A Case Study:

Distributed Object Technology
at
Wells Fargo Bank

by Erik S. Townsend, The Cushing Group, Inc.

Based on Interviews with Mr. Eric H. Castain, Vice President, Wells Fargo Bank

This article is based on a 25-page white paper published by The Cushing Group, Inc. The full-length paper is available free of charge (softcopy) from The Cushing Group's Web site, http://www.cushing.com. Bound copies are available for a nominal fee, and can be ordered from The Cushing Group, Inc. at (603) 883-0130, or info@cushing.com

---

In the 1980s, competition for investor's funds traditionally kept in banks expanded to include brokerage and insurance companies. To remain competitive, banks began offering products such as mutual funds and brokerage accounts – products historically outside the purview of traditional banking. This would result in customers expecting their banking activities to be structured in terms of their overall *relationship* with the bank, rather than in terms of a single account they might own.

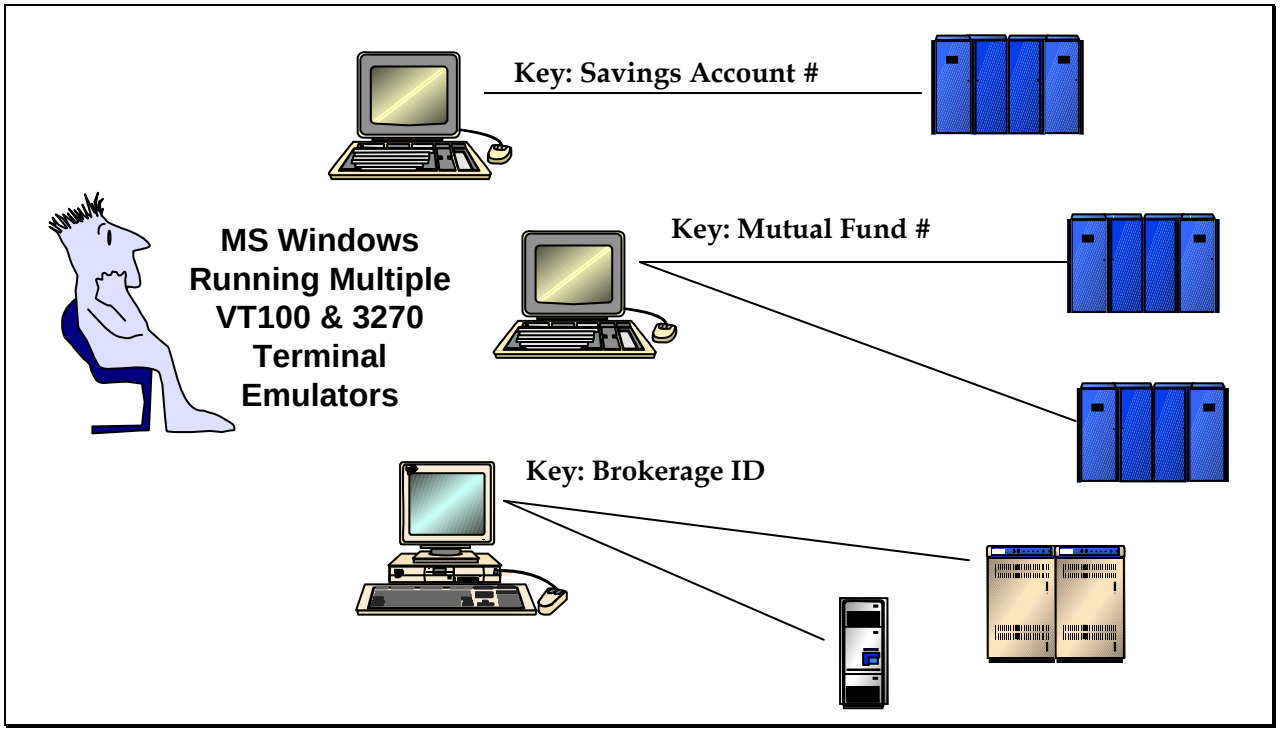

Figure 1 - Information Access Before

In the past, Wells Fargo's computer systems were optimized for account processing rather than providing integrated customer data. Various types of accounts were managed by completely different computer systems, ranging from Mainframes to VAXes to Tandems. A simple question from a customer such as *"How much money do I have in all of my accounts combined?"* could require an employee go to several separate different *systems of record*, none of which offered a graphical user interface. This became an impediment to efficient customer service.

Wells Fargo needed to seamlessly integrate these systems' functionality, and preferably deliver the integrated result through a graphical user interface. They began to evaluate approaches which could integrate data from several systems of record at a practical cost. A key objective was to deliver the integrated functionality through a *native* Microsoft Windows™ user interface.

In partnership with Wells Fargo, we scoped and defined a 3-month project to provide the telephone agents with a more integrated, intuitive, and easier-to-use interface to customer account information. The feasibility of using a three-tier client/server architecture to deliver a rapid solution of production-grade quality was confirmed by the project. The team chose to use Object Request Brokering (ORB) technology to meet these goals. Object request brokering offers a way to construct distributed application systems using object-oriented semantics to define an application-level communication protocol between client and server programs.

Wells Fargo's efforts would eventually evolve to become what is now almost certainly the world's largest CORBA-based production application system[1]. This technology would later enable Wells Fargo to

[1] We are aware of no other CORBA-based application system in production use which rivals the size and scope of the ORB-based applications at Wells Fargo. However, our experience has also been that many organizations working with this technology have avoided public disclosure of their efforts because they view the technology as a competitive advantage. If any effort of similar scale exists, its secrecy has been well guarded.

become the first bank to offer Internet-based access to account balances, not to mention several other innovations.

The Customer Relationship System

When a customer who owns several accounts calls the bank, they should not be required to know their account numbers. The customer will in many cases prefer to transact business in terms of their overall *relationship* with the bank, which might involve several different accounts. The existing systems did not allow the agent to specify the identity of a customer (Social security number, name, etc.) and then see a complete view of that customer's overall relationship with the bank - all accounts owned, their balances and status, etc.

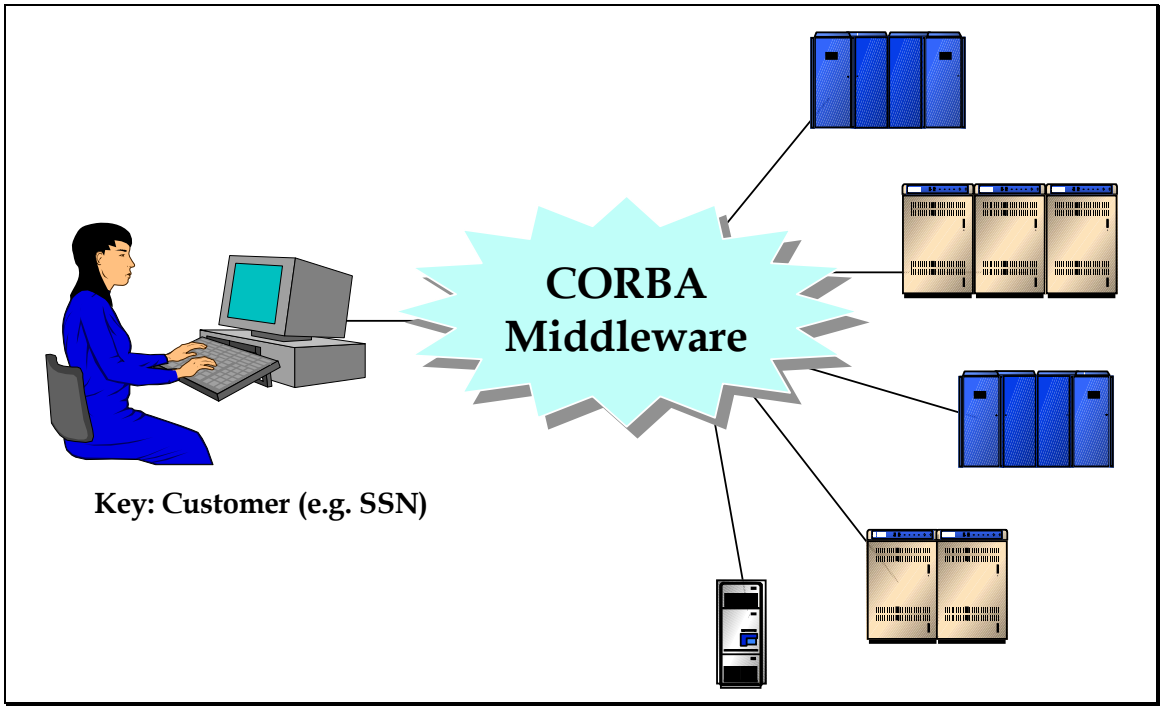

Figure 2 - Information Access After

The project team used *Object Request Brokering (*ORB) technology to build an application that allowed the service agent to retrieve a comprehensive profile of customer account information simply by entering the customer's social security number. The system then interrogated several systems of record including IBM MVS/CICS hosted applications, an application running on a DEC VAX/VMS system, and a Unix-based system. An organized, coordinated view of the customer's *relationship* with the bank was returned to the user and presented through a Microsoft Windows™ GUI. This application was named the *Customer Relationship System*, or CRS, reflecting its function of presenting all accounts in a customer's *relationship* with the bank.

Designing the CRS Application Using Distributed Object Technology

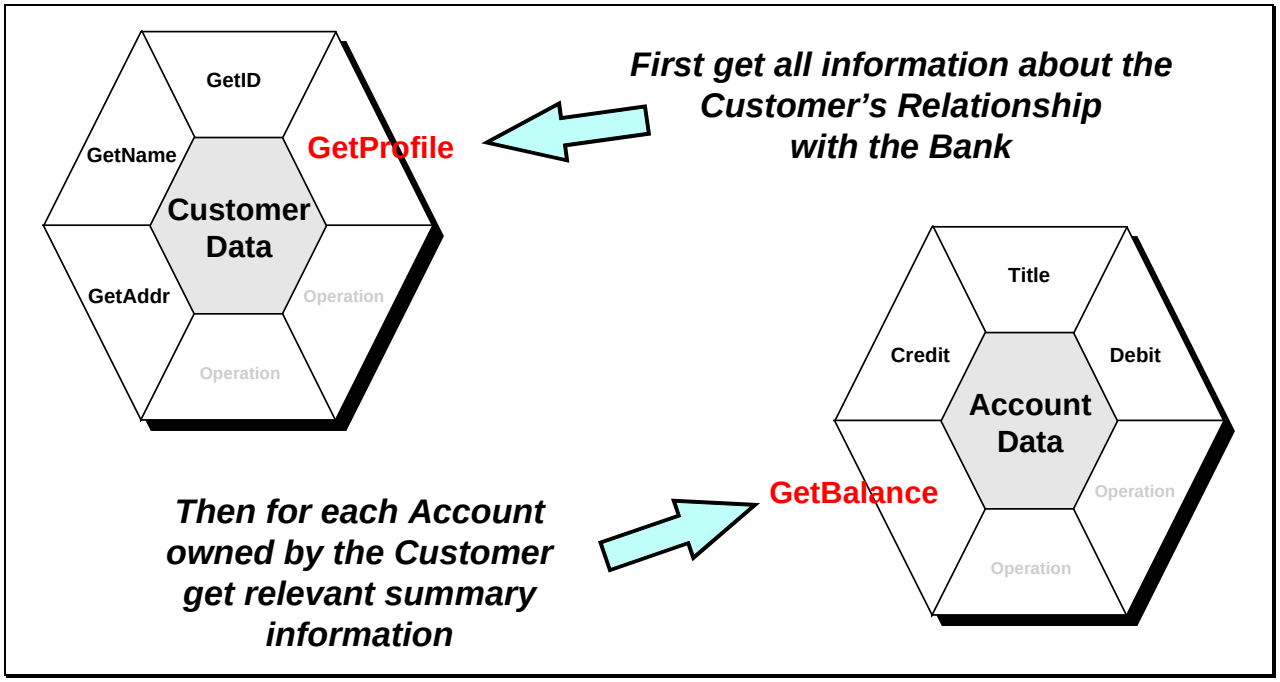

Figure 3 - Simplified View of Customer and Account Business Objects

The CRS application used distributed object computing to access functions of the bank's systems of record. These systems were exposed as CORBA distributed objects (Figure 3).

While the GUI designer worked with the users, the other consultants concentrated on defining the Business Object model, and mapping that model to the existing legacy applications' functions. UNIX-based methods were then developed to implement the business objects by relying on functions of the legacy applications. As the object model took shape, each operation was mapped to the corresponding mainframe application function, and data fields that would have to be accessed in order to perform its function were identified.

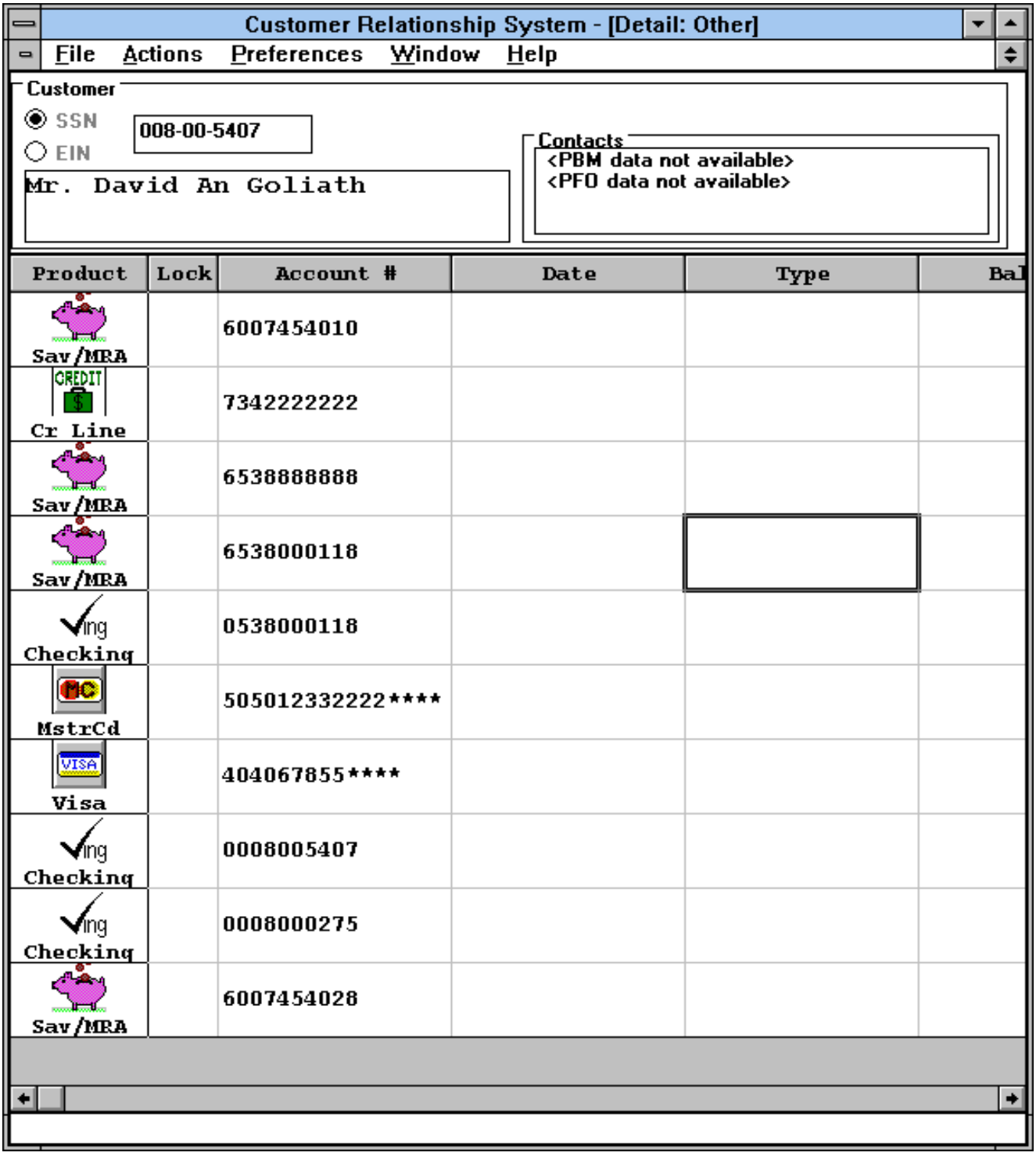

Figure 4 - The CRS Application Screen after entering a customer's Social Security number

The resulting application provided Wells Fargo's telephone customer service agents with a Microsoft Windows™-based graphical user interface to customer relationship information (Figure 4). The only necessary input from the end user is the customer's social security number.

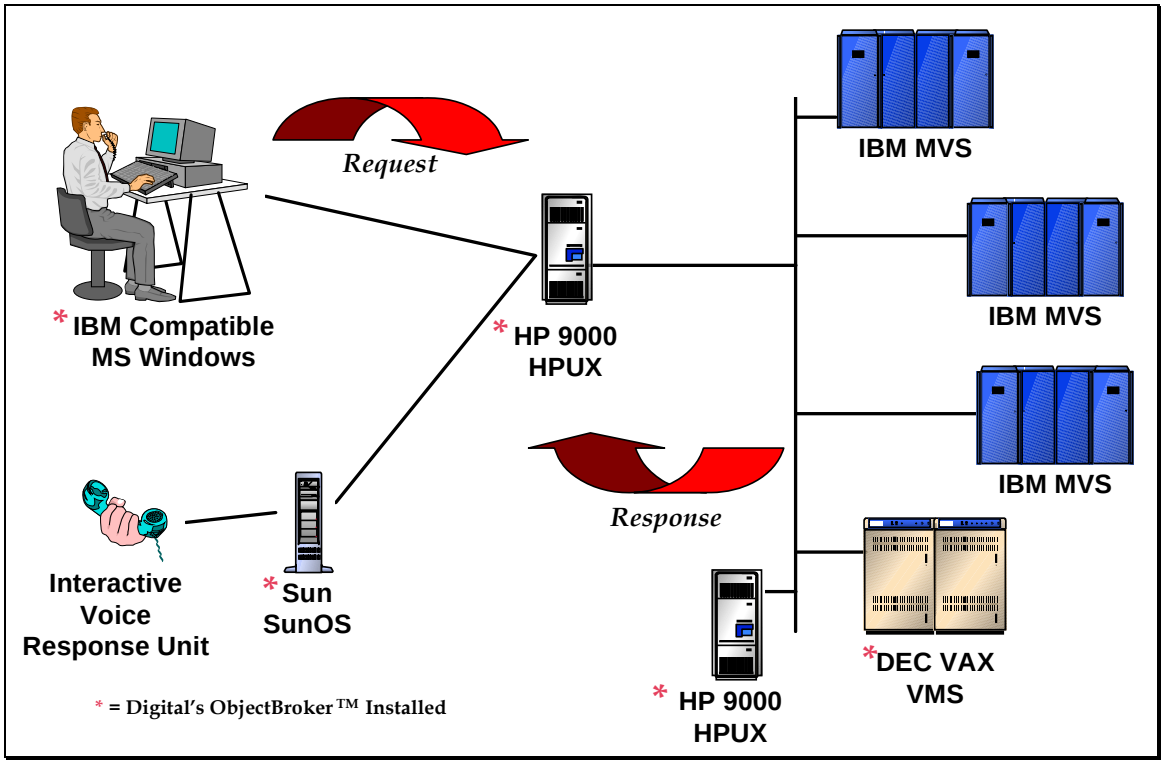

Figure 5 - Simplified System Diagram of the CRS application[2].

Digital's ObjectBroker™ was used to provide communication between applications on all platforms *except* the IBM Mainframe, which ran MVS (Figure 5). The CORBA servers running on the Unix platform implemented these business objects by communicating with one or more mainframe applications using non-CORBA mechanisms.

The CRS Experience in Retrospect

The CRS application was placed in full production use in January 1994, one hundred four days after we first arrived in San Francisco in early October. Our original target was 90 days to full production deployment. We feel that several last-minute additions of functionality (not to mention the winter holidays) justified the slip to 104 days. Wells Fargo invested in licensing the ORB middleware, 47 person-weeks of consulting services, and the personnel cost of three Wells Fargo employee participants. The result was a working solution that significantly re-shaped the way users interacted with the bank's core systems of record.

Expanding on CRS: IVRU, Brokerage, ATM, and Quotron

Fueled by the success of CRS, new ideas and opportunities emerged to use ORB technology to deliver new innovations to Wells Fargo's customers. These included:

- Addition of an *Interactive Voice Response Unit* which allowed Wells Fargo customers to interact more directly with the CORBA-based infrastructure

[2] The first version of the CRS application offered client access through a Microsoft Windows™ GUI only. The interactive voice response unit was later added as an additional client access channel, and is discussed later.

- Access to customer brokerage accounts (not part of the original CRS system) was added, which involved integrating the Tandem platform
- Automated Teller Machines (via a connection through the mainframes) were added as clients to provide access to customers' brokerage account balance information
- A stock market data application was purchased from Quotron was encapsulated and exposed as an ObjectBroker™ server, making real-time stock quotes available to client applications.

Through these projects, several new innovations were delivered to Wells Fargo customers. But perhaps more importantly, the object model was refined, yielding a robust library of truly re-usable business object services which encapsulated the bank's core systems of record. Wells Fargo was now poised to fully exploit the principal benefit of distributed object technology: Re-use of business object services to satisfy a new business need at a fraction of the cost and time-to-market which would be required using a traditional, ground-up approach to solving the same problem.

The World's First Internet-Enabled Bank

By March of 1995, the World-Wide Web (and the Internet on whole) had become prominent outside of the computer industry. Wells Fargo wanted to deliver on-line electronic banking services to customers over the Web, rather than limiting its Web site to containing product literature, as was common at the time.

Sixty days later, Wells Fargo was on-line with real-time access to account balances via the Web, and in so doing, became the first truly Internet-accessible bank.

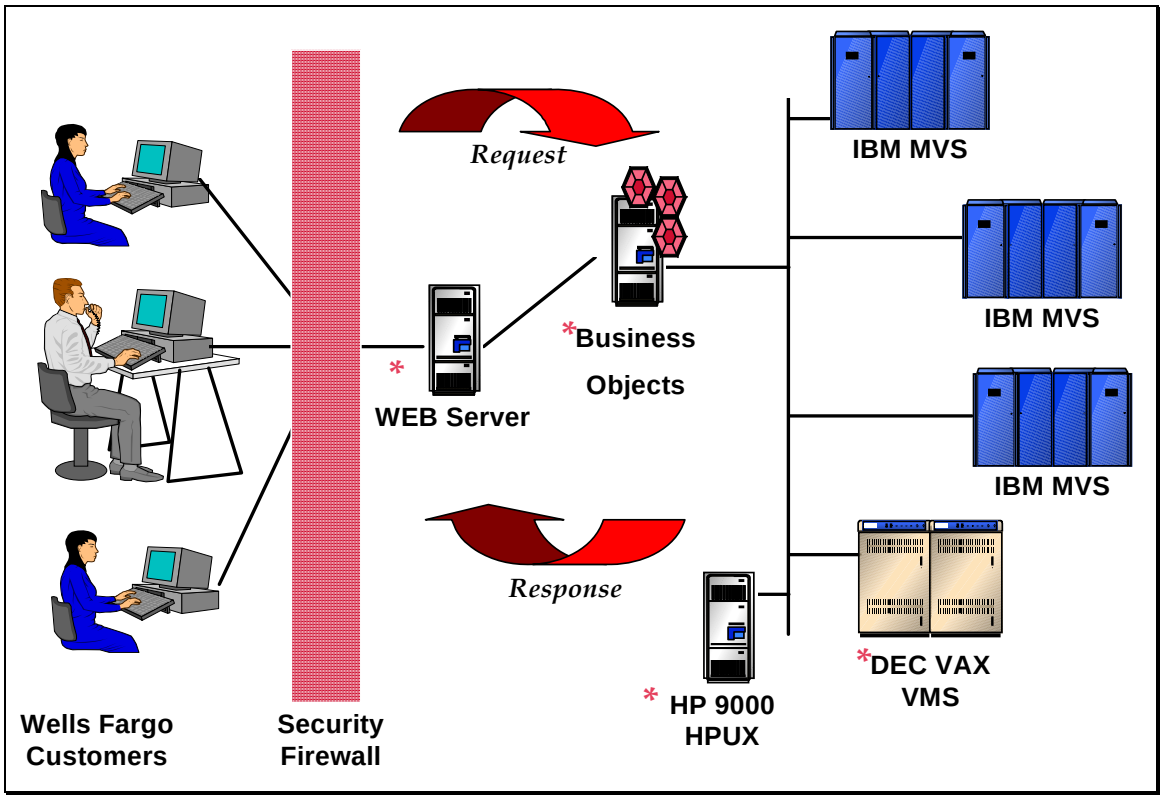

Figure 6 - Wells Fargo's first Internet Banking Solution - Implemented in 60 days.

The amazingly short 60-day project cycle was directly enabled by the re-usable business object services developed for CRS and other applications. The legacy systems which needed to be integrated had already

been exposed through distributed object interfaces. If not for security issues, the project would have involved little more than hooking up a Web server as an ObjectBroker™ client. Of course, security *is* of paramount importance to a bank, so the customer authentication and authorization checks, firewall protection, and so forth became a significant task - probably the lion's share of the project on whole.

Making Object Re-Use Work in a Large Enterprise

The key to enabling re-use lies in having, maintaining, and sharing a well-architected Enterprise object model. The authors believe that Wells Fargo's success is owed largely to the fact that they invested in developing such a model, and are very careful to maintain, extend, and use it as effectively as possible.

Wells Fargo has commissioned an *ORB Coordination Group*. Essentially, the function of this small group of people is to be the keepers of the Enterprise Object Model. Rather than allowing anyone so inclined to simply define, build, and use object interfaces, all development groups using the object middleware are required to work with the ORB Coordinator to ensure that their object definitions are not incompatible, overlapping, or redundant. This approach has worked very well, and has allowed Wells Fargo to effectively coordinate the work of several dozen programmers.

Managing the Run-Time Environment

Wells Fargo now depends on its CORBA-based infrastructure as a key component of its operational systems environment. Like other mission-critical systems, the ORB servers support the bank's ability to conduct business. Service outages are simply not acceptable. To support such mission-critical use, the run-time environment must be managed and controlled. Because commercial ORB products offer few if any of the features needed in this area, Wells Fargo has developed its own ORB System Management Facility, which is the subject of David Newman's article, also in this issue.

Organizational Issues: Half the Battle

Wells Fargo's success is largely attributable to their management's willingness to explore different approaches, and create an environment conducive to successful re-use of application software components.

The Internet application that was built at Wells Fargo provides an example. The goal was to quickly build the application and be the first to market. This was only attainable by the re-use of components already built for the customer service agents' desktop application. That re-use was enabled by management's willingness to view the object model and the servers which implement it as a core asset upon which other systems would be built. Without this management commitment, the developers would probably have built the Internet application from the ground up, at considerable cost and time-to-market penalty.

A very successful technique is to re-structure the organization itself to reflect the object model of the business. This relatively new technique is becoming known as *class-based organizational structure*, and offers an innovative way to structure an organization to truly promote and enforce reusability of software components.

Organizational readiness is also a key success factor. Wells Fargo made the following investments to support its use of distributed object technology:

- **Formal training.** Since 1993, over 100 Wells Fargo employees have been trained on the use of CORBA.

- **Mentoring.** Both consultants and experienced Wells Fargo employees have been used to formally mentor other groups within the bank on the design of ORB-based applications and how to take advantage of ObjectBroker™ features. This ensures that the technology is applied in a uniform and concerted manner.
- **Formal re-use coordination.** A key success factor at Wells Fargo has been the ORB Coordination Group discussed earlier.

Validating the Approach: An Outside Audit

With the use of ObjectBroker™ growing at an exponential rate in Wells Fargo, it was realized that the trend was moving rapidly toward reliance on CORBA as a standard software integration mechanism for several key applications. Wells Fargo does not take lightly the importance of risk management, and it was felt that the CORBA-based approach should be thoroughly audited before allowing its use to become more widespread within the bank.

In the spring of 1996, Mr. Dudley Nigg, Executive Vice President of the Direct Distribution Group at Wells Fargo, commissioned *KPMG Peat Marwick and Hewlett Packard* to conduct an in-depth evaluation of the prudence of using CORBA as a tool for high volume transaction processing. KPMG and HP's assignment was essentially to answer two questions:

1. Was Wells Fargo prudent in choosing CORBA as the basis of its extensive systems integration activities, and is it prudent to continue doing so?
2. If the answer to Question #1 is affirmative, was the selection of Digital's ObjectBroker™ appropriate, and should Wells Fargo continue to rely on Digital's product?

The study was completed and its findings were delivered to Wells Fargo management on August 5, 1996. The findings were as follows:

On question #1 (use of CORBA):

> KPMG found that Wells Fargo's selection of CORBA-based object-oriented systems integration was sound. After evaluating the risks and benefits, KPMG recommended continued use of CORBA technology.

On question #2 (selection of Digital's ObjectBroker™)

> After evaluating several competing products, KPMG found that Digital's ObjectBroker™ was a proper choice for Wells Fargo. During the evaluation, KPMG and HP evaluated ObjectBroker™'s performance and scalability, and found them to be adequate and in some respects superior to some alternative choices.

Wells Fargo Today: Banking Leadership Enabled by Technology

Wells Fargo has clearly become a technology leader in banking. We believe that distributed object technology has played a key role in helping Wells Fargo to achieve this leadership position. Although the original CRS application has now been replaced with an expanded, more versatile application, the same distributed, object-oriented architecture continues to be the mainstay of Wells Fargo's efforts to deliver innovative new channels for customers to conduct business with the bank.

Large-Scale Distributed Object Technology Leadership

Since it's inception, Wells Fargo Bank's Web-based banking service has grown to over 100,000 enrolled customers. At the time of this writing, Wells Fargo's ORB servers were processing as many as 200,000 business object invocations per day in the production environment, and at peak periods, as many as 300 simultaneous Internet customers were logged into the Web site. The ORB actually processes upwards of 750,000 CORBA method invocations daily, since several processes must interact with one another in order to process a single business object operation. The Wells Fargo team anticipates the need to support several million business operation invocations per day, based on projected growth estimates for 1997.

CORBA has from time to time been criticized in the industry as being "not ready for prime time" or "unable to support production-scale transaction volume", and has fallen victim to a host of other subjective, unsupported opinions. Wells Fargo appears to have disproved these arguments.

Conclusions

Distributed object technology is an enabling tool which has helped Wells Fargo to become a technology leader in banking, and to deliver innovative, competitive electronic access channels to its customers. What has occurred at Wells Fargo is best characterized not as the adoption of CORBA, but rather as the realignment of the organization to embrace a service-based, object-oriented distributed computing architecture.

The Financial Services sector is essentially an information industry. As the general public becomes more technologically astute, the quality and robustness of financial institutions' computer systems will emerge as a principal competitive advantage factor. Wells Fargo's willingness to embrace the latest technologies and adjust the organization to fully exploit them will help position the company as a leader, as the financial services industry continues to evolve.
