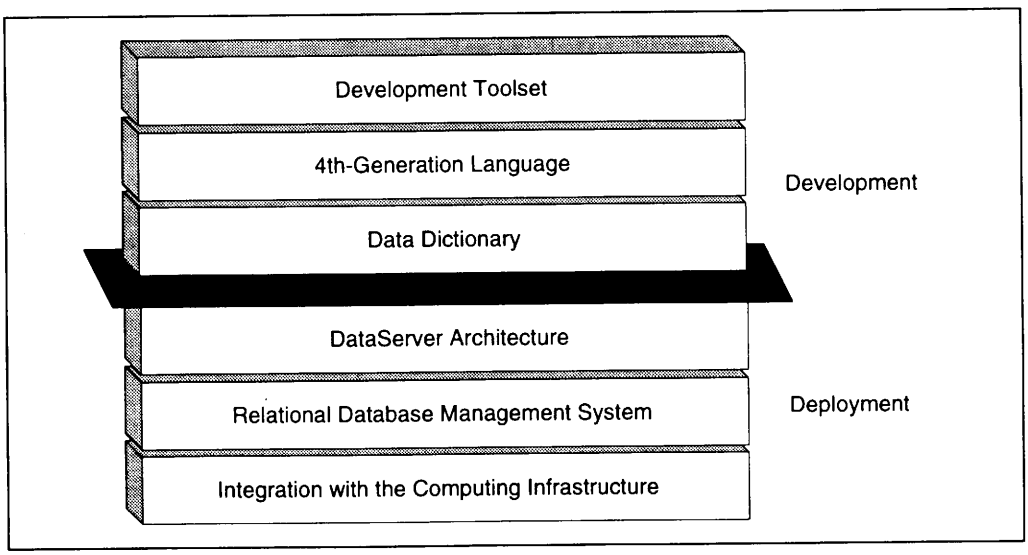

Illustration 1. The Progress 4GL (extended to handle graphical objects and event-driven applications) and Data Dictionary are the foundation for Progress Version 7 enhancements. A major objective of V7 is separation of the development environment from deployment.

Progress for Client/Server Development

The developer's release of V7 is shipping now, and deployment versions are due by the end of this year. In the past, Progress has been considered a company that provided an RDBMS and associated tools. With this release, the tools have been broken out separately and should be considered as viable for client/server application development. In this issue, we take an in-depth look at Progress V7—its overall architecture, the DataServer architecture, the new tools enhancements, and the implementation of triggers—and assess its overall position in the competitive client/server database and tools market.

Who Is Progress Software?

Approaching $100+ Million in Revenues

Progress Software continues to grow at an impressive rate. Revenues were $85 million in the fiscal year ended in November 1992 and $79 million for the first nine months of this fiscal year. The company's success is based on a combination of a strong 4GL-based application development environment, a good RDBMS on the back end, and a loyal value-added reseller (VAR) following. Because the company's primary distribution is through indirect channels, primarily VARs, Progress tends to be less visible in the general marketplace than its larger competitors.

The Progress Mission Is to Improve Developer Productivity

Progress has always been focused on improving the application development process and developer productivity. We have already summarized the strengths of the Progress development tools and of its DataServers for access to non-Progress data. Progress also has an end-user tool, Progress Results, for data access and reporting.

The Progress RDBMS on the back end is competent as well, featuring a multithreaded, multiserver architecture; sophisticated transaction management; a high degree of reliability (users state that they have never lost data with Progress); and an automatic two-phase commit for distributed transactions. Because Progress is a database vendor, it must pay attention to requirements on this side as well. In V7, Progress has optimized the server better for distributed client/server environments with improved data retrieval strategies and has added advanced server features—triggers, sequences, word indexing, and full support for ANSI SQL89 Level 2 (but not for declarative integrity constraints).

DOS, Windows, and AS/400, in addition to Unix and Novell

In addition to a wide range of Unix platforms, the complete Progress product line (Version 6.3) currently runs on DOS, OS/2, Digital Equipment Corporation's (Maynard, Massachusetts) VAX and Alpha OpenVMS, Unisys Corporation's CTOS, and LANs. Alpha/OSF/1 is expected by the end of November, and Alpha/Windows NT is planned for 1994. Although Unix is the company's primary platform, Progress has moved to two other strategic platforms over the past year. One is Novell Corporation's (Provo, Utah) NetWare, and the other is IBM's (Armonk, New York) AS/400 (providing both tools and a DataServer for the OS/400 Database Manager). Another strategic server platform for V7 is Windows 3.1, with Windows NT coming as well. Progress also has plans to port its tools to HP's MPE/ix (POSIX-compliant) platform. Progress will continue to port Version 6.3 to new platforms while it rolls out V7.

Seventy percent of Progress's licensing revenue is from the Unix market, but PC-based products (DOS, Windows, NetWare, and OS/2) represent 50 percent of license shipments.

DataServer Architecture for Heterogeneous Client/Server

Opening up the Progress Architecture

Progress's DataServer Architecture is a layered set of services and interfaces that enable a Progress application to access data in a distributed, heterogeneous environment. The Progress RDBMS is not required to connect a Progress application to a non-Progress data manager. One of Progress's goals is to provide a more open environment for heterogeneous clients and servers. In V7, the company is fine-tuning its DataServer architecture to give Progress

applications access to non-Progress data on the back end and is opening up the front end to allow non-Progress applications written to industry-standard interfaces to plug into the Progress environment.

Progress Client/Server Architecture

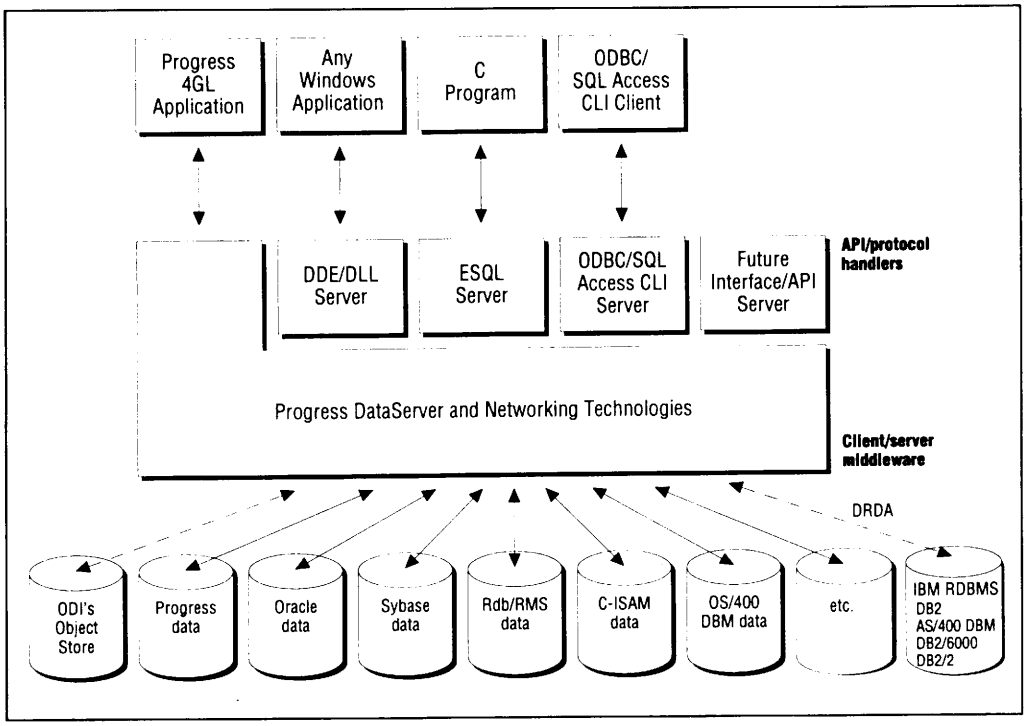

Illustration 2. Progress intends to support non-Progress interfaces as well as non-Progress data sources as part of its client/server infrastructure.

Data Integration and Database Independence

The DataServer architecture is designed to meet the following customer requirements:

- Simple data integration. The need to merge data from multiple, heterogeneous data sources, especially important in decision-support applications, is answered.
- Access to and/or migration from legacy data. The Progress DataServers for RMS, Rdb, C-ISAM, CTOS-ISAM, and IBM's OS/400 Database Manager are geared for access to and/or migration from legacy data.
- Database-independent applications. Progress is targeting Oracle and Sybase as the primary databases to support.

Progress is also trying to balance the needs of two different categories of customers. One is the company's current VARs, which want to sell against other databases with a single set of source code. They don't want to have to change the application when the underlying database changes. The other category is customers that want to develop new applications using state-of-the-art development tools, choosing the database manager separately.

The Data Manager Is Transparent to the Application

DataServers give Progress the ability to run a single 4GL program against any supported data manager without concern about the correct SQL dialect or data access language. Progress bears the burden for understanding the SQL dialect of the underlying DBMS and making the data manager transparent to the application. Thus, the application gets the same behavior from all data managers. The application always has the option to pass through native SQL to the underlying data manager if appropriate.

A Superset of Functionality on Both Sides

The V7 DataServers introduce significant enhancements over previous Progress gateway technology. The DataServers will ultimately provide access to all Progress functionality, whether the data manager supports it or not, and will emulate all native data manager functionality, even if Progress doesn't support it. For example, Progress will support scrollable cursors in its gateways, even though DBMSs like Sybase and Oracle do not. An example of a feature found in DBMSs such as Sybase and Oracle but not Progress is stored procedure calls. Progress is thus taking the ultimate "superset" approach to heterogeneous application development.

The current Version 6 gateways provide no emulation of non-Progress functionality (except for support for Sybase stored procedures, introduced in the Version 6.3 Sybase Gateway) and cannot support all-Progress functionality, such as "arrays" and "find previous." (Progress has, however, just released a V.6 Oracle gateway that does support scrollable cursors.)

The DataServer Architecture

There are several components of the Progress DataServer Architecture. (See Illustration 2). The API/protocol handlers will support non-Progress client interfaces on the front end, including ODBC and other versions of the SQL Access Group Call Level Interface (CLI), Windows Dynamic Data Exchange (DDE), and Dynamic Link Libraries (DLLs), in addition to the embedded SQL for C interface. Progress is evaluating support for Integrated Database Application Programming Interface (IDAPI) but is committed to supporting ODBC for both its client and server by the end of 1993. Its client/server middleware includes the DataServer Backplane, the Data Dictionary, and the DataServer itself.

DATASERVER BACKPLANE. The DataServer Backplane is a single interface to the application plus the integration point for data retrieved from multiple databases. The Progress middleware layer not only will support multiple APIs but also will provide heterogeneous distributed database functionality for distributed joins and transactions where possible. This enhances the value of the Progress solution. The key objective is to provide an open client/server computing environment. This architecture is similar to Sybase's approach with its Open Client Release 10 and OmniSQL gateway, and it is different from the approaches taken by both Borland International Corporation (Scotts Valley, California) with IDAPI and Oracle with Oracle Glue.

DATA DICTIONARY. The Progress Data Dictionary is a central repository for the schema for all supported databases, providing location and data transparency for the application. For non-Progress databases, this is an extended version of the native Progress data dictionary. Referred to as a "schema holder," it includes type of database, location, connection information, and triggers, in addition to standard data dictionary information. If the back-end data manager also supports triggers, the developer must decide where to implement them. Putting triggers on the Progress side means that they will always fire, regardless of the data manager.

DATASERVER. The Progress DataServer is the software that supports a specific data manager, translating Progress data access statements into the appropriate native language or calls. A key aspect of the DataServer is the use of native client/server connectivity. For example, Progress uses the Open Client API for Sybase and the Oracle Call Interface (OCI) for Oracle. This is necessary in order to provide good performance against each DBMS, enabling Progress to meet the requirements of OLTP applications, not just decision-support access. Progress can also pass through native SQL statements without touching them. For client/server implementations, the customer can use networking from the DBMS vendor or optional networking from Progress.

Migrating data from one data manager to another involves changing the data dictionary and recompiling the application. Progress provides a utility that converts a Progress schema to Oracle or to Sybase and generates the appropriate data definition language. The DataServer for HP's Allbase will support this conversion as well.

An Overview of the Version 7 Toolset

V7 Balances New and Existing Application Requirements

A major goal for V7 is support for industry-standard GUIs for both development and application deployment. However, compatibility with Progress V6 applications and the ability to continue deploying applications on character-based terminals were two key criteria for Progress in designing V7. Both are necessary to give existing customers an incremental migration path into the V7 graphical environment. In this regard, Progress is taking a different approach from many of its competitors.

The ability to effectively run an application designed for a GUI on a character-based terminal without modification is still important to many customers. Generally, developers do not want to have to use different tools to write applications for terminals and workstations, nor do they want to write an application twice to accommodate both types of desktops. Character-based terminals are not disappearing, and vendors like Progress, with a large installed base of applications, must continue to support that environment. The company added character extensions to V7 to allow Windows-like interfaces to be built on any character-based desktop (e.g., ASCII terminal or DOS). While Progress will fully support all capabilities in every GUI environment, it will also provide what it calls "gentle degradation" for the character-based desktop. (See Illustration 3).

Progress takes a "superset" approach to interfaces, supporting the same features across all deployment environments regardless of differences among native capabilities. The character extensions mentioned above are one example. Another is the new online help facility in V7. The developer can build a full Windows-compliant help system with support for advanced capabilities (hypertext links, cut and paste, annotations, graphics) and move it unchanged to a character (except for graphics) or Motif desktop.

Presentation Independence across Both GUIs and Terminals

The initial release of V7 offers Windows and ASCII terminal (under Unix) development and deployment options. Initial Unix platforms include HP 9000, IBM RS/6000, SunOS, and SCO. Motif development and deployment will follow by the end of 1993. Progress plans to add DOS character development and deployment by mid-1994 and is considering Macintosh and Presentation Manager for the future. Progress uses the native GUI toolkits for Windows and Motif, making calls directly to the windowing system to draw graphic elements on the screen.

The full V7 development toolset—the Data Dictionary, the User Interface Builder (UIB), the Procedure Editor, the interactive Debugger, the Translation Manager, the application compiler, and the online help system—runs on graphical platforms. Progress also offers a subset of the V7 toolset for character-based platforms—the Data Dictionary, the Procedure Editor, online help, and the application compiler. There are obviously no visual programming tools in the character-based environment, and it is unlikely that a developer would use character-based tools to create a graphical application. However, the new interactive GUI debugger can be used to debug character applications. A big benefit Progress provides is the ability to separate the development environment from the deployment environment. The same application can run on a Unix server with terminals or on Windows clients connected to the Unix server.

Key Foundations: The 4GL and the Data Dictionary

The foundation for V7 continues to be the Progress 4GL/Procedure Editor and the Data Dictionary. These two components of the toolset have always been tightly integrated, and both are enhanced in V7.

Presentation Independence in Progress V7

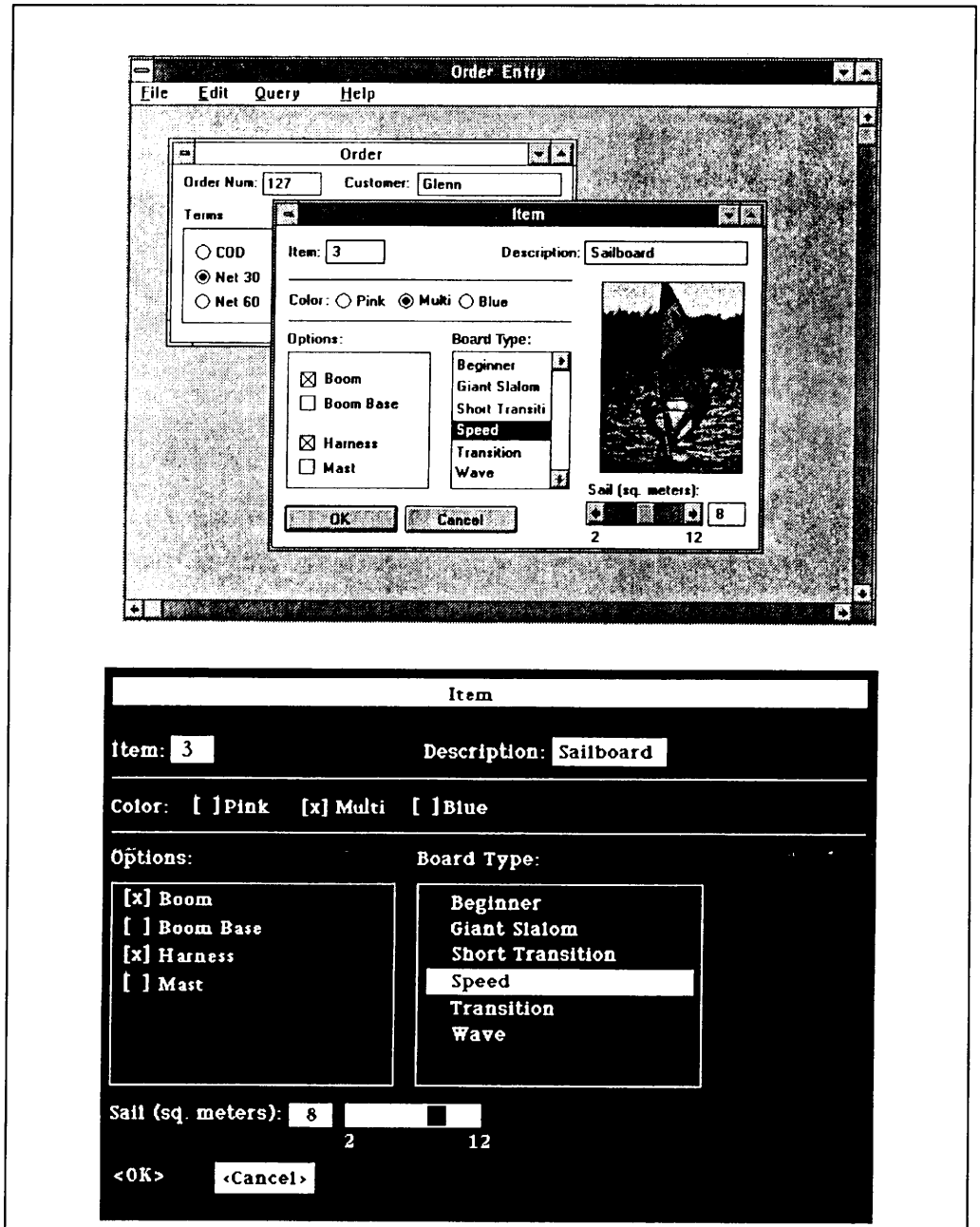

Illustration 3. The top screen is a Progress application running in Microsoft Windows. The bottom screen is the same application running on a Unix ASCII terminal. You can see how Progress V7 maps Windows graphical widgets in the character application.

The 4GL Now Supports Graphical Objects...

In V7, the 4GL language has extensions to support graphical objects: windows, buttons, toggle boxes, radio sets, scrolling editors, selection lists, menu bars, pop-up and pull-down menus, sliders, data browsers, dialog boxes, images, and complex text (e.g., hypertext links). The language also supports drag-and-drop and the ability to launch multiple windows from within Progress and have multiple frames active at the same time. In the Progress environment, a frame is a container for grouping, organizing, and displaying other objects.

Progress V7 not only extends the user-interface capability on the graphical side, but it also provides all these options on the character side. (The only exceptions are windows and images, which cannot be implemented in a character environment.) Text can be displayed as a scrolling editor, two-valued fields (e.g., shipped, exempt) can be captured as a toggle box, buttons can be used to execute tasks, etc. (See Illustration 3 for a comparison of character and graphical implementations of the same screen.) On DOS, the user will also be able to use a mouse to navigate around the application.

...And Events

V7 also has the ability to generate event-driven applications in addition to traditional top-down procedural applications. As in most event-driven environments, the developer can now attach 4GL instructions/procedures to predefined events associated with an object. Progress events fall into two categories. The first is database events, which are events associated with a table or field definition: insert, update, delete, find records in a table, or assign a value to a field. The second category includes interface events, such as enter or exit a field, or double-click on a button. The actions associated with these events are called triggers. The user interface events associated with a particular object depend on the type of object.

Persistent UI Triggers Apply at the Application Level

In addition to user interface triggers associated with events on particular objects on a window, Progress has the concept of *persistent* triggers, which apply globally to the application. An example is keyboard accelerators: Every time the user hits a function key or combination of keys, a particular menu option is executed. Persistent triggers stored in a separate file can be shared among multiple applications as well. This gives application developers reusable code across, not just within, applications.

Data Dictionary Enhancements: Triggers and GUI Defaults

Progress has added a graphical front end to the Data Dictionary tool. (See Illustration 4). The Data Dictionary now also contains database triggers (we discuss these in "The Progress RDBMS Adds Triggers" below), sequence generators, and visual defaults for the display of database fields. The visual defaults are implemented in a "view-as" phrase. For example, you could specify in the data dictionary that the default "view-as" for a long "comments" field is a scrolling editor of a particular size with both horizontal and vertical scroll bars and wordwrap. Anytime the comments field is placed in a window (without a widget being defined first), the default visualization comes with it automatically. The developer can always modify this default if appropriate. One Progress VAR is making heavy use of the "view-as" option to give end users a look and feel in an ad hoc environment that is consistent with the one they are familiar with in the VAR application.

We should note that Progress does not yet support full inheritance, so changing the "view-as" default in the data dictionary has no effect on existing applications. The "view-as" phrase also does not include attributes such as color or font. The Data Dictionary tool is included with the Progress development tools, the Progress RDBMS, and the DataServers.

UIB, Debugger, and Translation Manager Are New

NEW COMPONENTS. The Debugger, the User Interface Builder (UIB), and the Translation Manager are all brand new components of the toolset, contributing significant functionality to the Progress application development environment. The interactive Debugger can test all application components, including user interface logic (e.g., how objects are instantiated and why certain elements don't appear on the screen when you expect them to), procedures, database triggers (because they execute on the client), and batch and reporting logic. The developer can view and change structures while the application is running. The debugger is also extensible and customizable. In the following sections, we discuss the UIB and Translation Manager in more detail.

The Progress Data Dictionary

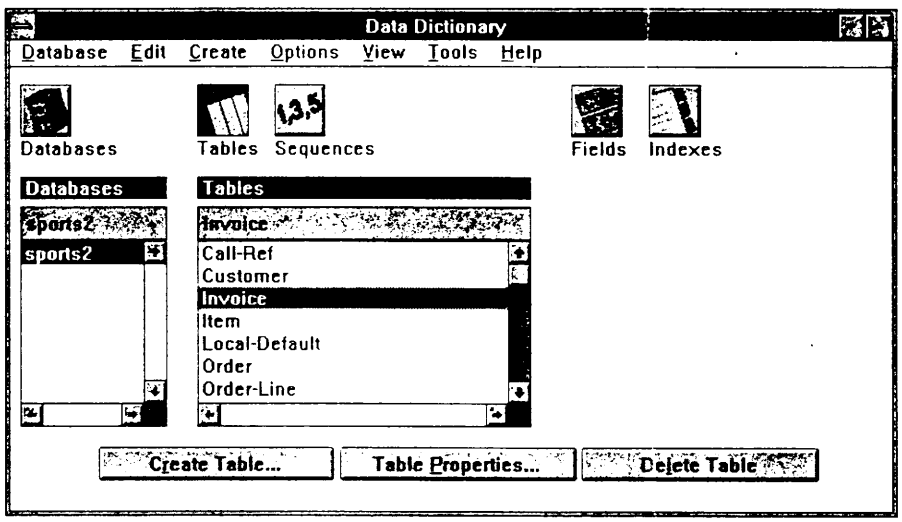

Illustration 4. The Progress Data Dictionary has a graphical interface and now includes database triggers, sequences, and "view-as" defaults for fields in a table.

Building Applications in Version 7

Flexible Application Development through Tight Integration

The UIB is the central tool for designing and building interactive applications. (See Illustration 5.) There are three goals for the UIB. The first is to provide an easy-to-use graphical screen/form painter. The second is to help developers familiar with a top-down, procedural development style understand the relationship between what the user sees on the screen and events (i.e., the ability to tie methods to objects). The third goal is to provide a way for V6 users to migrate to V7.

Start with the Database or the Interface

To fully understand the integration and flexibility Progress has built into V7, we will summarize the different ways the developer can go about creating applications. Typically, the developer starts with an existing database, selects one or more database fields, puts them in a window, and then customizes the layout and appearance. In this case, fields inherit all their default attributes from the data dictionary—format, label, help message, and graphic visualization (the "view-as" phrase). These defaults can be easily changed.

The developer can also add a widget to the screen first and then associate a database field with the widget. To add an element, such as a command button or slider, to the screen, you simply select the element from a palette of choices and drag the selection to the desired location. You can then double-click on the interface object to open the property sheet and define it further, including the selection of a database field for display in the widget. In this case, a widget you have defined in a window will override the default visualization for the connected database field. Developers will often "mix and match" these options, interactively, putting database fields on the screen and customizing their appearance, creating widgets on the screen and then connecting them to existing database fields, or creating new fields.

Another option is to prototype the application by designing the user interface without any existing database. Fill-in fields can be assigned a data type, sliders can be given a minimum and maximum, etc. When you are done prototyping, you do not have to throw away the

prototype. You can simply open an existing database or create a new one and then associate fields with the interface objects.

Progress supports any mix of database fields and variables on a frame or window. The developer can also toggle between window design and running the application using the "Run" icon. Although the UIB doesn't know anything about database triggers, the application, when run, will also enforce such triggers.

Using Events

In the UIB, the developer can click on an object and get a list of associated events (Progress preselects the event the user is most likely to want). The developer can then attach 4GL code (a trigger) to any event active for the object in the window. Beyond the need for defining user interface triggers, the UIB also helps developers define global application variables to coordinate activities among several active application modules. The UIB provides a Section Editor facility to assist developers in structuring their application sections and controlling complex application processing.

The UIB Generates Modifiable 4GL Code

The UIB includes a number of 4GL code-generation capabilities to build default forms and menus and to generate file-maintenance screens. Once the UIB automatically generates this logic, the developer can customize the resulting 4GL statements to meet specific application requirements. Code-generation capabilities can be extended because developers can add their own code and forms templates to the development environment. All the applications produced by the UIB are stored as standalone Progress 4GL procedures (commented code) in operating system files. The developer can change and maintain these procedures using whatever tool is appropriate—a text editor, a source-code control system, or another tool of choice. Even after modifying the application, the UIB can read in the application definitions and allow the developer to continue working in the UIB.

Cut and Paste between the UIB and Procedures

Everything done using the UIB can also be done using the Progress language. One nice benefit of this is the ability to cut and paste widgets and code between the UIB and the Procedure Editor. If you cut or copy a graphical widget from the UIB, pasting it into a procedure produces the 4GL code necessary to define the widget. Pasting the appropriate code from a procedure into the UIB results in a graphical widget.

UIB Is Graphical Only

The UIB runs only on graphical workstations, but it can be used to create either GUI- or character-based application interfaces. In the first release of V7, Progress documents a set of guidelines for deploying an application in character mode, such as, all buttons should be one character in height, and the default window should closely map to an 80-by-24 terminal screen size. In the next V7 release, Progress will automatically enforce these rules and present the user interface as it would appear on a character terminal.

Progress Supports Dynamic Widgets

Another area where Progress has made innovations is dynamic widgets—the ability to manage a dynamic structure by creating objects as you need them at runtime, rather than at compile time. This is how Progress wrote its UIB component. An example is a database containing organizational information and relationships. The developer doesn't know in advance how many employees there will be or the shape of the organization at any point in time. In this case, Progress would allow you to define a graphic element for every person in the organization, lay it out properly, and assign attributes. Another example is network management (e.g., the number of nodes associated with a file server). According to Progress, dynamic widgets are optimized, and the overhead is not expensive.

The Progress User Interface Builder

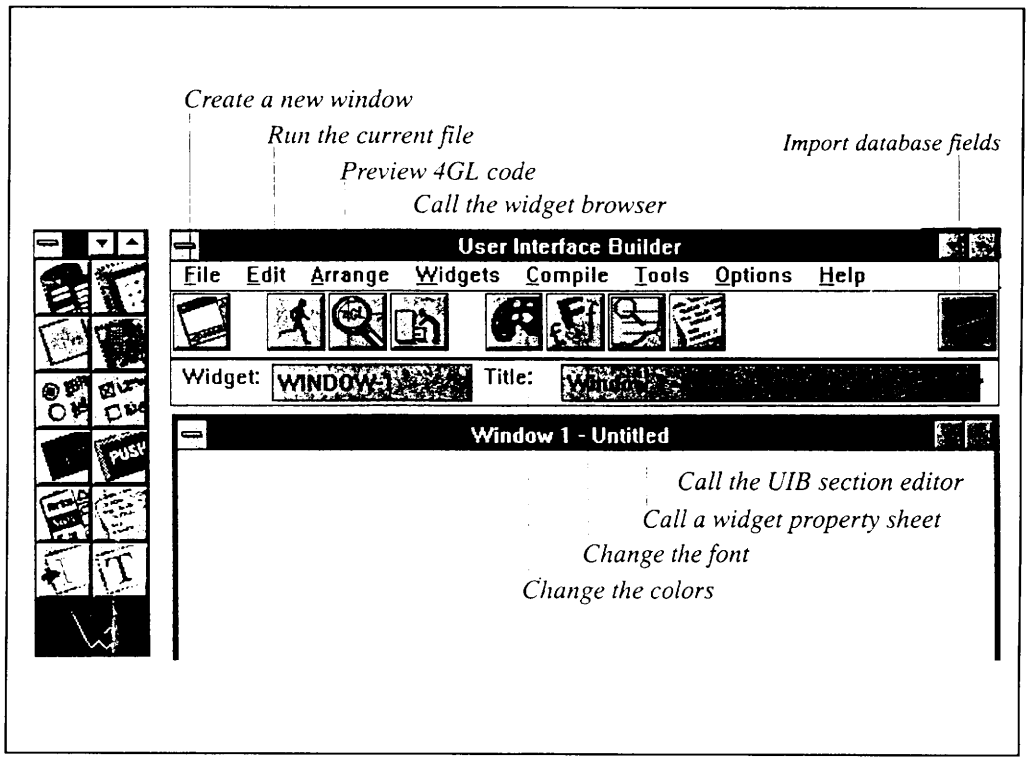

Illustration 5. The Progress User Interface Builder is a graphical screen painter for designing an application's user interface. The widget palette on the left can be used to drag and drop widgets onto a window. The widget can then be connected to a database field. Alternatively, the database field can be added to the window directly, and it will assume the default visualization specified in the data dictionary.

Translation Manager

Translation Manager Facilitates Global Deployment

The Translation Manager is a significant differentiator for Progress. This component of V7 automates, as much as possible, the manual process involved in translating an application into multiple languages. It also allows the language to change dynamically at runtime for labels, help messages, screens, error messages, and data formats. It can handle simple spelling and business-terminology differences (check vs. cheque, for instance). Yet the application source code itself is the same for every language. This is another example of Progress's emphasis on making application deployment easy and separate from development.

Translation Is a Deployment Decision

The translation process works like this: The developer first creates the application in his or her native language. Then the Progress Translation Manager automatically extracts text strings from the application code and presents them to whoever is doing the translation. This person enters the equivalent text strings for other languages as appropriate. (Progress helps here with the option to see text in or out of context.) The point is that the translator doesn't have to be an experienced developer. The application is compiled in multiple languages (the original source code plus a database containing the translations) and runs in multiple languages. The Progress memory manager pulls in only the language in which the user is working.

This is a powerful concept that recognizes the reality of the global organization, and it will appeal to large organizations and VARs deploying the same application in multiple

geographic locations. We are not aware of other products that offer a front-end tool to facilitate translation into multiple languages.

Progress Provides Migration Utilities to Assist the Developer

Although V7 can run V6 applications unmodified and V7 applications can run against a V6 database, most customers will want to move applications into the V7 environment. FreezeFrame is a utility designed to capture all form information from a running V6 application. It essentially takes a snapshot of the application and constructs form definitions in a format that can be used in the graphical environment. With FreezeFrame, V7 can open V6 applications, pull out form-language statements, and build windows the developer can modify and enhance. Progress developers will use FreezeFrame in two ways:

- To provide a graphical user interface without rewriting any code or restructuring the application. One example is replacing a text fill-in character field with a scrolling editor. In this option, the developer is only changing form (frame) definitions and is not creating application objects at the window level.

- To change the application appearance and restructure all or part of it as event driven. Here, FreezeFrame captures form information and creates a default event-driven framework so the developer can start cutting and pasting code from the V6 application into the associated events. FreezeFrame will also convert scrolling menus to buttons, logical fields to toggle boxes, etc.

The Progress RDBMS Adds Triggers

Triggers Greatly Enhance Existing Data Validation

Progress V7 now has database triggers. Triggers can be used for a wide variety of purposes, including cascading referential integrity, record-level security, and complex validation logic. However, there are major differences between the way Progress has implemented triggers in V7 and the way other vendors have implemented triggers.

Progress has always had the ability to define limited validation logic in the data dictionary at both the table and field level. Database triggers remove these validation limitations, giving the developer access to the full Progress 4GL for writing both data validation routines and procedures that execute automatically when the database is accessed or changed. Adding triggers to Progress greatly enhances its appeal to developers.

Triggers Support Insert, Update, and Delete Data Operations, Plus "Find"

At the table level, V7 supports four types of triggers: *create*, *write*, *delete*, and *find*. The "create" trigger fires before a new row is created in the table, allowing the developer to assign default field values before the record is displayed to the user, perform security checks, create an audit trail, etc. The "write" trigger is fired before any row is written to the database, covering both inserts and updates to the table, and the "delete" trigger is fired before a row is deleted from the table. Progress also supports an "assign" trigger for each column in a table. This fires anytime the user updates the value in the column. All of Progress's major competitors—Oracle, Sybase, Informix Software Incorporated (Menlo Park, California), Ingres Products Division (Alameda, California), and InterBase Software (Bedford, Massachusetts)—support triggers in one form or another on insert, update, and delete database operations.

The "find" trigger, on the other hand, is a powerful construct not found in competitive products. A find trigger executes every time a record is read in the table, before the record is given to the user. (Only records satisfying the search conditions—e.g., a "where" clause—will cause a find trigger to fire.) This allows the user to know every time a record is touched in the table, regardless of what the application is doing to the record, and it is aimed at

satisfying customers' database auditing requirements. It is important to use the find trigger with care, however, to avoid possible performance degradation.

Session Triggers Augment or Override Schema Triggers

Progress also differentiates between schema and session triggers. Schema triggers are defined in the data dictionary. Session triggers are defined in a Progress procedure, and they are in force only as long as the procedure is active. Session triggers can be used to run a V7 application against a V6 database (making database triggers part of the application, since V6 doesn't support them), to turn off database triggers, or to handle validation or security differently. Session triggers either augment or override schema triggers, within defined security and permission settings.

Triggers Fire at the Row Level

All of these triggers are row-level triggers, firing either before (in the case of "create" and "find") or after each row. Progress does not support triggers that fire once for an entire SQL statement, regardless of how many rows the SQL statement affects. (Although it supports SQL for data access, Progress is, in fact, a record-oriented database manager, and its trigger design reflects that heritage.) Database triggers have access to any data, and there are no limitations to the data that a trigger itself can change.

Database Triggers Are Executed on the Client

The developer writes a database trigger in the 4GL. A reference to the trigger is stored in the data dictionary, but the trigger code is stored in a file outside the database. In fact, triggers can be organized and located as appropriate for a specific computing environment. Unlike triggers of other products, which always execute on the server, Progress triggers are actually executed on the *client* side. Progress does this for two reasons: (1) It does not yet support any form of server-based procedure processing (e.g., stored procedures) in V7, and (2) running triggers on the client is often desirable to provide faster feedback to the user.

Progress Triggers Have Advantages

The Progress implementation has several pluses:

- Triggers can do everything they can do in other products, and they provide a single point of maintenance regardless of the calling application.
- There is no limitation on the language statements that can be included in a trigger. The developer has the full 4GL available. The only difference is that triggers are automatically invoked rather than explicitly called. In addition, the Progress Debugger can be used to debug trigger code.
- Front-end execution provides immediate feedback for the user. This can be a major issue for developers when triggers and stored procedures are executed in the server. It avoids making the developer weigh the trade-off of application responsiveness versus storing triggers and procedures in the server. In Progress, application variables can be shared with the trigger procedures without having to pass them to the server.
- Processing is offloaded from the server to an increasingly powerful desktop client.
- Triggers are portable across DBMS products supported by the Progress DataServer architecture in addition to Progress itself. For customers using a Progress application against a non-Progress database, moving the data into a different back-end server does not mean rewriting the triggers in another language. It is important to understand, however, that, in these cases, Progress triggers only control database operations executed via the Progress DataServer. Access to the non-Progress database from any other source will bypass the Progress trigger processing.

The Potential Downside Is More Network Traffic

On the other hand, Progress triggers have some limitations. Because triggers currently execute on the client, the client has to process complex operations, such as cascading referential integrity constraints or calculating complex totals after receiving all the relevant records. This can mean extra network traffic in order to give the client the necessary data with

which to execute the trigger. But it doesn't limit the amount of application code that can be stored in a central location, since the logic can all be in the Progress trigger code. The trigger is simply executed on the client.

Stored Procedure Support Will Allow Triggers to Execute Anywhere

Stored procedures are coming in a future release. Once procedures are stored in the server, Progress will give the developer the choice of executing trigger/stored-procedure logic in the Progress server or in the application. The goal is to be able to recommend an optimum execution strategy based on what the trigger does. Progress is unclear on how it will handle server-based triggers for non-Progress databases but is considering alternatives for this.

Progress Triggers Are Powerful and Flexible

The bottom line is that Progress triggers are very powerful, giving the developer a great deal of flexibility. Progress triggers are precompiled and preoptimized. They are recursive; they can cascade, access both local and distributed data, and be activated and deactivated; and they are under transaction management. The default action on trigger failure is to roll back the triggering statement, but the developer has the option to roll back the entire transaction. We should note here that Progress supports automatic two-phase commit (2PC) for distributed transactions, and this applies to trigger processing as well.

Client/Server Optimization

The Progress Server Handles Joins and All Selection Criteria

V7 introduces several significant improvements to reduce client/server network traffic:

- The ability for the server to send multiple records to the client in a single network packet.
- Reduced connection time when a client connects to a server.
- Improved data-retrieval strategies, which include enhanced query optimization and execution, particularly for decision-support applications in a client/server environment. First, all joins and record selection in queries now take place on the server. Second, Progress now takes advantage of multiple indexes concurrently to resolve a query that includes an "and" or an "or" data comparison. In V6, Progress can use only a single index per table; therefore, the client has to check additional criteria if they are not covered by existing indexes, creating additional network traffic.

The Progress RDBMS needed these enhancements to provide competitive performance in a client/server distributed environment.

What's Missing and What's Coming

GUI Compromises

Progress has made a few compromises on the GUI side in the initial release of V7, primarily to maintain compatibility with V6 and to support deployment on character terminals. V7 does not yet support a multiple document interface (MDI) on Windows or certain GUI features like combo boxes and sculpted (3D) fields. Progress plans to implement support for both combo boxes and sculpted fields in the immediate follow-on version of V7. Progress's goal is to support a combo box widget in X Windows/Motif and in character-based terminals as well.

There are two points to be made about MDI support. The first is that MDI is essentially a Windows-only issue. It doesn't make as much sense in the Motif environment, since Unix is multitasking and there tends to be more screen real estate available on Motif workstations. The second point has to do with representing database transactions in the user interface. Most DBMS servers, including Progress, only allow an application to have a single transaction active at one time. (InterBase is the only product we are aware of that supports multiple concurrent transactions for a single user.) Therefore, opening lots of windows (e.g., sales orders) on the screen concurrently means that they are all part of the same transaction. All of

them must commit or none of them will commit, and representing that visually to a user who wants to open and close multiple windows independently is difficult. So there are trade-offs between complete flexibility and the need to maintain database integrity through transactions and context for the user.

To avoid confusing the user, Progress provides the developer with three options in this situation: to allow the user to open only one sales order at a time, to invoke a Progress utility that detects when the user goes to open a new window when a transaction is already started and tells the user to commit or cancel the existing transaction first, and to run multiple instances of the application (multiple clients, each of which can have its own independent transaction). The negatives here are memory requirements and the fact that each is a separate user to the database, creating potential concurrency issues within the application itself. The issue that Progress wants to resolve is giving the user a flexible interface while maintaining transaction integrity—the "undoableness" of the transaction. The company will solve this in the future by implementing multiple concurrent transactions, each of which can be separately committed. While other tools vendors support MDI, all of them have the challenge of dealing with the database transaction issue.

There Is Still Much to Come

TOOLS FUTURES. Progress already has a long list of future enhancements for V7. We have mentioned some: implementing combo boxes and sculpted fields, and the ability to have multiple transactions open concurrently. Other futures include a repository to manage application objects and their relationships in addition to database objects; an application manager with check in/check out and version control for multideveloper environments; a graphical report writer to handle heterogeneous data and a wide variety of report formats; two-way CASE bridges (these are available for V6 but cannot yet propagate a V7 data dictionary to/from third-party CASE tools); object-oriented features such as inheritance and class libraries; and a graphical version of Progress Results. (Currently, the character version of Results has been recompiled for V7 but does not have a graphical interface.)

THE SERVER SIDE. In addition to stored procedures and declarative integrity constraints, Progress is missing other server features present in competitive products, such as binary large objects (BLOBs), data replication, a cost-based query optimizer, more sophisticated security options (like groups or roles), and support for multibyte character sets. All are currently under consideration for future versions of Progress, as is a parallel data query capability.

New Packaging Separates Tools and RDBMS

Progress has finally unbundled its tools from the database server in V7, a step the company agonized over for some time. It has eliminated language runtimes as a separate product, packaging the client runtime with either the database server or client networking products.

Progress also recently moved to user-based pricing. The company has placed every platform into one of four pricing tiers (instead of 26 machine classes). As an example, SCO Unix on a 386/486 is a Tier 2 platform, with pricing for 1 user, 2 to 8 users, 9 to 16 users, and 17 to 32 users. Beyond that, the customer buys a license for each additional 16 users or buys a single license for unlimited users. A nice feature of the Progress plan is that customers can mix and match user counts for development licenses, deployment licenses, and DBMS licenses on the same system (e.g., 1-user development license plus a 32-user DBMS license).

Progress thinks its own RDBMS will continue to be the one of choice for many customers because it is relatively inexpensive and requires less administrative overhead than competitive products such as Oracle or Sybase.

Competitive Assessment

With V7, Progress now has two sets of competitors—the independent RDBMS vendors, all of whom have their own development toolsets, and third-party vendors of database-independent development tools.

RDBMS Vendors Are behind on Migration Strategy, Database Independence, Language Extensions

Informix, with a well-respected 4GL, has traditionally been the closest RDBMS competitor to Progress on the tools side. But Informix is lagging behind in offering a GUI development environment for its Informix/4GL. We do not expect to see Informix/4GL++, with graphical and event-driven extensions to the 4GL and full graphical application deployment, until mid-1994. We see this as a real competitive problem for Informix. (Informix has just introduced a graphical screen painter in its new Informix-4GL for Windows product.) Informix is also creating some confusion in the marketplace with the introduction of HyperScript Tools, a separate development toolset which uses a different language than the Informix/4GL family of products.

Sybase and Oracle have both been relatively weak on the tools side in the past. Oracle is rolling out its new Cooperative Development Environment (CDE) toolset—including CASE tools—and touting presentation independence across a wide variety of client platforms (Windows, Motif, DOS, Macintosh, character terminals). However, Oracle's procedural language, PL/SQL, is not a full 4GL capable of handling complex applications without resorting to a 3GL, and it has not yet been extended to handle graphical, event-driven applications. Oracle is currently limited to a forms-based development environment with Oracle Forms.

Sybase has just announced a new object-oriented graphical development tool, Build Momentum, due out in early 1994 (See following article for more information.) Our initial impression is positive. Build Momentum appears to be a competent graphical tool targeted for Windows, Motif, and Macintosh applications, but it is too early to assess Sybase's overall tools strategy. Build Momentum will support forward migration of APT Workbench (Sybase's character-based toolset) applications, but only graphical deployment. One issue for Sybase is that APT has not been a popular toolset. Not many developers are using Sybase tools today, so marketing tools will be an uphill battle for Sybase. Many Sybase developers already use third-party development tools. The other issue is how APT and Build Momentum will integrate with Gain Momentum, a high-end, multimedia development environment. Without a common development language, Sybase is moving toward integration via a common repository in the future.

Currently, Ingres has the strongest graphical toolset of the RDBMS vendors in Ingres/Windows 4GL. But Ingres still lacks a migration strategy in either direction between Windows 4GL and its character-based development tool, Ingres/Vision, a major negative from our perspective. This will change in the future.

None of the major RDBMS vendors have migrated their toolsets to competitive DBMS products yet. The only exception is Unify Corporation (Sacramento, California). Sybase also has Oracle, Ingres, and Informix gateways, and it has access to Oracle via its OmniSQL Gateway.

Third-Party Tools Lack Portability or Functionality

Progress is really targeting third-party tools vendors, such as PowerSoft (PowerBuilder), Gupta (SQL Windows), and Uniface with V7. We think Progress can compete with PowerSoft and Gupta in two areas: portability of both development and deployment (across operating platforms, and with support for character-based as well as GUI clients), and the ability to develop complex, mission-critical applications. PowerBuilder and SQL Windows are Windows-only at the moment and have some difficulty scaling up on large applications. Progress will compete with Uniface strictly on language flexibility and the ability to develop

large-scale applications, since Uniface offers comparable presentation independence for deployment. Uniface's forms-based model for application development is less flexible than Progress's ability to develop applications using either a forms-based or language-based model. Progress also provides the developer with direct access to the full 4GL as well as SQL for transaction control, locking, etc. Uniface generates such commands on behalf of the application.

Progress Has Key Differentiators...

Progress has several strengths to leverage against its competitors. One is the ability to handle very complex applications without the need for a 3GL. Informix can compete well here but doesn't offer the portability and database independence that Progress does. Progress developers also stress Progress's relatively easy learning curve and the intuitiveness of the development environment. Progress provides an impressive combination of high-level defaults and low-level control over application processing. Another strength is portability across character as well as GUI clients and proprietary as well as open operating platforms. A third plus is the broad portfolio of VAR applications. This is important for customers who want to buy off the shelf and build applications in the same environment. A single development staff can address the needs of the entire organization. (This is another area where Informix is strong.)

Other Progress strengths include the ability to use the same language for everything—user interface, procedures, triggers, reports, etc.; the ability to generate (and debug) batch jobs and production reports that don't have a user interface attached (products like PowerBuilder cannot do this); and emphasis on features that ease application deployment for the developer (examples here are the Developer's Toolkit and the Translation Manager).

...But Needs to Keep up on the GUI Side...

Progress has to be careful that it doesn't allow concern for compatibility with V6, X Window, character terminals, or for its current customer base to blind it to the need to keep up with PowerBuilder and the others on the graphical development side. Progress must implement full graphical support as soon as possible, especially on Windows, to satisfy customers looking for graphical-only applications. This includes support for a multiple document interface, combo boxes, etc. Implementing these features on other platforms where they don't exist today will be a plus for the Progress environment, creating a true superset of functionality across clients.

Another competitive issue for Progress is the fact that some other third-party tools support a much broader range of data managers than does Progress. Progress also needs to articulate its strategy for incorporating object-oriented features, such as inheritance and class libraries, and multi-developer support in the future. Other vendors are moving fast in these areas, with many products already beginning to offer these capabilities.

...And Improve Market Visibility

MARKETING AND VISIBILITY ISSUES. Progress is well aware of the need to increase its name recognition vis-à-vis much larger and more well-known competitors like Oracle and Informix. Selling primarily to VARs can often mean that the VAR and its application have a much higher profile than does Progress as the underlying platform. VARs also benefit from better visibility for Progress, since they often find themselves in direct competition with Oracle et al. for business. Increased market visibility becomes particularly important as Progress competes more directly on the Windows platform.

The company's overall strategy is to move into a broader market, increasing penetration in the corporate MIS arena while maintaining a strong VAR program. Progress recently established a separate end-user sales force to focus on the corporate MIS market. The company intends to better advertise its tools for effectively developing and deploying applications (an issue for large users as well as VARs), its database architecture and support for heterogeneous distributed network computing, its strong positioning across major server platforms of interest to the Fortune 1000—Unix, VMS, NetWare, and the IBM AS/400—and the breadth of VAR applications available. V7 will definitely help Progress improve

Competitive Assessment

visibility, since the company can now compete directly with third-party vendors of database-independent tools.

THE BOTTOM LINE. Overall, Progress is aiming to be the best value for the money compared to the competition, a low-cost option in terms of purchase, deployment, administration, and operating platform/resources. The company recognizes that the tools business has more potential for growth than the RDBMS market. As Joe Alsop, Progress president, says, "Growth in the database market is peaking. The tools business could build another $1 billion company [à la Oracle], and the 4GL will be the enabling technology and foundation for this."

Summary

Successfully Balancing a Difficult Set of Objectives

Progress has taken a set of difficult and often-conflicting objectives and come up with a competent solution in V7. Most of its competitors have avoided the challenge of a graphical development environment that supports both graphical and character deployment, lamenting the fact that it requires too many compromises on the GUI side. But there are still many customers who need flexible deployment options. Progress has, in fact, made some of these compromises in the interest of maintaining compatibility and providing a migration path, and we are confident that the company will ultimately allow the customer the best of both worlds. Graphical-only customers will get full support, while character-too customers will continue to have to make sacrifices on the GUI side to gain compatibility and a single set of source code.

The Need for Visibility and Differentiation

The onslaught from fully graphical third-party development tools that can access multiple data sources is fierce. Since many are still Windows-only, Progress can exploit its portability across both Windows and Unix clients and its support for both character-based and GUI deployment. The competition on the server side is rolling out enhancements at a fast pace as well, and Progress must keep up. On the other hand, customers desperate for database features Progress doesn't have yet can always go the DataServer route and use a non-Progress back end. Progress will probably continue to lag behind the other database vendors in implementation of state-of-the-art server functionality with its emphasis on the development environment and the DataServer architecture as a backup option.

Progress Software Corporation
14 Oak Park
Bedford, Massachusetts 01730
(617) 275-4500

Progress has a lot going for it—comprehensive development and deployment tools, a strong RDBMS, happy developers, and a broad range of available applications. The company is right on target with its emphasis on supporting distributed client/server computing. Its overall strategy for heterogeneous data access and client API support is, in many ways, similar to that of Sybase. The separation of the 4GL from the RDBMS further strengthens the developer's position by offering application development tools that are independent of the underlying DBMS.

Next month's *Open Information Systems* will address Novell's Unix Strategy.
For reprint information on articles appearing in this issue, please contact Donald Baillargeon at (617) 742-5200, extension 117.

Progress still lacks market visibility for both the company and the product. And waiting too long or moving too slowly to fix this problem may mean missing a critical window of opportunity. Progress has worked hard to position itself well technically and has filled important gaps with V7. Now the company needs to carefully plan and implement a marketing strategy to effectively differentiate its offering.

If it can succeed in putting a graphical face on a still elegant development environment, add robust functionality to support the development of large, multi-developer applications, put state-of-the-art database functionality underneath (Progress or not), and support heterogeneous clients as well as servers, Progress will continue to be a popular choice for customers seeking an option for the development of serious, complex applications that meet real business needs.

Open Systems: Analysis, Issues, & Opinions

FOCUS: DEVELOPMENT TOOLS

Sybase Introduces Graphical Tools Strategy and a Windows Product

Having introduced a multitude of System 10 server products over the past year, Sybase Incorporated (Emeryville, California) is now extending its enterprise client/server model to encompass the development environment on the front end. The company recently articulated its long-term tools strategy and announced a new family of client/server tools called Momentum. The three components of Momentum are:

- Build Momentum, a new graphical, object-oriented application development environment that will be available in early 1994. Key features are a 32-bit multithreaded kernel (even on Windows), forward migration for existing Sybase applications, presentation independence across GUIs, and facilities to manage multideveloper projects.

- Gain Momentum, a high-end multimedia development environment. A new version of Gain, 2.0, now provides integration with both Sybase SQL Server and Oracle data.

- Enterprise Momentum, a model-oriented development environment for large, enterprise-wide applications, due in the second half of 1994. Key features are an integrated repository (the Meta Server) and automatic generation of applications from enterprise-wide models (a process Sybase calls Applications from Models).

Sybase clearly recognizes the huge potential market in development tools and is now ready to elevate its offerings to a new competitive level.

Emphasis on the Server Side

Over the past two years, Sybase has focused its efforts primarily on the server and connectivity side of its architecture. Components here include the client/server infrastructure (the Sybase client/server APIs); the back-end Sybase System 10 database server products—SQL Server and Backup Server, OmniSQL Gateway, Replication Server, Navigation Server; and system administration tools. (See *Open Information Systems*, Vol. 8, No. 3, March 1993, for an in-depth report on Sybase System 10.)

On the front end, or client side, the application development tools area has traditionally been a weak one for Sybase. This is one reason the company has been aggressive in developing partnerships with third-party tools vendors. A wide variety of third-party products (over 600) currently run against the Sybase SQL Server.

Sybase has also bolstered its tools offerings through acquisition. SQL Solutions brought in report-writing and administrative tools (1990); Deft brought CASE (1991); and Gain Technology Incorporated (Palo Alto, California) brought Gain Momentum for multimedia application development (1992). The big question has been how Sybase would tie together all these different toolsets into a cohesive product line, one that covers the required functionality while providing a migration path for users. We have had concerns about the company's ability to catch up on the development tools front while, at the same time, effectively integrating all of these acquisitions on a timely basis. The recent announcement indicates to us that Sybase is doing more and doing it sooner than we expected in the tools area.

Sybase Tools Address Different Application Environments

Sybase's strategic approach is to provide different toolsets—the application productivity tool (APT) Workbench, Build Momentum, and Gain Momentum—each addressing a different end-user application environment. Enterprise Momentum, rather than being a specific development tool, addresses a different approach to building applications, taking a top-down, model-oriented approach rather than a bottom-up, prototyping approach.

The Sybase APT Workbench will continue to provide a character-based forms development environment for *back-office* applications, where screen presentation is less important than the capability of the 4GL underneath to handle complex applications. The need for heads-

down data entry as part of online transaction processing (OLTP) applications (e.g., order entry) does not go away simply because graphical user interface (GUI) has arrived. And there are lots of companies that are not ready to replace hundreds of $200 terminals with $1,000+ PCs.

Build Momentum, which adds graphics to forms in presenting data to the user, is targeted for *front-office* applications. As organizations push business processing out from the back office in an effort to get it closer to the customer, different requirements emerge. A graphical interface becomes more important, and desktop workstations replace dumb terminals. With a GUI-based environment, users can understand more of the business process embedded in the application and can often do more business problem-solving at the point of data capture. However, the performance and functionality of the 4GL and transaction control are still important in these applications, since they tend to be a mix of OLTP and decision-support environments.

Because Build also supports APT SQL, it can provide a migration path from APT Workbench applications, it can be used to create graphical back-office applications, and it can support the customer's need to evolve back-office applications into front-office applications.

Gain Momentum is then positioned for *extended-office* applications. These are applications targeted at "volunteer" or "elective" users, users who do not have to use the computer to get the job done, but who will if it is easy and intuitive enough without requiring training or documentation. Such applications are often external, residing in the hands of customers, trading partners, or prospects. An example is someone renting a car and faced with the choice of waiting in line or using a graphical application on a touch screen.

We will first describe Build Momentum and the newest release of Gain Momentum, both of which are key building blocks in Sybase's tools strategy. Then we will cover Enterprise Momentum.

Build Momentum

Build Momentum is a GUI-based, object-oriented, event-driven development environment. Build Momentum itself is a Build Momentum application, demonstrating the power of the underlying language.

MULTITHREADED KERNEL. A key differentiator for Build is its multithreaded architecture. As a multithreaded tool, Build can generate multiple tasks in parallel even on Windows. Consider, for example, a user entering a new customer order on the screen. Entering the credit card number spawns a task to validate the credit card number in the background, including a lookup in a DB2 database on the mainframe. The user can continue entering data at the same time the validation process executes. Build does this by optimizing the use of its single connection thread to the Windows operating system. It still looks like a single application to Windows, but it uses the thread for multiple concurrent tasks.

Other differentiators are Build's 32-bit architecture and dynamic self-tuning capability. Both contribute to high performance with large applications (e.g., lots of objects on the screen, lots of forms and processing). The 32-bit architecture provides greater address space, even on 16-bit Windows, and should make the system perform better. A Build application optimizes its own performance at runtime by monitoring its execution environment, in particular the amount of memory available. Typically, Build executes in optimized "p code," which is much smaller than the compiled, machine-code version would be. Thus, less memory is needed to run the application, and code must be paged in and out of memory less often. This allows a Build application to have a relatively small footprint. However, if Build sees that free memory is available, it will automatically compile the most-used components into machine code to improve performance.

One of Build Momentum's core technologies was acquired with Wyvern Technologies (Ottawa, Ontario, Canada), a small Canadian company, this summer. This includes the multithreaded kernel, 32-bit architecture, and a powerful toolset with utilities for rapidly building development tools. Sybase sees this as its key differentiator, the one that keeps Build from being a "me too" GUI development tool. (See Illustration 1 for an overview of the Build Momentum architecture.)

OBJECT-ORIENTED GRAPHICAL DEVELOPMENT. Build Momentum provides a wide variety of predefined objects for the developer, including the expected graphics widgets such as buttons, fill-in fields, radio buttons, toggle boxes, etc. Build also has what are called "smart" objects, which know something about the database and provide standard database processing that is context sensitive depending on the contents of the window to which they are attached. One example is the familiar "First," "Last," "Next," and "Previous" buttons for navigating around database records on the screen. Another is buttons for adding, updating, and deleting records, each of which has certain default error-coding already attached.

The developer can also create custom objects using the Object Momentum language—new class libraries, new class definitions, etc. Eventually, Sybase will provide

access to third-party class libraries as well. Build Momentum supports both visual inheritance and programmatic inheritance. Multiple inheritance is supported visually.

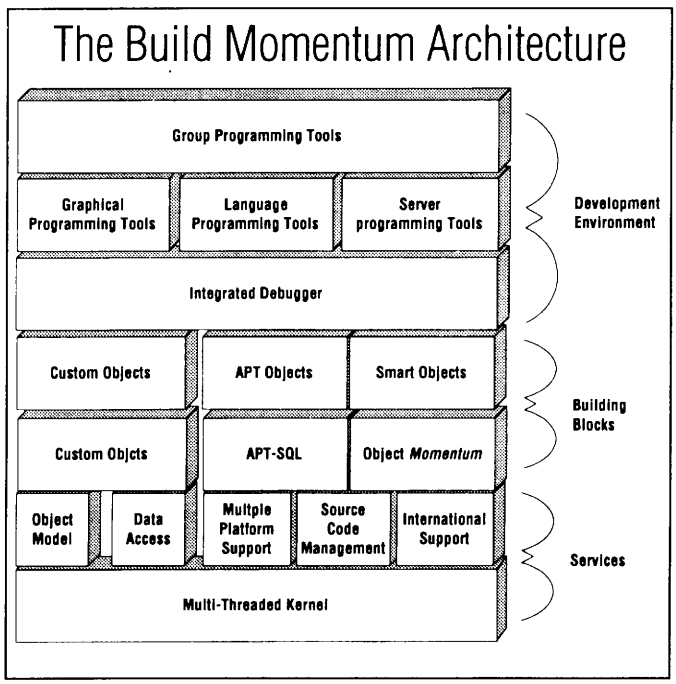

Illustration 1. Build Momentum has three major components: the development environment, building blocks for applications, and underlying services.

Object Browser. The developer has two ways to browse through objects using the project browser. One is a single-layer view of all objects with the ability to browse by type of object (window, button, etc.). The developer can also create a hierarchy of application objects. For example, the highest level could be "Financial Systems," with the next level including accounts receivable, accounts payable, general ledger, and the like. Specific objects can be associated with each layer.

Advanced Mode. Build Momentum supports two levels of developers. The first is the typical application builder, who creates an application using the existing environment. This developer uses standard graphical objects and attaches code to existing events as appropriate; there is no need to create new objects or events or to understand object-oriented techniques as such. The second level, more sophisticated customers and value-added resellers (VARs), may need to redefine an object or method or to create new ones. An example here is adding a triple-click event to the standard button object. These developers can flip a switch to turn on advanced mode, revealing the abilities to create class libraries and new events and to modify events predefined by Sybase (e.g., the "Next" button).

All developers, regardless of level, can work simultaneously on one application using the shared repository (see "Repository for Team Development" below). Everyone benefits from the underlying object-oriented technology, but only when using object-oriented techniques directly does the developer have to explicitly understand the details of object-oriented programming. (See Illustration 2.)

Languages. Build Momentum supports two languages, the current APT SQL 4GL that is part of APT Workbench, and Object Momentum. Object Momentum will be the language of choice for new application development. Object Momentum is similar in syntax to Visual Basic with if/then/else constructs, but it is object oriented and more extensible than Visual Basic. Object Momentum supports both a Visual Basic-like longhand and the "squiggly-bracket" shorthand familiar to C programmers. We should note that Object Momentum is not the same language as Gain Momentum's scripting language, the Gain Extension Language (GEL). Object Momentum and APT SQL can be mixed in an application. Build Momentum includes a symbolic debugger with break points, if clauses, the ability to monitor variable values, and other debugging functions.

SERVER PROGRAMMING TOOLS. Build Momentum provides full access to server-based code via method editors for writing stored procedures and triggers (using Transact-SQL). With its graphical utilities for manipulating the server environment, Build also can serve as a database administration tool for developers. Examples would be drag-and-drop for moving databases among servers, creating new tables, adding columns to tables, and the like.

REPOSITORY FOR TEAM DEVELOPMENT. Build Momentum has a shared repository for team development with an integrated source-code management system (check in/check out), version control, and shared libraries. A developer can also create a local version of the repository, check out objects, work on them locally, and then return them to the central repository.

The repository only works with Build Momentum and is not Sybase's long-term enterprise-level repository, the Meta Server. Sybase will ultimately have three repository products: Build Momentum, Gain Momentum (stored in Objectivity), and Enterprise Momentum (targeted at storing business models and rules). The company prefers to call the Build and Gain repositories "application object stores" because they store objects that make up Build or Gain applications only. A true

repository would be consistent across all of these products. According to Sybase, Build and Gain will always have their own object stores. The goal is to provide the ability for all of the tools to share objects.

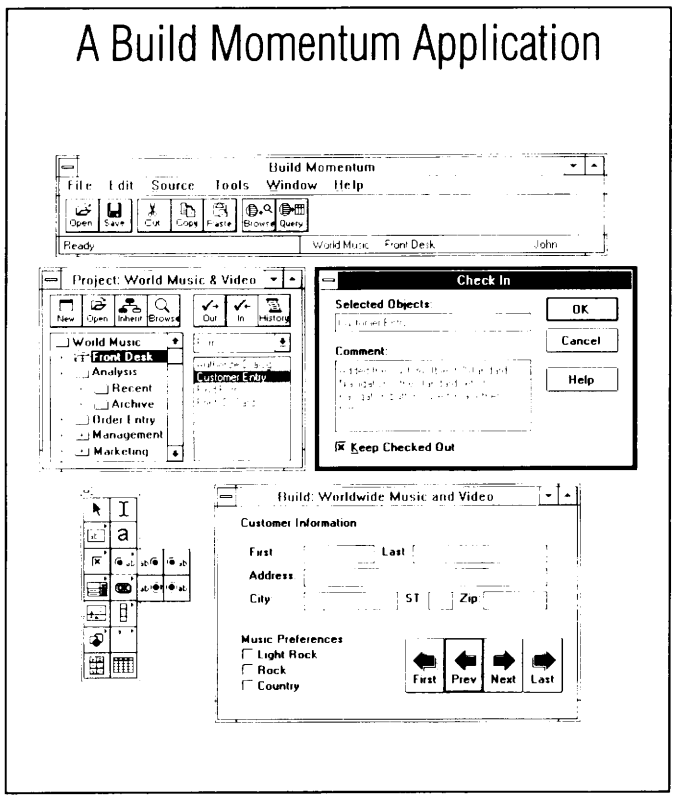

Illustration 2. This illustration shows several components of the Build Momentum development environment: the main Build menu/toolbar (top), the object browser (middle left), the checkin/checkout dialog box (middle right), the widget palette (bottom left), and a sample application window.

ACCESS TO HETEROGENEOUS DATA. Build Momentum applications access non-Sybase data via an ODBC driver (included in the first release) or OmniSQL Gateway (RMS and C-ISAM in addition to Sybase SQL Server data plus optional modules to access DB2 and Oracle). Sybase will certify the ODBC driver against specific data managers over time. Sybase stresses that both of these access options are appropriate for decision-support applications but not for high-volume OLTP access to the database. In the future, Sybase plans to offer direct access (via the Oracle Call Interface) to Oracle data from Build Momentum applications.

MULTIPLATFORM GUI SUPPORT. Initial platforms for Build will be Windows and Windows NT, with general availability in the first quarter of 1994; followed by Macintosh and Unix/Motif, with general availability in second quarter 1994. All GUIs will be available for both development and deployment with presentation independence across GUIs. Sybase's approach is to provide a superset of GUI functionality, adding to specific environments where necessary. One example is adding a Windows drop-down list to the Mac. Build Momentum will cost $2,000 to $4,000 per developer license.

MIGRATION PATH. Build can run APT Workbench applications, but it provides forward migration only. Build is strictly graphical, offering no character-based development or deployment. Sybase's philosophy is the same as that of other vendors who have chosen to separate character and GUI development and deployment: Maintaining compatibility on the character side requires compromises on the graphical side. Sybase does not want to make these compromises and may lose some customers who need to write an application once and deploy it in both character and GUI.

The good news is that current APT applications can be brought into the Build Momentum environment (unlike Ingres, which still has no migration strategy between its character-based Vision applications and Ingres Windows 4GL). The migration is not painless, however. Running a utility moves APT Workbench forms into Build, and the developer can reuse up to 90 percent of the existing APT SQL code as is. Some APT functionality cannot migrate, including nested groups and navigational code (navigation is not hard-coded in a GUI environment), and must be manually modified. The other "good" news for Sybase is that there isn't a huge volume of legacy applications written in APT Workbench, so the magnitude of the problem is not as great as it is for vendors like Informix and Progress.

As a note, with Build, APT for Motif goes away, as do the plans for APT for Windows. These were products designed to deploy an APT application with a graphical interface. Sybase sees character applications as a rapidly diminishing portion of the market, and believes that having a common language (APT SQL) between APT and Build is sufficient.

Gain Momentum

Like Build, Gain Momentum is an object-oriented, graphical application development. The difference is that the primary design issue for a Gain application is the interface itself. Gain would be targeted at extended-office applications, rather than front- or back-office functionality. While Gain can access SQL Server and do transaction processing, this is a secondary design issue. Build, on the other hand, is geared for OLTP applications that have a pretty interface as well.

POSITIONED AT THE HIGH END. Sybase is positioning Gain at the high end of its application development options in terms of application type and resources required. It is a separate graphical environment geared for sophisticated multimedia applications (applications that require more than forms and graphics, such as audio, video, animation, and others.). Gain is also expensive, starting at $10,000 a seat (down from $20,000), and it requires a 486-class machine with significant memory and disk resources. Gain currently runs on SunOS and Solaris, IBM RS/6000, and HP 9000, with Digital Alpha/OSF1 and Alpha/OpenVMS, Windows NT, with Silicon Graphics (IRIX) coming in the future. Gain applications are portable across Motif and OpenLook, with support for Windows NT and Win32S coming by the end of 1993.

Release 2.0 of Gain, shipping now, has been extended to support database applications. Previously, for example, a developer had to embed SQL statements in the Gain 4GL to access Sybase SQL Server data and then explicitly connect the retrieved data to an application object. Gain 2.0 offers full read/write access to SQL Server as part of the base Gain Momentum product, generating SQL transparently for the developer/user. It also supports direct access to Oracle as an option and concurrent access to both Oracle and Sybase data in a single application. (This is done via nested loops, not heterogeneous joins.) In the future, Gain will support Informix and possibly Ingres as well. Gain could also access SQL and other supported data sources via the OmniSQL Gateway. Release 2.0 of Gain is available on SunOS and Solaris, IBM RS/6000, and HP 9000.

There is no application migration path from APT or Build to Gain at the present time. Sybase's strategy here is one of application integration rather than migration. Because each development environment is targeted for a different category of application, the ability to integrate applications built in these different environments and to share objects among them is more important than the ability to move an application from one environment to another. Ultimately, the Meta Server repository will tie together all the Sybase and third-party tools via shared application objects and data models. However, even in today's world, Build can call Gain and Gain can call Build, so the developer can select the tool that best meets the needs of a particular application component.

Enterprise Momentum

There are two basic approaches to building applications in the market today. One is the "bottom-up" approach, which emphasizes prototyping. This method is also referred to as rapid application development (RAD), or joint application development (JAD), where the developer works closely with the end user. Most tools vendors offer a product for this approach. Build and Gain Momentum fall into this category. The other approach is the "top-down," or model-driven, approach taken by CASE vendors. Sybase estimates that 20 percent of the marketplace needs or already uses tools in this category. As we mentioned, Enterprise Momentum is targeted at this approach, combining object-oriented (which focuses on program objects) and model-driven (which focuses on a two-step process of capturing the analysis and design of the business process and then generating code from that) characteristics to arrive at what Sybase calls a *model-oriented* approach. This essentially treats the business model as an object.

Enterprise Momentum will not be available until the second half of 1994, so few details are available. However, it is based on the concept of "active" models with the ability to regenerate modified applications. Changes to the model dynamically change the application, preserving any screen and 4GL customization. Enterprise Momentum defines three active models: a process model, which includes error handling and impact analysis; a data model, which includes computations and business rules; and a GUI, or presentation, model, which includes event logic and valid data values. Another important aspect of Enterprise Momentum is the ability to develop a specification for both the client and the server and to generate both. The goal here is to separate the logical design of the application from the physical design.

Enterprise Momentum encompasses three product components: an Enterprise Modeler (Sybase's existing Deft CASE product will migrate here), the Meta Server, and Application Builders. The Meta Server will be the repository for Enterprise Momentum. (See Illustration 3.) It will have two APIs, one for the models to integrate non-Sybase modeling and CASE tools, and one for application objects to integrate other application builders and development tools. Meta Server will be the common object repository for Build, Gain, and Enterprise Momentum. In addition, Build and Enterprise will share the same 4GL in Object Momentum.

Enterprise Momentum will be available on Windows, Windows NT, Macintosh, and Motif.

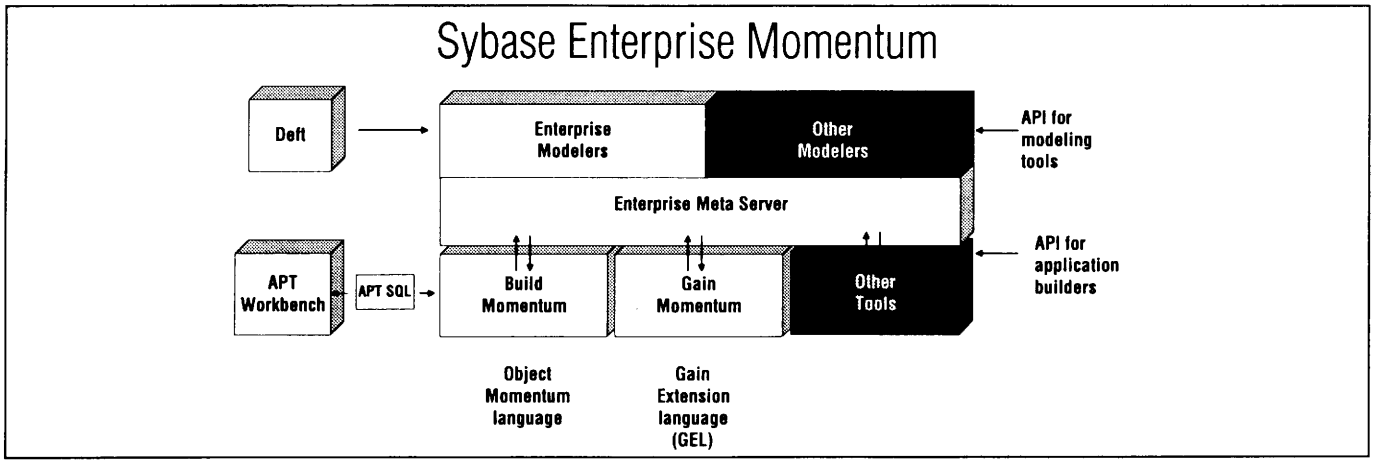

Illustration 3. Sybase's Enterprise Momentum architecture for client/server application development has three components: the Enterprise Modelers, the Meta Server repository, and Application Builders. There will be two APIs for the Meta Server, one for modeling tools and one for application builders.

Conclusion

Sybase is finally starting to weave a cogent architecture out of its disparate set of tools. We were impressed with the quick first look we got of Build Momentum, and the overall strategy seems to make sense. As it has on the server side, Sybase has a clear vision of where it is going and why. The company has acquired some excellent technology on the tools side and obviously intends to make good use of it. It also is limiting its investment in character-based tools, viewing this segment of the market as increasingly less important. This lack of deployment support on the character side (other than APT Workbench), although less of an issue for Sybase than for other database vendors with a stronger VAR following, means that Sybase will not have a satisfactory solution for some customers. But Sybase is focusing on the future and on a tools architecture that can handle the next generation of client/server applications. — *J. Davis*

FOCUS: PLATFORMS

SCO Becomes Windows Friendly

Santa Cruz Organization (SCO, Santa Cruz, California) has announced a new strategy for coexisting with MS Windows, along with a new product that may be the answer to many customers' prayers for better Unix and MS Windows integration. The strategy is called Windows Friendly, and it entails a comprehensive product-positioning as well as delivery plan to make SCO Unix complement Microsoft Corporation's (Redmond, Washington) direction, rather than compete head on. Considering the interchangeability of the two companies' operating systems on the same hardware, SCO's strategy seems very reasonable.

SCO's Windows-Friendly Strategy

Although there are many products on the market that allow Windows PCs to interoperate with Unix, no Unix vendor has really stepped up to the plate to proclaim Windows interoperability as a central strategy. In fact, in the face of Windows NT, many Unix vendors have fled in the opposite direction. SCO's bold move may position it as the "acceptable" approach to Windows/Unix integration.

There are several elements to the SCO strategy:

- Running Microsoft Windows application binaries, which SCO accomplishes with products like SCO Merge and SunSelect's Wabi.
- Giving Unix applications an MS Windows appearance for MS Windows users. This is the role of its new product, Win-tif.
- Connectivity of MS Windows to Unix, including clients as well as Windows NT servers. Currently, SCO's LAN Manager for Unix fills this requirement, and it will be supplemented with other products in the future.
- Application interoperability for integrating the use of Unix applications running on large servers and hosts with personal productivity applications. This includes data access and distributed application infrastructure support.

The Windows NT Alternative

The SCO initiative is not limited to Windows. It can encompass Windows NT as well. There are many areas, both in the Windows NT client as well as the Windows NT Advanced Server, where Microsoft's product is incomplete from an interoperability perspective. Although TCP/IP is supported and, in fact, strategic, key TCP/IP applications are missing. For example, Windows NT lacks a Domain Name Server, NFS server, SMTP support, and NIS+ support. Although SNMP is supported, there is no SNMP management application. And although telnet out is supported, remote users cannot telnet into a Windows NT server. All of these present SCO with opportunities to provide server configurations that complement Microsoft's Windows NT strategy. Microsoft is likely to address some of these shortcomings, but SCO is much more likely to deliver solutions that make Unix customers feel comfortable.

The UnixWare Alternative

Users could go the UnixWare route, but UnixWare is new, has few applications, and lacks the multiprocessor support for scalability that SCO can provide. In addition, Novell Corporation's (Provo, Utah) UnixWare strategy has many elements, such as AppWare, that offer alternatives to Microsoft's strategic direction. SCO is saying, "Users should be able to use any Microsoft technologies they want and still have the benefits of Unix servers, Unix networking, and Unix standards."

Putting Windows Dressing on Motif Applications

SCO's subsidiary, IXI Limited (Cambridge, England), has announced Win-tif, a runtime library set that can replace the Motif library and window manager, which gives applications written for Motif an MS Windows 3.1 look and feel. Win-tif is intended to be used in conjunction with any X Window server for MS Windows. It gives MS Windows users the ability to have Motif applications run on a Unix host and display in an X Window on their MS Windows PC with an MS Windows interface instead of a Motif interface.

Since the Motif application is unmodified, the users' ability to use MS Windows features like cut-and-paste, DDE, OLE, and drag-and-drop is limited by the analogous support for those capabilities provided by the X Window System and Motif. To the extent that the COSE Common Desktop Environment (CDE) will support those capabilities, future Motif applications can have them, and they can be supported in the MS Windows interface.

For some customers, Win-tif could supplement SunSoft Company's (Mountain View, California) Wabi by allowing Unix servers to run Windows and Motif applications and give users of X terminals a consistent Windows look and feel.

Appeal to Customers More than ISVs

Although independent software vendors (ISVs) will find Win-tif a potential way to avoid doing native Windows ports of their Motif applications, the real appeal will be to customers who have internally-developed Motif applications that they want Windows users to be able to access without having to train and educate those users in Motif. The application can be developed once and then deployed on any Motif platform or on an MS Windows PC; it will assume the native look and feel on each.

Motif and CDE Gain Status

With this approach, both Motif and CDE gain an interesting status. They become standard application programming interfaces (APIs) to which a developer can write and have the application "portable" across Motif and Windows. With the analogous Mac-tif technology from Apple Computer Incorporated's (Cupertino, California) AppleSoft unit, which should be available in 1994, portability to the Macintosh is achieved as well. Instead of writing to a proprietary API, such as that provided by Visix's (Reston, Virginia) Galaxy, XVT's (Boulder, Colorado) XVT, or Neuron Data's (Palo Alto, California) Open Interface, the developer need only develop a CDE/Motif application, and, at runtime, that application is linked to the appropriate toolkit, Motif, Win-tif, or Mac-tif.

Fully implementing this strategy depends on SCO making Win-tif available on other vendors' platforms. Since it does not have a unique API, Win-tif does not affect any existing standards. It just has to be installed on an application server so an administrator can configure an application to use it at runtime. Original equipment manufacturers (OEMs) might want to bundle Win-tif as a part of every system, or they may just make it an option. In any case, widespread adoption of Win-tif could be a windfall for SCO.

This approach is not perfect, however. Performance is less than what would be achieved natively. Many Windows features are not available, such as OLE and access to certain hardware devices.

Is This Enough for SCO to Compete?

SCO has long been the established leader of providing Unix on Intel PC platforms. Its position has been subject to increased challenges, first from SunSoft, and then from Novell. SCO has a distinct advantage over both. Since it is not a hardware manufacturer, the company does not have to balance its software interests against its hardware interests the way SunSoft must. Also, since SCO does not have a proprietary networking scheme to advance, as Novell does, it need not push that agenda. In some ways, SCO remains the purest example of an open systems software vendor in the industry. The long list of technology license trademark credits that scrolls across an SCO Open Server start-up screen attests to that position.

While SCO and Microsoft compete for system software business, Microsoft is also a part owner of SCO, with about 15 percent of the company's stock. Although Microsoft doesn't play a direct role in setting strategic direction, SCO is aware that its own strategy must take into account Microsoft's directions. Laying out a complementary strategy allows SCO to continue to support its Unix-only customers and still gain business in the growing Unix and MS Windows integration market. This strategy makes a lot more sense than trying to challenge Microsoft head-on, even if SCO were a much larger company. — *M. Goulde*

FOCUS: VENDORS

The New NCR: Calling Home

When AT&T acquired NCR a few years ago, the big fear in the industry was that AT&T would interfere with NCR's new Open Cooperative Computing strategy, which it had unveiled the previous year. A few years passed, AT&T pretty much kept hands off, and NCR's new strategy and products did not take off as many had predicted. There were several reasons for this, including products delivered behind schedule; software that was too monolithic, hard to install, and hard to configure; some bad choices of priorities; and weak marketing.

AT&T had already suffered through one computer strategy debacle and was not about to suffer through another. NCR's management was changed, and, at a recent consultant's conference in Dayton, Ohio, NCR unveiled a new direction that has a much tighter fit with AT&T's direction than it had previously. The goal for this new strategy is to grow NCR into a $10 billion business by 1996.

The Network Is THE NETWORK

AT&T's strategic asset is its worldwide telecommunications network, which provides connectivity, services, and intelligence around the globe. It also provides massive revenues to AT&T. The more the network is used by customers, the more revenue is achieved. NCR's new mission is to find ways for its customers to find value in the AT&T network.

AT&T's goal is to extend today's networks in areas such as wireless and cellular, make them work better, make them more accessible, and make them more useful to a wider variety of applications. In order for the services provided by the network to be accessed, new types of devices will often be required. These might be new types of intelligent telephones, new hand-held devices, and computers that are oriented toward the network. NCR will have a key role to play here.

Changing the Organization

AT&T had 20 business units in 1988. It now is organized with hundreds of profit centers. There are many cross-group initiatives that are trying leverage the various competencies that exist within AT&T. Over the past five years, the company has examined its structure and organized itself around five business groups, each headed by its own CEO. NCR, headed by Jerre Stead, is one of the five. The others are Global Operations, Network Systems, Communication Services, and Multimedia Products and Services.

The company has started six cross-group initiatives designed to leverage core competencies. Its challenge has been to bring its business units and profit centers into alignment with these initiatives and embed these initiatives in all of the business units. The initiatives are:

- Voice/audio processing
- Messaging
- Wireless communications
- Visual/multimedia communications
- Data communications
- Scalable computing

Obviously, NCR is in a position to play a key role in most, if not all, of these initiatives. In order to do so, the company has had to change policies, procedures, and rules so that employees can make decisions more effectively. Two tasks high on Stead's agenda are: to have NCR do a better job of listening and to adopt a much stronger marketing focus. The new CEO is organizing the entire company into customer-focused teams.

Internal reorganization efforts have already begun. Large pieces of NCR's diverse software offerings, including Process IT, Cooperation, Top End, and StarPRO Messaging, have been put under the leadership of the vice president of a new Software Products division. That person's mandate is to rationalize and clarify NCR's direction, make adjustments in products and strategy, and enter new markets. NCR will be uncoupling its software strategy from its platform strategy, and many NCR software products will find their way to other vendors' platforms.

Bringing Computing and Communications Together—Again

NCR's objective is to be the world's best at bringing computing and communication solutions together to provide people with easy access to information and to each other anytime, anywhere. There is nothing particularly unique in this objective. However, NCR now seems to be positioning itself to capitalize on AT&T's telecommunications infrastructure as well as other assets, not the least of which is Bell Labs.

In addition to mobile computing and wireless communications, synergy between AT&T and NCR will be sought in areas such as materials research and stress resistance to build more durable and reliable products; video to capitalize on AT&T's video research and experience with standards; and, of course, data networking, an area where AT&T has R&D efforts underway in all emerging communication technologies. NCR's continuing strategy of offering a broad family of systems, from notebook computers to massively parallel database engines, provides AT&T with the platforms on which to deliver advanced applications.

In the past, NCR has underutilized the capability of Bell Labs, arguably the best software development resource in the world. That is ending, and NCR has already begun introducing many development tools that were being used internally at Bell Labs as NCR products. It has only begun to scratch the surface of what is available and what might be packaged for the market.

Get It, Move It, Use It

Much of the effort at uniting NCR and AT&T technologies is focused on the concept that the way we interact with information can be described by three processes: getting it, moving it, and using it.

"Getting it" refers to information capture. In addition to the traditional keyboard, getting includes a range of point of sale (POS) devices, including automated teller machines, smart cards, scanners, various item processing devices, hand-held devices, and, of course, the AT&T telephone. All are devices for getting information and introducing it into the system. NCR's traditional POS presence in retail and banking gives it strength here.

The next stage in processing information is moving it. Here, of course, the information collected is moved through AT&T networks. The breadth and depth of communication services—including both local and wide area networking and hardware, software, and switching technologies—are important in this area.

Finally, using information covers all application types, ranging from decision support to OLTP. Applications, tools, databases, systems and network management, and computing platforms are all key enablers for allowing information to be used both in business and in the home. NCR believes that it has the products to accomplish this mission; now it has to convince the market.

Can't Do It Alone

But NCR needs help. Delivering solutions that get, move, and use information requires services in addition to products. Services including everything from high-level business analysis, process re-engineering, and IT architecture consulting to system and application design, integration, and installation are all required. NCR does not have the expertise to do this, and neither does AT&T. The company recognizes that it must find partners in the consulting and systems integration community if it is to succeed. It is actively seeking partners in these areas throughout the world and can be expected to announce a series of agreements over the next year.

NCR has tended to go its own way in the past. Although Cooperation was an integration of technologies from dozens of suppliers, NCR has done its own implementations in areas as diverse as Object Request Brokers and transaction monitors. Part of the new direction of industry partnering will be creating alliances that can promulgate NCR technology into the industry and also bring needed technology into NCR's product line. The AT&T connection is important in this area, but so are partnerships with others in the industry.

Targeted Markets

One of NCR's weaknesses has been in marketing. In part, it lacked a clear focus of what its target markets were. The company has now identified those markets and organized its sales function around them. It has not gone as far as Digital Equipment Corporation and organized the company around vertical industries, however. The focuses for NCR will be:

- Its traditional retail and banking customers
- The communication industries, including the RBOCs and PTTs, an obvious concentration for an AT&T company
- Consumer goods manufacturing, leveraging its strengths in retail and the trend for intercompany information exchange
- Transportation, because of the strength of AT&T and NCR's Large Computer Products Division (formerly Teradata) in that industry
- Public sector, because of its open systems bent and the need for cost-effective solutions

In addition, NCR will apply additional resources in emerging markets, including health care, petrochemicals, and utilities.

Differentiating in a Competitive Market

NCR was an early convert to open systems and to the use of commodity technology to deliver value-added solutions. However, it lost momentum between 1990 and 1992, not only because of the merger with AT&T, but also because it failed on execution of its strategy. The key to NCR's success with its second-generation strategy will be its ability to execute. None of the other computer companies has the resources available to it that NCR has in the form of AT&T. The initial reluctance to leverage those resources was understandable. AT&T had failed miserably in its attempt to enter the computer industry through its now defunct Computer Systems Division, and NCR didn't want to be tainted by that division's failure.

The new strategy represents a second life. There is synergy possible between NCR and AT&T, and Jerre Stead does seem to be the kind of leader who can make that synergy pay off. Although NCR's hardware and software offerings have lost the panache they had when they were first announced, the company is in a position to succeed in the approaching era of information superhighways and the combination of computing and communications. *— M. Goulde*

1994 International Conference on
Applications of Databases
ADB-94

June 21-23, 1994
Vadstena Monastery Hotel, Sweden

CALL FOR PAPERS

SPONSORED BY:
The Swedish Board for Technical Development (NUTEK), Hewlett-Packard, Softlab AB, Ellemtel AB, GMD. In cooperation (requested) with ACM, AFCET, AICA, GI, IEEE, IFIP, INRIA, VLDB

SCOPE

Successful database research and development has made many general database systems available. The industrial offerings range from traditional large mainframe systems, through workstations & PC servers, to massproduced $100 personal systems. Users, application designers, and the software industry strive to get the most from this technology. Database researchers, and developers try hard to further satisfy application needs. The conference aims at developing synergy between these communities. It is intended as the forum to explore innovative applications of databases, and innovative database services for specific applications.

PAPER SUBMISSION

We solicit research, industrial and user's papers, panel proposals, and tutorial proposals. Papers should be original, not exceeding 6.000 words, (25 double spaced pages), and with abstract. Accepted papers will be published by Springer Verlag. All submissions will be notified of acceptance or rejection by April 1, 1994. Camera ready copy will be due for the conference proceedings by April 14, 1994.

Six copies of the submission, including e-address or fax number, should be sent by February 14, 1994 to:
ADB-94 Secretariat, email: adb94@ida.liu.se, Anne Eskilsson, Dept of Computer and Information Science, Linköping University, S-581 83 LINKÖPING, Sweden.

LOCATION

The town of Vadstena is located about 300 km. south of Stockholm, bordering lake Vättern, the 2nd largest lake in Sweden. Vadstena is a medieval town with cobbled streets, an ancient abbey and castle. It is near the town of Linköping, center for high-tech industry, and home of one of the largest universities in Sweden. The conference site is the monastery originally built as a royal palace in 1258, pre-dating Stockholm. It is the oldest secular building in Sweden. The conference week includes the longest day of the year, when there is light all night in Vadstena and midnight sun in northern Sweden. June 24 is also the national 'Midsummer' holiday in Sweden, when many ethnic celebrations take place. June/July is about the best time for vacations in Scandinavia.

General Topics of interest will include but are not limited to:

APPLICATIONS
- General Enterprise Management
- Financial and Business Decision Support, Reporting, Spreadsheets
- Interpersonal Communication: Datebooks, Contacts, Meetings, Email
- Electronic Documents, Publishing, and Libraries
- Application Development: CUI, GUI, CASE, Development Libraries, Interfaces, and Utilities
- Telecommunications: Switching & Billing, On-Line Services, National Information Infrastructure
- Personal Assistants for Mobile Users
- Manufacturing: CAD, CAM, Signal Processing, Robotics
- Consumer Services: Insurances, Banking, Real-Estate, Travel, Sales, Marketing & Distribution
- Mathematics: Formal Calculus, Numerical Analysis, Statistics, Computational Geometry
- Applied Sciences & Engineering: Geography, Chemistry, Aeronautics, Biotechnologies
- Healthcare
- Law & HumanitiesTraining & Education
- Leisure & Entertainment

SERVICES
- Interoperability of databases, and with legacy applications
- Visual Database Languages
- Databases on multicomputers, parallel computers, and data highways
- Mobile databases
- Agents and Mediators
- New storage media and technologies
- Text, Image, Sound, and Video in databases
- Object Management and Libraries
- Data migration to databases
- Database performance, benchmarking, and tuning
- Large Scale Information Retrieval and Mining
- Time, Event Management, and Monitoring
- Customization of database manipulations
- Use of standards: SQL, ODBC, OLE, DDE, WOSA, DRDA, CORBA, DCE....
- DBMSs on new OSs, FMSs, and Transaction Managers: Windows NT, OS2, Chorus, Mach, AFS, Encina, Tuxedo,.....
- Security

IMPORTANT DATES
Submissions: February 14, 1994,
Notif. of acceptance: April 1, 1994
Conference: June 21-23, 1994

Patricia Seybold's Computer Industry Reports Order Form

Please start my subscription to:

		U.S.A.	Canada	Foreign
☐ *Workgroup Computing Report*	12 issues per year	$385	$397	$409
☐ *Open Information Systems*	12 issues per year	$495	$507	$519
☐ *Distributed Computing Monitor*	12 issues per year	$495	$507	$519

Please send me a sample of: ☐ *Distributed Computing Monitor* ☐ *Open Information Systems* ☐ *Workgroup Computing Report* ☐ *Paradigm Shift—Case Studies in Distributed Computing*

Please send me information on: ☐ Consulting ☐ In-Depth Research Reports ☐ Conferences

Total $_____ ☐ **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: _____

Fax No.: _____ 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 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. **IO-1093**

Send to: Patricia Seybold Group: 148 State Street, 7th Floor, Boston MA 02109; FAX: (617) 742-1028; Phone: (617) 742-5200; MCI Mail: PSOCG

1992 and 1993 Feature Reports from Patricia Seybold Group
To Order these Back Issues, Fax (617) 742-1028, or Call (617) 742-5200.

Office Computing Report

Volume 16 issues—$50

- ☐ 1/93 Microsoft Access—"Cirrus" Database Project Gets down to Earth

Workgroup Computing Report

- ☐ 2/93 WordPerfect Information Systems Environment—WordPerfect Reveals Its Blueprint for Workgroup Support
- ☐ 3/93 Lotus Notes Release 3—Extending the Notes Paradigm
- ☐ 4/93 Can Windows NT Meet the Challenge?—Microsoft's Next Generation Operating System Stirs the Industry
- ☐ 5/93 Action Technologies' Workflow Products—Coordinating the Activities of People as They Work Together
- ☐ 6/93 Implementing Groupware—Groupware May Be Hazardous to Your Organization's Status Quo!
- ☐ 7/93 Issues in Remote Computing—Notebook PC Boom Raises Question: How to Handle Detachable Computing
- ☐ 8/93 Scheduling-Based Workflow—IET's W-6 Attacks the Complex Area of Time-Dependent Processes
- ☐ 9/93 Portfolio's Office IQ and ProdeaSynergy—Innovative New Products for Workgroups
- ☐ 10/93 Rules-Based Messaging—Using Electronic Mail as the Vehicle for Customized Workflow Applications

Back Issue Total $_____

Open Information Systems

Volume 7 issues—$50

- ☐ 9/92 Oracle's Version 7—Can It Leapfrog the Competition?
- ☐ 10/92 Galaxy from Visix—Application Portability Breakthrough?
- ☐ 11/92 European Open Systems Architectures—Europe's Vendors Strike out for Open Distributed Systems
- ☐ 12/92 The Unix Data Center—Fact or Fiction?

Volume 8 issues—$50

- ☐ 1/93 X/Open in the 1990s—Making Open Systems Safe for Users
- ☐ 2/93 Highly Available Open Systems—Expanding Today's Definition
- ☐ 3/93 Sybase System 10—Can It Manage Enterprise Data?
- ☐ 4/93 Unisys ASD Framework—Meeting the Challenges of Software Development in the 1990s
- ☐ 5/93 Unix and PC Interoperability—Toward the Utility Era of Computing
- ☐ 6/93 Open Electronic Mail—Interoperability through Standards
- ☐ 7/93 SAP's R/3 Software Suite—Architecturing Open, Integrated Software for Distributed Enterprises
- ☐ 8/93 Tomorrow's Microkernel-Based Unix Operating Systems
- ☐ 9/93 Apple Computer and Open Systems—The Other Desktop Standard
- ☐ 10/93 Progress Software's Version 7—Solving Client/Server Migration Issues

Back Issue Total $_____

Distributed Computing Monitor

Volume 8 issues—$50

- ☐ 1/93 Component Software—A Market Perspective on the Coming Revolution on Solutions Development
- ☐ 2/93 Switched Internets—The Coming Gigabit Revolution in Enterprise Networking
- ☐ 3/93 IBM's System Object Model—Cornerstone of an Open Distributed Object Computing Environment
- ☐ 4/93 Encapsulating Databases—Practical Uses of Object Technology to Inprove the Value of Relational Data
- ☐ 5/93 OMG's CORBA 2.0—Industrial Grade Standard for Distributed Object Computing?
- ☐ 6/93 Enterprise System Management—The Quest for Industrial Strength Management for Distributed Systems
- ☐ 7/93 Detachable Computing—Vendors and Users Scramble to Support Occasionally Connected Computers
- ☐ 8/93 Modeling Network Behavior Using Objects—Horizon Strategies' AROs Increase Simplicity and Flexibility of Distributed Applications
- ☐ 9/93 Migrating to ATM—Four Scenarios to Take Users from Today's Networks to Switched Gigabit "Fabrics"
- ☐ 10/93 The Distributed Enterprise—The Logical Next Step in Distributed Object Computing Is Shared, Reusable Business Models, Not Just Objects

Back Issue Total $_____