Architectures for Large-Scale Reuse

Roger P. Beck
Satish R. Desai
Doris R. Ryan
Ronald W. Tower
Dennis Q. Vroom
Linda Mayer Wood

Large-scale reuse of software architectures and components is an important element for improving software productivity and reducing the time needed to design, develop, and deploy software systems. For maximum benefit, the application of reuse technologies must be coupled with effective process changes. In this paper, we focus on AT&T's approach to software reuse, particularly the use of the AT&T BaseWorX™ applications platform to achieve reuse. We also discuss the process changes associated with software reuse, and the results that have been achieved from applying reuse to a wide range of software products.

Reuse Approach

If AT&T is to remain competitive, it is clear that we need to reduce the cost of building and maintaining software systems and reduce the time to market.[1] In response to market pressures to deliver software systems to customers more quickly and at lower cost, AT&T has implemented new processes and technology to improve software production. A key aspect of this strategy is large-scale reuse—specifically, the reuse of proven software architectures and components that speed the development of software systems. This paper discusses our approach to software reuse and to the process changes associated with software reuse.

Reuse has received much attention in the industry.[2] The goal of our approach to software reuse is to change software-system production from a *labor-intensive* process into a *capital-intensive* process. The fundamental concept is to build and accumulate software components and, then, reuse them in new software systems.

Our approach to software reuse has analogies in the electronics industry. Over the past twenty years, the process of building electronic hardware systems has largely moved away from using discrete hardware components (e.g., resistors, capacitors) to building large-scale, integrated, hardware components and reusing them in new hardware systems. We see the process of building software systems changing from one that uses discrete software components (such as library routines) to one that builds large-scale, integrated, software platforms and reuses the platforms in new software systems.

To achieve the maximum benefit from software reuse, we believe that three elements are essential:

- *Architecture frameworks* for reuse must be developed. Reuse at the architectural level, rather than at the subroutine level, is required to achieve large-scale gains in productivity.
- The high-quality, flexible *software components* needed to realize those architectures must be provided. Components must be developed, integrated, collected, tested, and distributed to the development community.
- *Support services* must be provided to facilitate reuse of the frameworks and components. These services include consultation, documentation, and training.

Our experience has shown that, to establish a successful reuse program, all three elements are interdependent and indispensable. In addition, we have created an *integrated platform* to maximize reuse from these elements. This paper describes our efforts in each of these three areas. (Panel 1 defines acronyms and terms used in this paper.)

Objectives for Reuse

Work on reuse and the AT&T BaseWorX applications platform (formerly known

Panel 1. Abbreviations, Acronyms, and Terms

ANSI — American National Standards Institute

API — application program interface; software the application program can reference to access services

ASCII — American Standard Code for Information Interchange

ASN.1 — abstract syntax notation 1

CCITT — International Telegraph and Telephone Consultative Committee (Geneva, Switzerland)

CCP — common communications platform

directory services — identify the current address of the receiving process

CMIP — common management information protocol (an ISO/OSI network-management protocol)

CPU — central-processing unit

DBMS — database management system

DCS — AT&T Display Construction Set, a user-interface management system

Ethernet — a local-area network that interconnects personal computers via coaxial cable. It was jointly developed by Xerox Corporation, Digital Equipment Corporation, and Intel Corporation.

IEEE — Institute of Electrical and Electronics Engineers

ISO — International Organization for Standardization

GDMO — Guidelines for the Definition of Managed Objects; part of the CCITT's Telecommunications Management Network standards

managed object — the object-oriented representation of resources in the OSI-system-management model

MNLS — multiple national language support

MOG — managed-object generator

OA&M — operations, administration, and maintenance

OSI — Open Systems Interconnection, a set of ISO/CCITT standards; defines a computer-network architecture that divides network functions into seven layers

pipe — a common UNIX system interprocess-communication mechanism

POSIX — IEEE Standard 1003.1 for portable operating systems

RBC — reusable backplane connector; software that connects a user-interface component to the communications backplane

ROAM — reusable operations, administration, and maintenance

SQL — structured query language (pronounced like *see kwul*), an ANSI-standard language for database management

TCP/IP — transmission control protocol/internet protocol; an internetworking standard for OSI levels 3 and 4

time to market — the interval from identification of a customer need to delivery of a product

TMN — the CCITT's telecommunications management network standards

VCS — virtual circuit switch

X.25 — the CCITT standard for a packet-switching protocol at the link and packet layers

as RAPID/NM) began in 1987,[3] and has grown considerably in scope and impact within AT&T.[4–6] To provide a framework for large-scale reuse, the BaseWorX applications platform delivers an architecture, a set of integrated software components, and numerous customer-support services. This integration, the large number and variety of hardware platforms on which the software operates, and the accompanying services set the BaseWorX platform apart from other development platforms or operating environments.

We established and achieved six objectives with our platform approach to software reuse:

- We designed the platform to be conducive to prototyping and rapid development. Thus, developers can build a prototype of their application to get early customer feedback or quickly develop and deploy the application.
- We based the platform on standard protocols, such as SQL for database access[7] and standard CCITT interfaces[8] for network communications. (CCITT is the International Telegraph and Telephone Consultative Committee, which is based in Geneva, Switzerland. SQL stands for *structured query language* and is an ANSI-standard language for database management. ANSI is the American National Standards Institute, based in Washington, D.C.)
- Whenever possible, we used commercially available components such as commercial database management systems (DBMSs).

- We designed the platform to allow applications developers to select the components used in their systems, such as a DBMS, and to permit customers to program or customize their systems to meet their ever-changing business needs.
- We provided extensive reusable operations, administration, and maintenance (ROAM) capabilities to provide a uniform interface and method for managing and operating the deployed systems.
- Finally, we designed the platform to run on a wide range of computer systems that are based on the UNIX® operating system and comply with the IEEE POSIX standards.[9] (UNIX is a registered trademark of UNIX System Laboratories, Inc.)

As a result of meeting these objectives, the platform has proved to be flexible, extensible, and applicable to a wide variety of software applications. However, our primary domain has been network-management systems and operations systems for the telecommunications market. Operations systems are large software systems that are used to operate and administer telecommunications networks. These systems provide a wide range of functionality for customers—from billing and provisioning, to switch maintenance and circuit testing—but have many similar architectural elements. This similarity is what created a strong potential for large-scale reuse.

Frameworks for Reuse

We used informal domain-analysis techniques[10] to study several classes of applications to determine their architectural similarities and their common software elements. Through this analysis and abstraction, an architecture framework and a set of components were identified to serve as the base for the platform.

For example, all the applications studied had these elements in common: a user interface; networking; database management system; interprocess communications; and operations, administration, and maintenance (OA&M). Therefore, the BaseWorX applications platform provides many of these "horizontal," or system-level, components so that applications developers can concentrate on creating the customer's application and the application-specific pieces needed to complete it. The base of components in the platform continues to grow as more needs are identified through application-domain analysis and as components are developed or contributed by other projects.

For each application class, an architectural template was constructed that identifies the components an applications developer needs from the BaseWorX platform and the application pieces he or she must develop.

Two architectural configurations—a client/server paradigm and a manager/agent paradigm—are provided with the BaseWorX platform and are supported by its components. This enables the platform's users to incorporate the method best suited to their application. With these paradigms, the platform can be used to develop on-line transaction processing systems and event-driven network-management systems, as well as any hybrid of the client/server and manager/agent architectures.

Client/server Configuration. In the client/server paradigm, the client (i.e., a process) requests a particular service that is supplied by one or more servers. The clients need not know the physical locations or operating characteristics of the servers, nor does a server need to know this about its clients. Thus, applications can be moved transparently onto different machines.

Within the client/server paradigm, communications can be modeled in different ways, i.e., connectionless and connection-oriented.

Communications can be modeled as a series of unrelated requests, where the load can be balanced among multiple servers. Each server handles requests from many clients. Updates are often regarded as atomic transactions, where all or no parts of the transaction are complete. (*Atomic* means indivisible; either the transaction occurs fully or not at all.) This is sometimes referred to as transaction processing, and is typically implemented using *connectionless* communications.

In other situations, a client needs to carry on a dialog with a specific server, because it is necessary to maintain state information between interactions. (*State information* refers to information about the status of the process, client, or server.) This is called peer-to-peer communications, which uses a *connection-oriented* model.

Examples of both models are common in operations systems. For example, a repair-service bureau—where many customers call the bureau's customer-service representatives, who perform independent operations on a central database—is often modeled as a transaction-processing system. On the other hand, a network-management system might have direct connections to monitor the operations of network

elements, such as switches, and would provide maintenance information or generate alarms based on the operation of the network elements. Such systems would typically be implemented using peer-to-peer communications.

Manager/agent Configuration. The manager/agent model is a special type of client/server paradigm that is oriented toward OSI system management. (OSI stands for *Open Systems Interconnection*, a set of standards issued jointly by the ISO and CCITT. It defines a computer-network architecture that divides network functions into seven protocol layers. ISO is the International Organization for Standardization, which is based in Geneva, Switzerland.)

In the manager/agent model, resources are modeled as objects. One process (the agent) provides a managed-object view of managed resources to another process (the manager), which sends commands to and receives event notifications from the agent. (A *managed object* is the object-oriented representation of resources in the OSI system-management model.) The agent hides proprietary details of the managed objects from the management system.

The OSI system-management model is a special implementation of the manager/agent model and is needed to support the CCITT's telecommunications management network (TMN) standards.[11] To address this need, the BaseWorX platform provides high-level interfaces to CMIP and tools to simplify writing code for managed objects. (CMIP is the common management information protocol, an OSI network-management protocol.)

Reusable Software Components

By using the results of our domain analysis and the architecture frameworks that had been identified, we developed reusable software components as part of the BaseWorX platform.

The platform is based on the concept of a *software backplane* that supports the core communications and OA&M for the platform and developed applications. The backplane provides the main set of services needed for applications to communicate and to share common resources. To provide the management needed for the application, the backplane also monitors and controls the application and the rest of the platform.

A variety of other services that applications need surround this basic infrastructure. These services include user interfaces, database management systems, and platform-development tools.

To build an application on the platform, applications developers first select the appropriate paradigm and architecture framework. They then choose the appropriate reusable components from the backplane and from the reusable database management systems and user interfaces to meet system requirements. Next, the applications developers build specific clients and servers (or managers and agents) that contain the target functionality of their application (e.g., billing, provisioning, trouble ticketing), reusing the platform as an effective base.

Consistent with our charter of reuse, we try to reuse proven software components from other sources wherever possible, rather than embark on internal development efforts. Several of the software components used in the platform are provided by industry vendors. Some of the components have been adapted from other projects within AT&T. The rest were developed by the BaseWorX platform product team because no existing software met our needs.

Components from these different sources are selected based on the component's performance, functionality, adherence to industry standards, and suitability to the BaseWorX platform architecture. Thus, projects that build applications on the BaseWorX platform have the benefit of reusing a proven software base of high quality and known performance characteristics.

Figure 1 shows the architecture of the AT&T BaseWorX platform. The sections that follow describe the major classes of reusable components included in the platform.

Communications and OA&M Backplane. The software backplane, which consists of the *communications* and *OA&M subsystems*, provides the infrastructure on which applications and the rest of the platform run. The backplane provides the types of communications services that applications need. It also manages applications, allowing them to reside on one or more machines or to share the same machine, to maximize the effectiveness of computer resources.

The backplane works over a variety of communications mechanisms and supports communications and administration of distributed applications. Application processes do not have to know the physical location of either their communications partners or the software that is managing the processes. Instead, *directory services* are used to identify the current address of the receiving process. Applications can be redistributed across

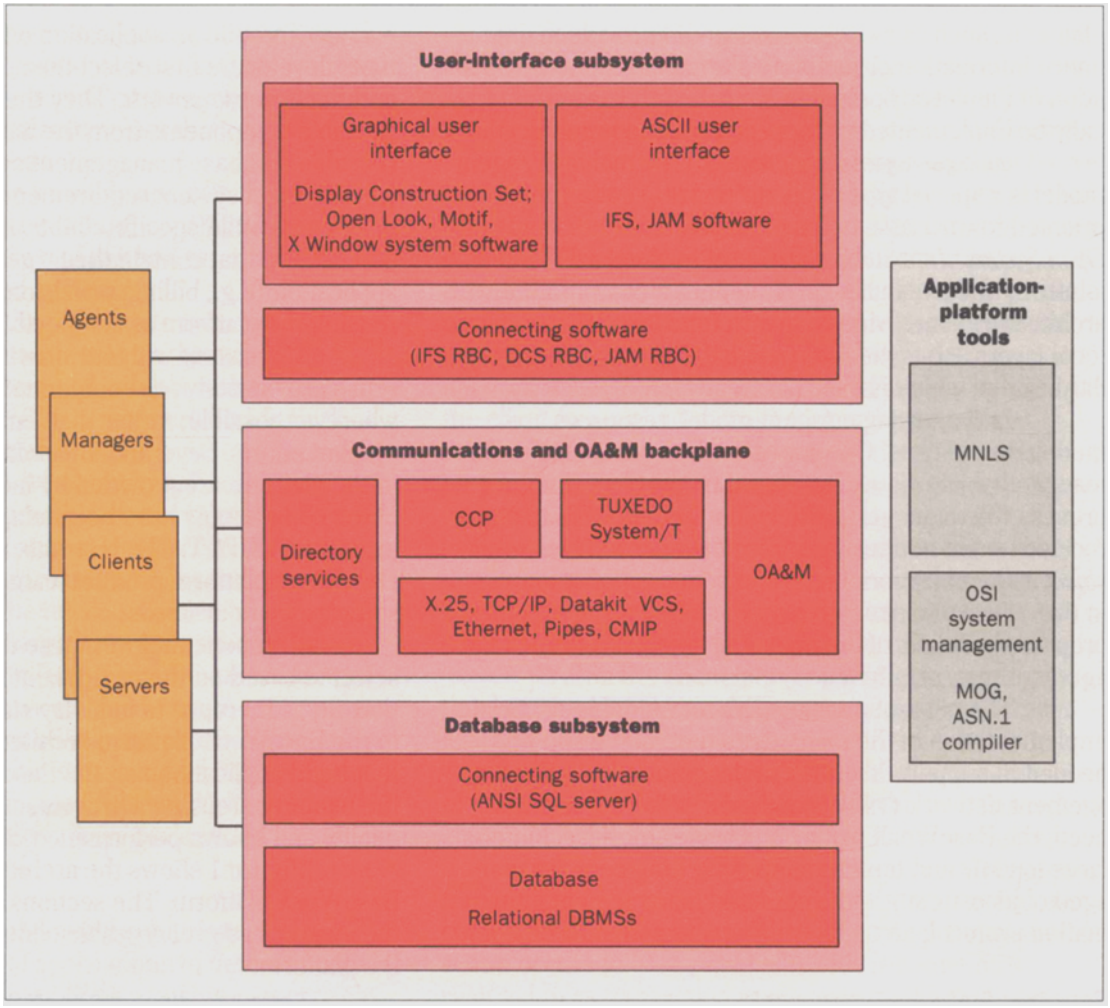

Figure 1. The AT&T BaseWorX applications platform delivers an architecture, a set of integrated software components, and numerous customer-support services that applications developers can reuse. The platform is based on standard protocols and is designed to run on a wide range of UNIX system machines that comply with POSIX standards.

machines by changing the configuration information.

The software backplane includes:

- The BaseWorX common communications platform (CCP) component and TUXEDO® System /T transaction processing software[12] to provide communications. Each provides specific classes of communications services. (TUXEDO is a registered trademark of UNIX System Laboratories, Inc.)
- ROAM to provide operations, administration, and maintenance. ROAM manages the platform and applications and can control both CCP and TUXEDO System /T processes.

Communications subsystem. The BaseWorX platform supports a wide spectrum of reusable communications services to satisfy the needs of many operations-systems and network-management applications. The platform supports communication within a processor, as well as networking between processors and between different systems. Both the client/server (i.e., peer-to-peer and transaction processing) and the manager/agent (i.e., OSI system management) paradigms are implemented.

The communications subsystem is designed to shield applications from the underlying subnetworks [e.g., X.25, TCP/IP, AT&T Datakit® virtual-circuit switch (VCS), pipes, Ethernet, or CMIP). This transparency makes system design and development easier, facilitates reuse, and enables the application to work over a variety

of networks. The flexibility it provides also allows the communications mechanism to be changed in the future with minimal changes to the application software.

The CCP component provides a peer-to-peer message communications service, using a uniform applications-programming interface (API). This interface is based on UNIX-system-style primitives (i.e., open or close, read or write) across different environments.

The CCP translates an application's request for service into a request to the underlying networking mechanism, and maps or generates return codes as needed. Applications establish bidirectional paths called *associations* with remote or local applications, which are then used to send messages. The remote application is identified by a symbolic name that is mapped internally to a network address when the association is established. An application may have any reasonable number of active associations, as determined by the system configurations and availability of resources. An application can read data from a specific association or from all active associations. All this is done in a consistent way across the various subnetworks.

Transaction-processing platform services support an enhanced client/server model, distributed transaction processing, and a two-phase commit protocol. (The commit protocol ensures that transactions are atomic.) These services follow the X/OPEN® model for transaction processing and associated applications-programming interfaces. (X/OPEN is a registered trademark of X/OPEN Ltd.)

TUXEDO System /T uses a name server to map the server locations and transparently sends requests to servers on the same machine or on different machines. The system automatically balances the requests load among servers that offer the same service. Transactions can span several servers, possibly on different machines. TUXEDO System /T will handle the two-phase commit protocol needed to synchronize the different participants in the transaction and ensure that the entire request or none of the request is completed.

In a typical operations system, clients usually access devices, such as terminals; and, then, interact with TUXEDO System /T's applications-programming interface to talk to servers. Server applications also typically interact with a resource manager, such as a DBMS. The transaction manager interacts with resource managers using the X/OPEN DBMS interface protocol.

OA&M subsystem. As applications become more complex, they need to be managed just as much as networks and network elements do. However, system management of applications is often overlooked.

To address this need, the BaseWorX platform provides a comprehensive approach to managing applications via the ROAM subsystem. The reusable OA&M software, an essential part of the backplane, was built to provide maximum benefit to the platform and applications and to encourage reuse across projects.

The BaseWorX platform provides a full set of OA&M functionality for use in centrally administering the platform and the applications being developed. In addition, the OA&M subsystem can lay the foundation for an application to provide OA&M services for other applications within the network, such as transmission and switching equipment. The platform's OA&M capabilities provide a common look and feel for the OA&M interface of applications built on the BaseWorX platform.

Platform OA&M follows the OSI network-management model, providing support for fault management, configuration management, and security management. The OA&M module uses the networking facilities of the communications subsystem to support a fully *distributed* operations model. Therefore, the applications and processes administered may reside on the same processor or on different processors.

The specific functionality provided by the ROAM subsystem includes:

- *Installation and packaging of applications.*
- *Distributed execution management* that allows groups of processes to be started and stopped on one or many machines and their status to be checked.
- *Distributed fault management* that allows fault messages to be collected from local or remote sources. It also supports filtering, logging, alerting, and browsing of these messages, as commonly used in operations-systems software.
- *Log management* that collects and centrally logs messages from vendor log files for third-party software that is used with the platform.
- *Backup and restore* capabilities for applications and databases.
- *User interfaces* for entering and maintaining configuration data.

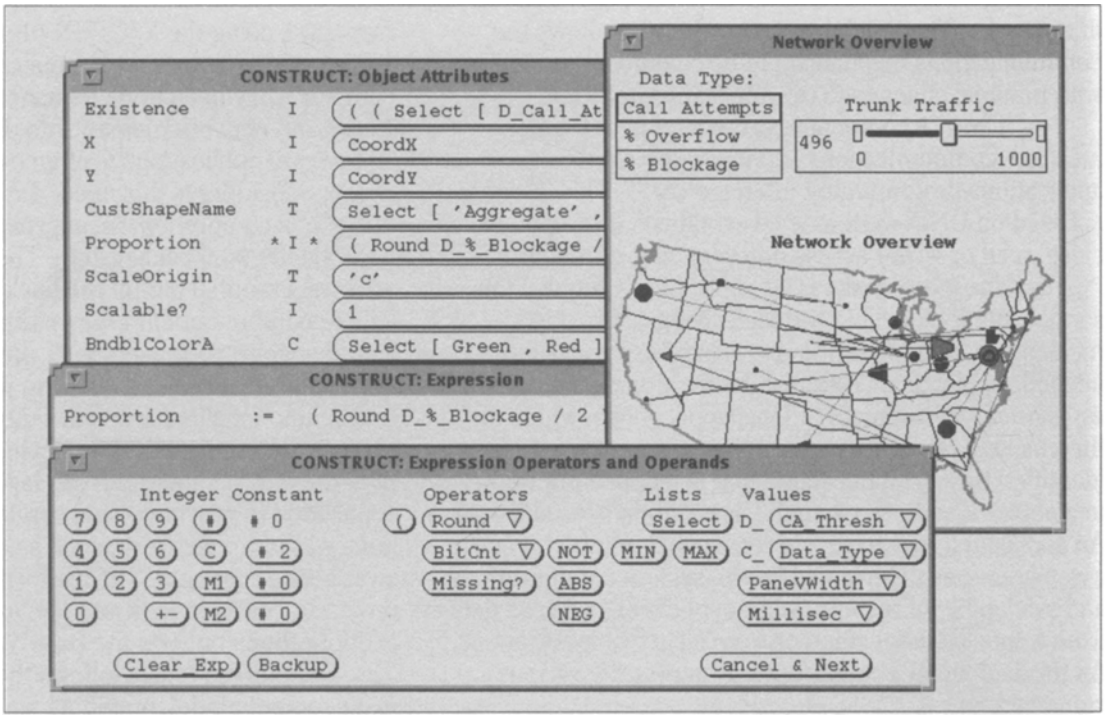

Figure 2. Display generated using the AT&T Display Construction Set, a user-interface management system. This system enables developers (and customers) to build and customize graphics displays that respond to user input and dynamic data. Even those with minimal time and programming skills can use the set of objects that the Display Construction Set supplies for reuse or can create their own objects.

- *Security management* via a resource-capability management system (a user-group security mechanism) that controls what administrative commands a user can run, or what applications users can access. A security-audit log is also provided.
- *Performance management* to collect application-performance data.

User-Interface Subsystem. Most applications require a user interface, which will vary depending on the application. For example, some applications require graphical user interfaces with bit-mapped graphics. Others require an ASCII user interface with either a form-and-menu interface or a simple command-line interface.

The platform provides reusable tools to help applications developers build both graphical and ASCII user interfaces. In addition, several of these interfaces connect to the communications backplane via a set of backplane connectors (i.e., connecting software). The reusable backplane connector or RBCs gives the application access to the platform's communications subsystem for the full range of communications and networking options already mentioned.

The BaseWorX platform supports development of graphics applications built with X Window System™ software, Open Look® software, and the Motif™ graphical user interface. (X Window System is a trademark of Massachusetts Institute of Technology. Open Look is a registered trademark of UNIX System Laboratories, Inc. Motif is a trademark of Open Software Foundation.)

In addition, the platform supports the AT&T Display Construction Set (DCS),[13] a user-interface management system that lets developers (and customers) build and customize graphics displays that respond to user input and dynamic data. These displays can be built with the rich set of objects that the Display Construction Set supplies for reuse, or users can create their own objects. Minimal time and programming skills are required to build displays with the Display Construction Set. Also, prototypes that use such displays can be converted quickly into a product simply by replacing the simulated data with live data and network connections. Figure 2 shows a sample display that was generated with the Display Construction Set.

To meet the character-interface needs of a variety of applications, the platform supports two ASCII user-interface packages. AT&T's Interpretive Frame System (IFS) provides[14] a powerful, task-oriented, high-level language for creating forms and menus and managing

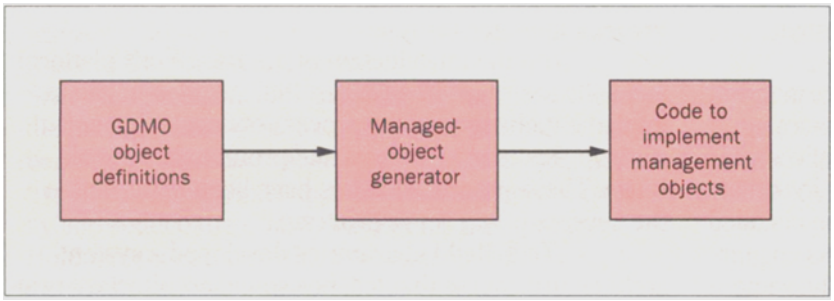

Figure 3. The managed-object generator uses object definitions to produce a set of C-language structures and encoders for use with the common communications platform (CCP). [The template syntax complies with the Guidelines for the Definition of Managed Objects (GDMO).] Applications developers can fill in or access these structures and let CCP handle the details of encoding, decoding, and interfacing.

applications in the UNIX system environment. JYACC's JAM™ user-interface manager enables developers to build ASCII user interfaces quickly and provide a common look and feel across a variety of host operating systems. (JAM is a trademark of JYACC, Inc.)

Database Subsystem. Most applications need to store information in a database and retrieve it in various forms. For example, large databases in telecommunications applications may contain records for all the lines in a switching office, or billing information for all subscribers. To provide this capability, the platform integrates several commercial DBMS packages.

Applications that use the platform can interact directly with vendor-supplied DBMSs. Alternatively, applications can go through a SQL server and be insulated from the DBMS. This server is the connecting software between the platform's communications and OA&M backplane and the database. The SQL server provides a message interface that is based on ANSI SQL. Databases can also be administered through the OA&M facilities that the platform provides.

Applications-Platform Tools. The BaseWorX platform also provides several development tools, including OSI system-management capabilities and support for multiple national languages.

As stated before, resources are modeled as managed objects in the OSI system-management model. Although applications are expected to interact with these managed objects in specified ways, writing the code to handle managed objects still can be a complex task.

The OSI system-management subsystem provides an application generator, called the managed-object generator (MOG), that helps applications developers deal with the complexities and diversities of managed objects. This reusable platform component, in turn, encourages the development and sharing of reusable objects among projects. As input, the managed-object generator accepts OSI-standard object-definition templates in a syntax from the TMN standard's Guidelines for the Definition of Managed Objects (GDMO).[11] From this, it produces a set of C-language structures and encoders for developers to use with CCP. (See Figure 3.) Applications developers can then fill in or access the structures and let CCP handle the details of encoding, decoding, and interfacing with the communications protocols.

Because of the increasing globalization of the telecommunications market, having reusable software components that facilitate the internationalization of applications is cost-effective. The BaseWorX platform provides multiple-national-language-support (MNLS) tools, which enable developers to build user-interface screens whose information can be displayed, as needed, in any of several languages. These tools support message cataloging and collating sequences. The MNLS interface complies with Issue 3 of the X/OPEN Company's portability guide.[15] These tools allow the system to change the language it uses to display or enter information after it is deployed, without having to recompile the system's software. The MNLS tools available to the applications developers are also used by the platform. These tools provide a library interface that:

- Supports functions for output (and, in the future, input) of strings, using message catalogs.
- Allows applications to switch between message catalogs by setting the locale. For example, a server whose clients use different languages would need this capability to switch languages for different requests.

Platform Integration

The BaseWorX applications platform integrates all these components into a unified product. Platform software is delivered as both a development environment and a run-time environment.

In the development environment, developers

create applications on the platform, using the components mentioned previously.

The run-time platform provides the corresponding execution environment for the target processors in the field. This environment is an independent deliverable that can be installed and run on its own. Applications that are based on the BaseWorX platform can then be installed and associated with the platform at run-time. This encourages the developers of multiple projects to use the same integrated platform in a consistent way, thereby maximizing reuse across products.

In addition, the run-time platform allows multiple applications to run on a single processor. When the design of these applications supports this paradigm, processors are used more efficiently and administration is easier across systems.

Reuse Support Services

The customer support provided with the platform is an important ingredient to overcome some of the cultural resistance to reuse. Documented policies for maintenance, software updates, support intervals, and strict upward-compatibility requirements give applications developers a feeling of confidence that the component interfaces will be maintained and the platform will continue to meet their needs.

A BaseWorX platform customer-support engineer is assigned to each project that uses the platform. This person helps the project define its architecture and use the platform appropriately in its development and run-time environments.

Communications with these customers are maintained through a "hot line," user-group meetings, and problem reports. Customer needs are ascertained through these mechanisms, and through a periodic needs analysis.

The BaseWorX platform training program gives these customers information about the platform, its components, and how they interoperate. The training program is effective in helping projects to start using the BaseWorX platform.

Software reuse is also supported through extensive documentation that is designed to help developers who build an application on the BaseWorX platform. The documentation includes a description of error and system messages, and a specification for the interfaces and capabilities of each component.

Process Changes for Reuse

With the introduction of the BaseWorX platform and software reuse in AT&T in 1987, we also began to implement changes to the process of developing software applications to support and promote large-scale reuse. These process changes have been important to the success of our reuse program.

AT&T Bell Laboratories developed a system-architecture course that teaches sound architecture principles. The course emphasizes that system architects:

- Have a clear understanding of the scope of the problem they are trying to solve with their new system.
- Address architectural issues (such as system performance, OA&M, and error recovery).
- Evaluate reasonable approaches to an architectural solution.

It also stresses the importance of platform and software reuse as a vehicle for addressing these issues and generating a solution. This course is presented to developers who are involved in the design of software systems. We have found that more formal training for our architects leads to better system architectures.

In addition, an *Architecture Review Board* was created to oversee the review of the designs of new systems and promote architecture-level reuse in these systems.[16] A separate review team assesses each project. New projects participate in an architecture review early in their development cycle, motivated by mechanisms such as the Operations Systems business unit's new-product introduction process.[17] These reviews focus on improving the quality of the application's design through emphasis on system-performance analysis and architecture reuse. Information about the platform is provided as input to these reviews to promote greater reuse.

The collected results of each review are presented to the Architecture Review Board so it may assess the architectural integrity of new systems, monitor progress toward reuse objectives, and evaluate trends in applications development.

A key to overcoming cultural resistance and achieving large-scale reuse has been the multipronged approach used in implementing the reuse program. The three aspects of this approach to promote the reuse program were:

- Gain the support of top management. As a result, we were able to establish policies that require new projects to participate in architecture reviews and,

wherever possible, base their application design on reuse of the BaseWorX platform.
- Share the results from "showcase" projects with middle managers to enlist their support. The cost versus benefit of large-scale reuse on real projects was effective in convincing managers to work to improve quality and reduce the time to market for their products.
- Hold one-on-one consultation sessions with the technical staff on new projects. Thus, staff members come to understand specifically how architecture reviews, software reuse, and the BaseWorX platform can help them be more productive and improve the quality of their product.

All three have been important in changing the culture in favor of large-scale reuse.

Results of Reuse

Significant results have been achieved with the use of the BaseWorX platform. We have applied these concepts to the development of a wide range of applications for the United States and international markets (for telecommunications and for manufacturing, medical-information systems, and business-office systems).

The BaseWorX platform has been reused in more than fifty software systems. These projects have ranged from five to more than one hundred developers, and from several thousand lines of code to over a half million lines. The platform has proved to be extremely flexible, allowing development organizations to select the architecture and components that provide the necessary functionality and performance.

Reuse rates have varied with the applications, but have reached as high as 90 percent on several projects. Equally, if not more, important has been the significant reduction achieved in time to market for these systems. Software reuse has been a major contributor to the interval reductions reported elsewhere in this issue for the new-product introduction process.[17] Systems with high-reuse rates have been delivered to customers in less than six months, an interval that was uncommon years ago.

In addition to the cost and time reduction realized with software reuse, we are realizing other benefits. By reusing time-proven software, the total faults in the delivered product are reduced. We are also finding that the architectures of these systems make future changes easier to accommodate and less costly.

Future Directions

Because of the potentially large benefits offered by software reuse, additional progress is desirable. Future work on reuse will focus on three areas:
- Remove further cultural barriers to reuse.
- Broaden the functionality of the platform by continually incorporating new reusable technology and tracking emerging international standards.
- Promote greater reuse at the application level.

A major barrier often cited is the programming culture that thrives on new and innovative output, instead of emphasizing business goals. While we have made good progress in combating the "not invented here" syndrome, future work is needed to understand this cultural aspect even more to promote better use of the talents of AT&T's people, its most important software resource. A specific area of interest is to design appropriate reward and motivation systems. Such systems would encourage greater reuse by giving greater recognition to those employees who achieve customer satisfaction and shorter time to market.

Also, we are committed to the continuous integration of new technology into the platform to make available the best reusable components from a wide variety of sources in the industry. We will continue to track and conform to evolving industry standards and consortia directions.

Another area being pursued is the availability of additional software components at the application level. Here, the goal is to spare applications developers the task of having to develop software from scratch. Instead, they should be able to use the platform, select application-level components from a software warehouse, and then expend most of their effort on assembling and testing applications and working with their customers to customize applications. This goal is being pursued as part of the Project Silver Bullet initiative.[1] Just as it was possible to create platform-level components in the BaseWorX platform, we believe it is possible to abstract and create additional components in specific application domains (e.g., to handle alarms), to achieve higher levels of reuse. However, this will require further advances in specifying, maintaining, and supporting reusable assets.

Acknowledgments

This paper presents the collective work of many members of the Software Technology Center at AT&T

Bell Laboratories, and its many valued customers and suppliers throughout AT&T. Their significant contributions are gratefully acknowledged.

References

1. S. J. Gelman, F. M. Lax, and J. F. Maranzano, "Competing in Large-Scale Software Development," *AT&T Technical Journal*, Vol. 71, No. 6, November/December 1992, pp. 2–11.
2. W. Tracz, *Tutorial: Software Reuse: Emerging Technology*, IEEE catalog number EH0278-2, IEEE Computer Society Press, Los Alamitos, California, ISBN 0-8186-0846-3, 1988.
3. K. J. Anderson et al., "Reuse of Software Modules," *AT&T Technical Journal*, Vol. 67, No. 4, July/August 1988, pp. 71–76.
4. R. P. Beck et al., "Software Reuse: A Competitive Advantage," *ICC '91, International Conference on Communications, Conference Record, Communications—Rising to the Heights*, Denver, Colorado, June 23–26, 1991, IEEE, New York, New York, 1991, pp. 1505–1509.
5. Y. S. Bagga et al., "Application Platform for Operations Systems," *NOMS '92, Networks Without Bounds, IEEE 1992 Network Operations and Management Symposium, Symposium Record*, Memphis, Tennessee, April 6–9, 1992, IEEE Communications Society, Piscataway, New Jersey, 1992, pp. 90–101.
6. AT&T Bell Laboratories, "BaseWorX™ Applications Platform Technical Product Description," 1991. To order this document, send your request via electronic mail to `attmail!software`.
7. American National Standards Institute, *Database Language: Embedded SQL, American National Standard for Information Systems*, ANSI X.3.168-1989, American National Standards Institute, New York, New York, 1989.
8. Technical Committee ISO/TC 97, *Information processing systems—Open systems interconnection, Basic reference model*, Reference No. ISO 7498-1984 (E), International Organization for Standardization, Geneva, Switzerland, 1984.
9. Institute of Electrical and Electronics Engineers, *IEEE Standard Portable Operating System Interface for Computer Environments*, ANSI/IEEE Standard 1003.1-1988. IEEE Computer Society, IEEE, New York, New York, 1988.
10. R. Prieto-Diaz and G. Arango, *Domain Analysis and Software Systems Modeling*, IEEE catalog number 91-11714, IEEE Computer Society Press, Los Amalitos, California, ISBN 0-8186-8996-X, 1991.
11. International Telegraph and Telephone Consultative Committee, "Recommendation M.30—Principles for a Telecommunications Management Network," *CCITT Blue Book*, Vol. IV, Fascicle IV.1, XIth Plenary Assembly, Melbourne, Australia, November 11–25, 1988, International Telecommunications Union, Geneva, Switzerland, 1989.
12. T. J. Dwyer, "TUXEDO™ Transaction Processing for 3B Computers," *AT&T Technology*, Vol. 3, No. 1, 1988, pp. 40–45.
13. J. P. Rotella, A. L. Bowman, and C. A. Wittman, "The AT&T Display Construction Set User Interface Management System (UIMS)," *CHI '92, Conference Proceedings, Human Factors in Computing Systems: Striking a Balance*, Monterey, California, May 3–7, 1992, P. Bauersfeld, J. Bennett, and G. Lynch (eds.), Association for Computing Systems, New York, New York, 1992, pp. 73–74.
14. K-P. Vo, "IFS—A Tool to Build Integrated, Interactive Application Software," *AT&T Technical Journal*, Vol. 64, N. 9, November 1985, pp. 2097–2117.
15. X/OPEN Company, *X/OPEN Portability Guide*, Issue 3, Vols. 1–7, Prentice Hall, Englewood Cliffs, New Jersey, ISBN 0-13-685835-X, 1989.
16. M. Hosein Fallah et al., "Development Process Audits and Reviews," *AT&T Technical Journal*, Vol. 70, No. 2, March/April 1991, pp. 99–108.
17. G. W. Arnold and M. C. Floyd, "Reengineering the New Product Introduction Process," *AT&T Technical Journal*, Vol. 71, No. 6, November/December 1992, pp. 12–19.

(Manuscript received July 17, 1992)

Roger P. Beck *is supervisor of the Application Reuse and Assets Group in the Software Platforms Department with AT&T Bell Laboratories in Columbus, Ohio. His group, which is part of the platform product team, develops software for the BaseWorX applications platform and develops architecture frameworks that will achieve high levels of application-software reuse. Mr. Beck joined the company in 1967. He has both a B.S. and an M.S. in electrical engineering from The Ohio State University (Columbus).*

Satish R. Desai *is supervisor of planning and architecture for the BaseWorX applications platform and is with AT&T Bell Laboratories in Columbus, Ohio. His group is responsible for planning, architecture, technology evaluation, systems engineering, and product directions for the BaseWorX platform. Mr. Desai joined the company in 1979. He has an M.S. in operations research from the University of California at Berkeley; has completed the in-house Bell Laboratories Masters' Work Program in computer science; and has a Ph.D. in systems engineering from The Ohio State University (Columbus).*

Doris R. Ryan *is head of the Software Platforms Department with AT&T Bell Laboratories in Columbus, Ohio. Her department is responsible for design, development, and support of the BaseWorX applications platform. Ms. Ryan joined the company in 1978. She has a B.S. in computer science from Cornell University (Ithaca, New York), and an M.S. in computer science from Stanford University (Stanford, California).*

Ronald W. Tower *is a distinguished member of technical staff in the Software Platforms Department with AT&T Bell Laboratories in Columbus, Ohio. He is responsible for the planning and architecture for the BaseWorX applications platform, and currently is working on performance analysis of the BaseWorX platform. Mr. Tower joined the company in 1985. He has a B.A. in mathematics from the University of West Florida (Pensacola, Florida).*

Dennis Q. Vroom *is supervisor of the Platform Development group in the Software Platforms Department with AT&T Bell Laboratories in Columbus, Ohio. His group, which is part of*

the platform product team, develops and integrates software for networking, OA&M, and the user interface for the BaseWorX applications platform. Mr. Vroom joined the company in 1981. He has a B.S.E.E. from the University of Tennessee (Knoxville) and an M.S.E.E. from Carnegie-Mellon University (Pittsburgh, Pennsylvania).

Linda Mayer Wood *is a member of technical staff in the Software Platforms Department with AT&T Bell Laboratories (Red Hill) in Middletown, New Jersey. As part of the Product Management Group, she is responsible for bids and proposals, customer negotiations, licensing terms and conditions, and contract preparation and review. Ms. Wood joined the company in 1981. She has a B.S. in chemistry and mathematics from the State University of New York at Binghamton and an M.S. in computer science from Stevens Institute of Technology (Hoboken, New Jersey).*