

Association for Information Systems AIS Electronic Library (AISeL)

ICIS 2010 Proceedings

International Conference on Information Systems
(ICIS)

2010

IBM POWER SYSTEMS AND SERVICE ORIENTED ARCHITECTURE AT BANK OF AMERICA'S FOREIGN ITEMS SYSTEMS OFFICE

Keng Siau

University of Nebraska-Lincoln, siauk@mst.edu

David DeWester

University of Nebraska-Lincoln, david_dewester@yahoo.com

Follow this and additional works at: http://aisel.aisnet.org/icis2010_submissions

Recommended Citation

Siau, Keng and DeWester, David, "IBM POWER SYSTEMS AND SERVICE ORIENTED ARCHITECTURE AT BANK OF AMERICA'S FOREIGN ITEMS SYSTEMS OFFICE" (2010). *ICIS 2010 Proceedings*. 226.
http://aisel.aisnet.org/icis2010_submissions/226

This material is brought to you by the International Conference on Information Systems (ICIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in ICIS 2010 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

IBM POWER SYSTEMS AND SERVICE ORIENTED ARCHITECTURE AT BANK OF AMERICA'S FOREIGN ITEMS SYSTEMS OFFICE

Teaching Cases

Keng Siau

University of Nebraska-Lincoln
Department of Management
College of Business Administration
Lincoln, NE 68588-0491
ksiau@unl.edu

David DeWester

University of Nebraska-Lincoln
Department of Management
College of Business Administration
Lincoln, NE 68588-0491
ddewester2@unl.edu

Abstract

This is a teaching case on a real life scenario in an organization involving IT solution evaluation and selection. This case involves the Bank of America's Foreign Items Systems office in Toronto. The officers from the Bank of America's Foreign Items Systems office in Toronto were discussing how to respond to a Request for Proposal (RFP) from a major U.S. bank to provide foreign currency services. The case discusses three options: Standard J2EE Web Application Global Foreign Currency, Modified J2EE Web Application with Single Sign On, and Service-Oriented Architecture (SOA) via Web Services. The case can be used to complement chapters in MIS and IT courses that are related to Systems Integration, IT Infrastructure or IT Implementation. The case can also be used to discuss IT planning, management, evaluation, and trade-off issues. The case is intended for use by senior undergraduate and MBA students.

Keywords: Systems Integration, IT Planning, IT Infrastructure, IT Implementation, Service Oriented Architecture, J2EE

It was a pleasant autumn day in Toronto in 2004. Raymond Yeung and Anna Watcher were having a Chinese meal of dim sum at a famous restaurant in downtown Toronto. They were planning to have a meeting with their colleagues after lunch to discuss how to respond to a Request for Proposal (RFP) from a major U.S. bank to provide foreign currency services. Raymond and Anna worked at the Bank of America's Foreign Items Systems office in Toronto. The Bank of America's Foreign Items Systems office was supporting the Bank of America Global Banknote Services business unit, which was being considered as a provider of foreign banknote services to a major U.S. bank. We will name the major U.S. bank as X-Bank in this teaching case (the real name of this major U.S. bank is concealed for security reasons).

Introduction

In 2004, a major U.S. bank, renamed X-Bank in this teaching case, needed a supplier for foreign banknote services. Any foreign banknote services provided by X-Bank would be outsourced to the supplier for processing. Bank of America's Global Banknote Services specialized in banknote foreign exchange, while the Bank of America's Foreign Items Systems office provided system support.

The Bank of America Global Banknote Services standard offering for financial institutions was the Global Foreign Currency (GFC) system, a Web-based application allowing clients to input transactions to the Bank of America's Foreign Items Systems. This application was fully integrated with the Bank of America back-end systems and allowed transactions to be processed automatically from inception to fulfillment.

One challenge in providing foreign currency services to X-bank was foreign currency orders had to be placed to X-Bank's system (where X-Bank's client account is stored) and to Bank of America's foreign banknote services system simultaneously. Bank of America's processing and operation departments would access the Foreign Item Systems transactions to fulfill the transactions according to the service agreements. The requirement was to make the transactions available to both X-Bank's platform and Bank of America's Foreign Items Systems office. The goal was to deliver a solution with the best return of investment to both organizations.

This case is based on true events in 2004. A few years later, Bank of America's Foreign Items Systems office would win an award for their solution to this case. The Foreign Items Systems office has a long history of solving problems with IBM Power Systems, a family of IBM servers.

In 2006, Bank of America won the award for creative use of IBM Power Systems:

... Canada Branch integrated all of its business software on Power Systems and increased the number of transactions processed on Power Systems by 115 percent. This reduced its IT maintenance staff requirements by 22 percent and doubled the business revenue of one of its key business units. The bank has trusted Power Systems and its predecessors to run its in-house-developed business applications for over 20 years and was today honored with the award for Power Systems Leadership. (IBM, 2006)

They won another award in 2007, this time for their solution to the problem discussed in this case:

Bank of America's Foreign Items Systems office, based in Toronto, successfully implemented a Service Oriented Architecture and Web services solution that provided a return on investment of over 385 percent and an estimated 32 percent savings in operating cost after the first year. The solution, powered by Power Systems, supports the integrated foreign currency banknote needs between Bank of America's Global Banknotes division and a top U.S.-based banking customer. Bank of America successfully integrated two disparate systems with transaction flow in real-time, enabling support for a network of over 2,000 branches and tens of thousands of tellers. The company has earned the award for Solution Innovation. (IBM, 2007)

Although we can look back now and know that the Service Oriented Architecture (SOA) solution pursued by Bank of America's Foreign Items Systems office was successful, it was not clear at the time that it was the best solution. SOA was in its infancy and was constantly evolving when Bank of America's Foreign Items Systems office considered it. They also looked at another approach, the Java 2 Platform, Enterprise Edition (J2EE) set of technologies. J2EE was established and tested, and the office had experience implementing business functions in J2EE.

IT-based problems often have many possible solutions, none of which is perfect or ideal. Based on trade-off analyses, each solution carries a certain amount of risk. This case presents the three options the Bank of America's Foreign Items Systems office considered: the Standard Global Foreign Currency J2EE Web application, Modified Global Foreign Currency J2EE Web application with Single Sign On, and Service Oriented Architecture (SOA) via Web Services.

If you were in Bank of America's Foreign Items Systems office in 2004, what would you suggest? Which systems integration approach would you pursue? Would you prefer to use a new technology like SOA or more established technologies?

This case study will take you through the SOA solution implemented by Bank of America's Foreign Items Systems office and also provide information on SOA and other technologies relevant to this case. At the end of the case, we will revisit the above questions.

Background

History and Organization of Bank of America

Bank of America was formed in 1998 when two U.S. banks, NationsBank and BankAmerica, merged. Both banks had a history of mergers and acquisitions and Bank of America continued that tradition by acquiring other financial institutions such as FleetBoston Financial in 2004, MBNA in 2005, and Countrywide and Merrill Lynch in 2008. In 2009, Bank of America has more than 5,800 branches and operates in more than 20 countries across the Americas, Europe, and Asia. In 2007, their revenue was USD66 billion. The BankersAlmanac Web site ranks Bank of America as the 14th largest bank in the world (BankersAlmanac, 2008) before the Countrywide and Merrill Lynch acquisitions.

Foreign Currency Services

Bank of America has several divisions. In addition to supplying banking services to individuals, Bank of America also provides services to businesses and other banks. One set of services involves foreign currencies. In addition to obtaining regular cash, companies that conduct international business need foreign currency services such as obtaining and storing foreign cash.

Every day, the equivalent of almost USD 4 trillion passes through the global foreign currency market. Although governments and businesses exchange money directly, banks handle most transactions. Banks need foreign currency both for internal use and to support business customers.

Many larger banks participate in foreign currency exchange in order to make money through currency exchange rate fluctuations. Because exchange rates between currencies change constantly, banks can make a profit by converting money to another currency and then converting the money back into the original currency.

Banks also engage in foreign currency trading on behalf of their corporate customers. Businesses that buy, sell, and trade with firms in other countries need a variety of foreign currency services. When arranging an exchange of goods or services for payment with a business in another country, the currency of payment can affect the profit of the deal. Businesses therefore need consulting services on the best currency for making payments. Making payments for goods or services can also require the use of a bank in order to maximize profits. By converting money to and from different currencies, and making payments in the correct currency, businesses can avoid fees such as charges assessed by credit card companies for making payments in another currency.

Businesses also need to protect themselves from the negative effects of exchange rate fluctuations. If exchange rates change between the time a company is billed and the time they pay the bill, the company could lose money. Companies can, however, make the exchange rate work for them by buying currencies when the exchange rate is favorable and then using the purchased currency to pay bills as needed.

Providing foreign currency services to a company therefore involves two sets of services: one set of services allows companies to buy and sell foreign currencies as needed and the second set of services provides methods for companies to insulate themselves from the negative effects of exchange rates. Bank of America provides both sets of services to banks and corporations.

Use of Technology and Power Systems at Bank of America

Banks have always been heavy users of technology to provide reliable security and service. In the past, banks were the first to use security technologies such as locks that could only be opened at certain times and safes made of metals with different melting temperatures to self-seal any holes cut with torches. Today, banks use anything from GPS tracking of armored cars to retinal scans on safes in order to secure money and other valuables. In the 1980s, banks used telephone-based systems to allow customers to access their accounts from home. In the 1990s, Bank of America mailed floppy disks back and forth from customers' homes so customers could access their accounts on their computers. Today, banks provide Web-based and mobile access to accounts, and Web-enabled ATMs to improve customer service.

Corporations are now moving toward smaller computers for some uses, but they continue to use large enterprise computers or servers for transaction-intensive purposes. In the 1950s and 1960s, enterprise computers, such as mainframes, were the only option available for corporate computing. As personal computers were introduced, companies found them useful as employee workstations, but many companies continued to use servers for their back-office computing. Servers – with their powerful computing capabilities, disaster recovery, self-healing abilities, and support for virtualization and service-oriented architecture – continue to have an important role in modern IT departments.

The Bank of America's Foreign Items Systems office has used IBM servers for over two decades and continues to use them today. They have solved many IT problems with IBM computers and they continue to find award-winning solutions to IT problems today.

In 1986, the Bank of America's Foreign Items Systems office began developing custom applications to run on their IBM S/36 system in order to automate several manual processes. When they purchased IBM's popular AS/400 computer system in 1990, they were able to easily migrate all of the applications to the new system by using the AS/400's S/36 emulation and native mode. Further, by using real-time mirroring software (i.e., one of earliest adopter of MIMIX in the summer of 1990), they were able to provide Business Continuity Operation by constantly mirroring all data from the production systems to a remote backup data center. Throughout the 1990s, Bank of America's Foreign Items Systems office continued to develop applications to run on IBM computers. In 1993, they developed a system that printed documents with magnetic ink. Computers could then read printed checks and automatically process those checks. A variety of bulk processing and data management solutions were developed resulting in increased efficiency and lower costs.

Bank of America's Foreign Items Systems office programmers found that asynchronous processing of data frequently had advantages. By allowing applications to read and write data independently of each other, applications would not spend idle time waiting for each other. If applications are no longer communicating directly with each other, then it becomes necessary to encode data in such a way that all applications can read from and write to data repositories. This can be done using data-queues, as Bank of America's Foreign Items Systems office did from 1996-1999 when they built new Customer Relationship Management (CRM) and Enterprise Resource Planning (ERP) applications. Processing of data encapsulated in standard formats is one of the basic characteristics of how Web services and Service-Oriented Architecture work.

Bank of America's Foreign Items Systems office has used IBM technology to continually improve their Foreign Currency services. In 1994, a bulk foreign-exchange payments solution was developed. By 1998, a JAVA-based system was used to serve foreign exchange rates to clients, and in 2001, IBM's WebSphere Application Server was used to securely make Power Systems' applications available over the Internet with the Global Foreign Currency product line. Further improvements to the Global Foreign Currency system were made in 2002 and 2003. The enhancement done in 2004 and 2005 was the reason for Bank of America's Foreign Items Systems office winning IBM's Innovation Awards in 2006 and 2007.

Exhibit 1 presents a list of milestones Bank of America's Foreign Items Systems office achieved using IBM computers.

IBM Power Systems

Bank of America had been using IBM servers for many years so the Foreign Items Systems office proposed using IBM Power Systems to provide X-bank with foreign currency services.

IBM produces many families of computers, one of which is Power Systems. Power Systems is a very powerful and popular family of IBM servers. The Power Systems are capable of running a currency exchange system that communicates with 2,000 branches of X-Bank while simultaneously processing the requests through Bank of America's other systems.

Power Systems can run the IBM i and its "cousin", the System p (also known as AIX). Computers need an operating system in order to function, and AIX and IBM i are two of the best operating systems available today. The best capabilities of these two families were recently combined to form IBM's newest line of computers, Power Systems. Power Systems also provide extensive support for a popular operating system called Linux.

Possible Solutions to the Problem

Let's review the problem faced by Bank of America's Foreign Items Systems office. It is the year 2004. The Bank of America's Foreign Items Systems office in Toronto is trying to develop a proposal to process requests for foreign currency services from thousands of branches of X-Bank. A number of options were available to integrate the systems from Bank of America's Foreign Items Systems office and X-Bank, so how did Bank of America choose? A brief look at the history of Bank of America's Foreign Items Systems office helps us understand how they made this decision.

The Bank of America's Foreign Items Systems office had supported the technology behind the Foreign Banknote business for more than two decades. After the millennium, Bank of America decided to grow the business and to explore new frontiers. They had commissioned Foreign Items Systems office to build an enterprise-level Web application to service Foreign Banknote business customers. The key requirements and decisions were to use the "Internet", "browser", "Java", "Model-View-Controller pattern", "industrial standards and best practices", and others as the backbone components instead of proprietary methods.

The J2EE (Java 2 Platform, Enterprise Edition) Web application, Global Foreign Currency system, running on IBM WebSphere Application Server environment, was built and launched in early 2001. This Global Foreign Currency system Web application allowed Foreign Banknote Services customers to do business with Bank of America Global Banknote Services anywhere and anytime. The application has been a very successful product that allows the Foreign Banknote Services to grow year after year.

Technologies Behind the Solutions

Before presenting the three possible solutions to the problem, the technology behind the solutions is presented. Two of the solutions rely on the J2EE platform and the third solution relies on SOA.

About J2EE

Java is a very popular programming language developed by Sun in 1995. As with any language, it is possible to write programs in Java from scratch. As with many languages, integrated development environments (IDEs) exist

that assist developers in rapidly writing programs. Unlike most languages, however, Java has optional components that support the rapid development of very large programs.

The Java 2 Platform, Enterprise Edition, called J2EE at the time of this case but currently known as Java EE, is a set of technologies that allow enterprise-level programs to be developed much more rapidly than would be the case if programmers actually wrote every line of code by hand. J2EE is not just the Java language, but includes a very large set of objects and other tools that allow for rapid development of very large programs much more quickly than can be achieved using other languages. It has built-in code libraries that support networking and distributed operations, which makes it relatively easy to write code that will allow the X-Bank branches to communicate with Bank of America. J2EE is very popular with many corporations and talented programmers are in strong demand. Bank of America's Foreign Items Systems office had experience implementing business functions using J2EE at the time of this case. Additional information on J2EE can be found in Exhibit 2.

About Service-Oriented Architecture

Service-oriented architecture (SOA) is a method for enabling geographically distributed computers to work together through services. A service is a business function that has been converted into a program. The program provides an interface that allows other computers to send data in a standard format to the service, which then acts on that data in a prescribed manner and sends the result back to the first computer. An SOA system consists of one or more computers providing one or more services to other computers. SOA does not specify how the data is to be exchanged between the computers. Networks and the Internet are common methods of exchanging data, but they are not the only ones.

To use SOA, two companies need to use either an existing industry standard format for messages or develop their own. Web-based SOA is usually platform independent, both in terms of hardware and operating system. The two systems at each company need to be able to format and process messages according to the agreed-upon standard, but there is a lot of flexibility in how the systems accomplish these tasks.

SOA can be designed to provide a platform-independent system of exchanging data with fast response times. The SOA solution would not be proprietary and the implementation could cleanly separate the business logic, presentation, and data components (i.e., three-tier architecture) of the system, but it was a new technology in 2004 and therefore subjected to unforeseen problems. Pursuing an SOA solution at that time might have been risky.

Additional information on SOA can be found in Exhibit 3.

Possible Solutions

Three solutions to integrate the X-Bank and Bank of America systems were considered.

Standard J2EE Web Application Global Foreign Currency

The first potential solution was to continue using the Global Foreign Currency system that was in place at the time of the case. The system was a J2EE Web application running on an IBM WebSphere Application Server. The system allowed authorized persons to log onto the system and request a foreign currency service. The system could be extended to X-Bank by providing user accounts to X-Bank employees and giving them access to Bank of America's system.

The advantages of continuing to use the original system were obvious. The system was mature, tested, and proven to work. Operationally, providing user accounts to X-Bank employees to access Bank of America's system could be done easily. Further, Bank of America would not need to spend time or money to develop a new system.

The primary disadvantage of this approach was scalability, i.e., the ability of a system to work well for a small number of users as well as for a large number of users. Although the IBM WebSphere Application Server was capable of handling a very large number of requests and therefore would scale to accommodate X-Bank, the process of creating and managing user accounts would not scale well because it required manual intervention by Bank of

America staff. Bank of America employees would have to work with the relevant branches of X-Bank over phone or e-mail to create accounts. Account problems, such as a forgotten or expired password, would require manual intervention by an employee of one or both banks. Multiply these problems by the 2,000 branches of X-Bank and the original system might not provide a high enough level of service to X-Bank.

Another disadvantage of this approach was that the Bank of America's system did not integrate with the X-Bank's computers seamlessly. A teller at X-Bank would need to both use the X-Bank's computer and log into Bank of America's computer. In other words, to process a foreign currency request, the X-Bank's teller would need to process the request both on X-Bank's computers and on Bank of America's computers because the X-Bank's computers need to have a record of the transaction. A typo of a single penny would cause problems because the two banks' records would not reconcile with each other at the end of the day. Turnover rate of tellers and training of tellers were other issues.

Modified Global Foreign Currency J2EE Web Application with Single Sign On

This solution would have involved modifying the Global Foreign Currency system to allow X-Bank's users to sign on to Global Foreign Currency system with X-Bank's credentials. This would eliminate the administrative costs associated with supporting tens of thousands of user IDs and passwords (as was the case for the first approach). User ID management and password reset were some of the most expensive support costs for an IT department.

The advantages of improving the existing system was that fewer people-hours would be spent managing accounts and resetting passwords. The improved system would build on the original system that already worked. An improvement project had less risk and would very likely succeed. The improved system would be written in J2EE, the same language used to write the original system, thus avoiding any compatibility issues between the two systems. Bank of America's programmers had extensive J2EE experience so development could proceed rapidly because they would not have to learn a new system.

Although the improved system would automate some processes, many functions would still need to be performed manually. The improved system would be more scalable than the original system, but possibly still not effective and efficient enough to handle 2,000 branches of X-Bank. Orders would still be handled separately in the two banks. The information flow would not be seamless and workflow efficiency was a concern. The improved system could not automatically keep the accounts of X-Bank and Bank of America perfectly reconciled, leaving open the possibility of mistakes that would require many hours of human intervention to fix.

Also, while it was technically possible to run Single Sign On between the two banks, there was a huge unknown as to whether it would be acceptable for Bank of America and X-Bank in terms of corporate policy, security, privacy, and risk. Allowing the X-Bank's users to sign on to Global Foreign Currency system with X-Bank's credentials was not only a security concern, but it might also have legal implications.

Service-Oriented Architecture via Web Services

This solution would have involved modifying the Global Foreign Currency system that would open up the Global Foreign Currency system business functions to X-Bank. This would allow X-Bank and Bank of America information systems to communicate in real time using request and respond patterns to merge two different operations into one integrated process. To speed up implementation, the solution would be based on well known architecture, industry standards, and best practices.

This solution would alleviate the challenges discussed. X-Bank users would no longer need to sign on to the Global Foreign Currency system. All they would have to do is use their X-Bank native application, which would communicate with the Bank of America's Foreign Items Systems office in the background. Information would flow between the two organizations in a timely and sequential manner. In fact, X-Bank users might not realize they were placing orders to another organization. X-Bank's native application could send orders to Bank of America's Foreign Items Systems office and debit their customer's payment at the same time, thus eliminating manual booking on two platforms.

The advantages of the new system were the potential to alleviate the problems and limitations discussed previously. The system would integrate with X-Bank's systems and allow for Single Sign On, thereby removing the need for Bank of America to be involved in password resets and other account administration. The information for each transaction would only be entered once and therefore X-Bank's and Bank of America's records would always line up. The system would be scalable to accommodate any number of X-Bank branches.

The disadvantage of the new system was its high risk. SOA was a relatively new technology and unforeseen problems during development would delay the project or result in higher development cost. The future of SOA was also uncertain at that time. Many new technologies did not last long. Also, Bank of America's developers had little or no experience with SOA. The new system could possibly not work well enough or soon enough to be acceptable to X-Bank, which would cost Bank of America money and ruin its reputation. Opening up the Global Foreign Currency system business functions to X-Bank was also a security risk and had legal implications.

The Final Decision

Raymond Yeung and Anna Watcher wondered what their colleagues would suggest during the meeting after lunch. Would they suggest the new and untested SOA? Or would they opt for the more established J2EE they had experience with?

Eventually, Bank of America went with the SOA solution, and they were successful. Exhibit 4 details the approach taken by the Bank of America's Foreign Items Systems office. Exhibit 5 includes a diagram illustrating how Bank of America's SOA solution works.

Was this a case of excellent planning and execution or, despite the successful outcome, an example of what not to do?

Discussion Questions

Characterize the Bank of America's Foreign Items System office competitive environment. What is the role of IT in Bank of America daily operation and competitive strategies? What is the role of IT in Bank of America's Foreign Items Systems office daily operation and competitive strategies?

What were the systems integration issues facing Bank of America's Foreign Items Systems office with regards to the Request for Proposal (RFP) from a major U.S. bank to provide foreign currency services? What were the possible solutions?

Compare the SOA and Global Foreign Currency system (J2EE Web application) approaches. How do they differ? How are they similar?

Do you think Bank of America made the correct decision in using SOA? What were the risks, if any, that the Bank of America's Foreign Items Systems office took by using a new technology instead of the more established technology?

If you were Raymond or Anna, would you have picked SOA or the more established Global Foreign Currency system (J2EE web application)? Would your answer change if you were asked to make the same decision in 2010? Is SOA still a new technology in 2010?

If you were in the top management of Bank of America or X-Bank, would you support the SOA proposal in 2004? How about in 2010? Why or why not?

Note: Instructors adopting the case can contact Professor Keng Siau for a copy of the Teaching Notes.

References

Bank of America. (2007). 2007 Annual Report. Retrieved November 4, 2008 from <http://www.bankofamerica.com/annualreport/2007/>

BankersAlmanac. (2008). Top 50 Banks in the World. Retrieved November 4, 2008 from <http://www.bankersalmanac.com/addcon/infobank/wldrak.aspx>

IBM. (2006). IBM Honors Its Most Innovative System i Clients. Retrieved November 4, 2008 from <http://www-03.ibm.com/press/us/en/pressrelease/19483.wss>

IBM. (2007). IBM Honors Most Innovative System i Clients in Third Annual IT Innovation Awards. Retrieved November 4, 2008 from http://findarticles.com/p/articles/mi_pwwi/is_200705/ai_n19046189

Further information on the topics discussed in this case is available from IBM. In addition, there are some well-written articles on Wikipedia.org that provide more detail on SOA, EDI and other topics.

IBM Power Systems

<http://www-03.ibm.com/systems/power/>

http://en.wikipedia.org/wiki/IBM_Power_Systems

IBM System i

<http://www-03.ibm.com/systems/i/>

http://en.wikipedia.org/wiki/IBM_System_i

IBM System p

<http://www-03.ibm.com/systems/p/>

http://en.wikipedia.org/wiki/IBM_System_p

IBM AIX and Linux

AIX: <http://www-03.ibm.com/systems/power/software/aix/index.html>

Linux: <http://www-03.ibm.com/linux/>

Bank of America

Home page: <http://www.bankofamerica.com/>

Foreign currency services: <http://globalforeigncurrency.com>

Other Topics

UNIX: <http://en.wikipedia.org/wiki/Unix>

Linux: <http://en.wikipedia.org/wiki/Linux>

SOA: http://en.wikipedia.org/wiki/Service-Oriented_Architecture

EDI: http://en.wikipedia.org/wiki/Electronic_Data_Interchange

J2EE: <http://en.wikipedia.org/wiki/J2ee>

Exhibit 1 Use of IBM Computers at Bank of America's Foreign Items Systems Office

The Bank of America's Foreign Items Systems office has a long history of using IBM computers to solve problems. The following list of milestones was submitted by Bank of America's Foreign Items Systems office as part of their 2006 IBM Innovation Award nomination form.

- 1986 – Major productivity gains were achieved when manual processing was automated using IBM S/36 with in-house-developed applications.
- 1990 – An upgrade to AS/400 with all applications running in S/36 mode and the establishment of a hot, contingency, data center using MIMIX for remote data mirroring (one of the first MIMIX implementations in Canada) reduced operational risk, permitted application growth, and increased automation.
- 1991 – Conversion of all S/36-mode applications to AS/400 native applications, reduced operational risk, modernized applications, and laid the foundation for component-based architecture.
- 1992 – Development of Asynchronous Processing Model applications using Data Queues improved automation and was a first step into SOA – Service Oriented Architecture.
- 1993 – Development of a MICR printing application using ACOM printing technology produced MICR-encoded foreign drafts, reduced clearing costs, and allowing compliance with international regulations.
- 1994 – Development of a CRM application based on message queuing facilitated the coordination of sales, marketing, and customer service support.
- 1994 – Development of a bulk foreign-exchange payments solution enabled a dramatic increase in transactions processed, reduced the cost of transaction processing, and generated a new revenue stream.
- 1995 – An upgrade to a 64bit, RISC AS/400 model 500 (one of the first RISC Production Implementations in Canada) enabled further growth.
- 1996 – 1999 Account reconciliation was running on personal computers and unable to cope with the increased number of transactions. Development of a multi-currency / multi-account reconciliation system, which utilized the superior indexing capabilities of the DB/2 database, provided a fast, fully-automated solution which conformed to the 'all systems and services on one server' concept.
- 1996 – 1999 Introduction of new services, FAX/400, SWIFT, etc., to enhance user experience.
- 1997 – Upgraded to a S20 server and continued implementation of the Asynchronous Processing Model improved system resource utilization and response time for interactive jobs.
- 1998 – First JAVA client server application implemented to serve foreign exchange rates to clients. This implementation used IBM JAVA-Toolbox/400.
- 2000 – Development of an in-house intranet application, using WebSphere version 2.x, provided a common repository for company news, forms, and documents which reduced costs and improved procedure distribution.
- 2000 – An upgrade to iSeries 830, with all applications, accommodated increased business demands.
- 2001 – Implementation of www.globalforeigncurrency.com, a Global Foreign Currency (GFC) application using WebSphere 3.0, focused on transaction security (client digital certificate authentication using iSeries Digital Certificate Manager, IBM 4758 crypto processors), ease of use, performance, and reliability. The Web application is fully integrated with systems already deployed on the AS/400.
- 2001 – Creation of an EDI data parser for a new, high-profile client provided a cost-effective solution when there was no volume justification for a full, EDI system purchase.
- 2002 – An upgrade of the Global Foreign Currency application to WebSphere 3.5 and expansion of the Global Foreign Currency system offering to clients with intermittent internet connection accommodated Cruise Lines and airports.

- 2003 – Development of an application which utilizes an IBM IP2000 in combination with advanced AFP printing (adopted IBM mainframe AFP printing technology to iSeries), enabled Bank of America Canada to print, insert and manage 100,000+ duplex documents, with various types of encoding (MICR/OMR/OCR), within a 72-hour timeframe. The ability to adopt AFP technology on the iSeries helped in business gain.
- 2003 – Implementation of Connect:Direct (Sterling software) for iSeries enabled Bank of America Canada to support critical, bidirectional, secure data transmission (i.e., up to 5G of transaction data in one transmission) with client mainframes.
- 2003 – Global Foreign Currency system was upgraded to WAS 5.0.
- 2004 – An upgrade to i5 model 550 using LPAR to consolidate servers, simplified IT infrastructures, and reduced total cost of ownership. An MQ application was developed to support Bank of America Retail Banking network.
- 2005 – Development of a Global Foreign Currency Web Services application expanded Global Foreign Currency system's functionality and allowed business to support new client segments and continue to offer innovative on-demand services.

Exhibit 2 J2EE

Computers have always required some sort of programming languages. The very first computers had hard-wired connections soldered from one vacuum tube to another. Computers could be programmed by physically flipping switches on their front panels. Next, assembly language allowed the development of what we call programs today, but assembly language was very low-level and tedious. This motivated the development of higher-level languages, such as BASIC and Pascal, which used commands closer to English.

Early higher-level languages were procedure-oriented, which means that programs were written in a very linear fashion with calculations being performed sequentially. Procedure-oriented languages are ideal for mathematical computations and many other types of programming, but they are not adequate for all situations.

Object-oriented languages involve creating things called “objects.” An object has well-defined properties and interacts with other objects through clearly defined methods. Consider, for example, a video game with many complex characters. In an object-oriented language, each character is an object. The character’s features, abilities, clothes, and possessions are all stored as part of the character’s object. Two characters interact by exchanging objects, such as damage from being hit with a weapon, improved health by receiving a healing spell, or transferring ownership of an inventory item by moving it from one character’s object to another. Thinking of characters as objects makes sense both from an intuitive point of view and a programming point of view. It is possible to write such a game using procedure-oriented languages, but it is much more difficult. Any system in which the “things” being acted upon can be thought of as objects from a human point of view can probably be thought of as objects from a programming point of view.

Some of the earliest object-oriented languages were modifications or extensions of procedure-oriented languages. For example, object-oriented C++ is an enhanced version of the procedure-oriented C programming language. Although C++ provides some support for programming using objects, it is not a fully object-oriented language and it is possible to write programs in C++ that have no objects at all.

Java was developed as a fully object-oriented language. Everything in Java, with the exception of simple data structures, is an object. Java was designed to allow the development of independent objects that could easily be designed for one program, but then copied and used in future programs. This allows programmers to develop their own library of objects that can be used over and over again, without having to re-write everything from scratch every time a new project is initiated.

J2EE takes the concept of object libraries one step further. J2EE is not only just the Java language, but also a very large set of objects and other tools that allow for rapid development of large programs much more quickly than can be achieved using other languages.

J2EE provides Application Programming Interfaces (APIs) that allow programmers to tap into the many features provided by the J2EE platform. J2EE makes it relatively easy to interact with a database, read and write XML messages, and allows programs to communicate with each other via the Internet. J2EE can even manage the functions of certain types of servers, coordinate the various programs running on it, and provide security and other services to the programs.

If a project has needs that closely match the services, objects, and APIs provided by J2EE, then it can usually be completed very quickly by piecing together the relevant pieces of J2EE into a single functioning program or system. On the other hand, if a project requires custom programming, then Java code can be written by hand and integrated with other parts of J2EE. Note that projects that require features not provided by J2EE or that have other problems such as requiring extensive customization might be difficult to achieve with J2EE. J2EE is not the solution to every programming problem.

A detailed description of how J2EE could help with the problem presented in this case is not provided because the many services provided by J2EE and the high degree of customization possible with them allow for many, many ways to implement a banking transaction system.

Exhibit 3 Service-Oriented Architecture (SOA)

Service-Oriented Architecture (SOA) consists of services performing different functions that work together by exchanging messages. There are different versions of SOA, the key difference between the versions being where the services are located and how they communicate with each other. In this case, the services are located on different computers and communicate over the Internet. (It is also possible to have all of the services existing on a single computer, with different programs providing the different services.)

The following outlines a simple SOA transaction, in which two computers exchange information over the Internet. It begins with a computer that has data but cannot process the data. For example, the computer cannot process tax-related data because it does not have access to the current tax laws, or the computer cannot process inventory-related data because it does not have access to the inventory database. In order to solve this problem, the computer finds another computer that knows how to process the data. In this exhibit, we assume that the two computers already know about each other.

Before sending the data to the second computer, the first computer must place the data into a pre-arranged format so that the second computer can understand the data. The data is first placed in an XML (eXtensible Markup Language) format. XML helps computers understand data by placing descriptive tags around the data. For example, a book on SOA might have the title, "Understanding SOA." A computer might not recognize those words as the title of a book. XML places tags around the words that describe what the words mean. XML would convert the title into, "<book_title>Understanding SOA</book_title>." The opening and closing tags form a pair that describes the enclosed words and therefore allows computers to understand the purpose of the words. XML allows any data to be wrapped in user-created, computer-readable tags. Because XML documents are text-only, they are readable by any computer system, regardless of hardware or operating system. By using XML as a basis for exchanging data, SOA can draw on the many programs and tools that already know how to generate and process XML data.

After the data has been converted into a computer-readable XML format, it must then be converted into a format suitable for transmission over the Internet. Although XML documents can be transmitted by e-mail or FTP, a method called Simple Object Access Protocol (SOAP) has been designed specifically for exchanging XML messages between services running on the Internet.

The SOAP message is then sent over the Internet, usually using the same HTTP protocol used for transmitting Web pages.

The second computer receives the SOAP message over the Internet and recognizes that it is a request for data processing. The computer removes the SOAP wrapper to access the XML message, examines the tags that describe the data, and extracts the data from the XML message. The computer can then process the data and get a result. The information obtained by processing the data is then sent to the first computer, usually after placing the results into XML format and a SOAP message. The first computer can then take the results and use them to continue processing the data.

Exhibit 4 Approach Taken by Bank of America's Foreign Items Systems Office

After making the initial educated decision, the Foreign Items Systems office's team began the process of delivering the solution:

- a. The first step was to convince X-Bank's IT team that SOA was the right direction. The solution needed to have two collaborating and willing partners to achieve the desired result.
- b. Second, they had to design and execute a Proof of Concept exercise to confirm SOA was a workable solution.
- c. Then, they needed to obtain top management approval and support for the innovative approach. Nothing could proceed without top management buy-in.
- d. Next, they had to work with all levels of stake holders (Information Security, Enterprise Architecture, Compliant, Legal, Operations, Project Management, etc.) to lay out the road map and resolve all information road blocks.
- e. The next step was to design the specifications and build the solution.
- f. Then, the system had to be tested.
- g. Finally, they performed implementation, monitoring, and setting-up support.

The Bank of America's Foreign Items Systems office implementation of SOA ran on IBM Power Systems. The computers in Bank of America's Foreign Items Systems office would receive orders of foreign currency from the X-Bank's computer system via the Internet using secured communication protocol. The information was structured in XML format and wrapped in a SOAP message for transportation.

When the Toronto computer receives one of the pre-established requests, the request is parsed according to established rules and processed. The result of this processing is returned in a pre-established format to the requester. The response is sent back to X-Bank in real time, resulting in a request-respond operation that links the two disparate systems together to fulfill the business need.

The entire system was developed in less than six months, even though the Bank of America's Foreign Items Systems office had no prior experience in developing applications that used Web Services Description Language (WSDL), SOAP or Web services over the Internet. The team was able to develop a cutting-edge system in such a short period of time because they strictly adhered to best practices when the Global Foreign Currency system was developed many years ago.

One such standard is the use of the three-tier architecture (similar to the Model-View-Controller design pattern), which divides IT systems into three parts. Such a division allows one part to be modified without affecting the other parts. The parts are the data storage system, the business logic system, and the user interface system. Data storage and user interface are self-explanatory. The business logic system is the programming that achieves what the system is designed to accomplish. In a well-designed three-tier system, it would be possible to change any one tier with little or no changes to the other tiers. For example, in theory, a new database system could replace the old system without any changes being necessary to the business logic or user interface.

The system entered service in early 2006 and less than one year later had processed more than 172,000 requests. After the system was extended to all of the X-Bank branches, it was processing more than 5,000 messages and conducting about 1,000 transactions per day. Bank of America's Return on Investment (ROI) was over 385% and they estimated that their solution had an operating cost of about 32% less than any other non-SOA solution could have achieved. The entire system was implemented using hardware already owned by Bank of America's Foreign Items Systems office and did not require the purchasing of any commercial off-the-shelf software. Because the system was hosted on IBM Power Systems, it was capable of processing information in real time, which eliminated the need for an additional system to support the business.

Exhibit 5 Bank of America's SOA Solution

The diagram below, provided by Bank of America's Foreign Items Systems office, illustrates how their solution functions.

Two computers exchange SOAP messages. The SOAP messages contain XML data. Each system is capable of acting on the data in certain ways. By passing data back and forth, each computer is able to perform its business functions and together they are able to complete the foreign currency banking transaction.

Bank of America's Foreign Items Systems office (the Publisher) solution to the problem of providing foreign currency services is to publish their Services using WSDL and to make it available to (the Consumer) X-Bank. When a branch in X-Bank needs foreign currency, the user signs on to the X-Bank's native application to order foreign banknotes, and X-Bank's computer communicates with Bank of America's Foreign Items Systems office to perform the business process (e.g., to request an exchange rate). X-Bank's computer system puts the request in XML and then in SOAP format according to the rules specified in the WSDL, and sends the message to Bank of America's Foreign Items Systems office. The Bank of America's Foreign Items Systems office computer unwraps the request and either fulfills the request or produces an explanation if the request cannot be fulfilled. The result is put into XML and then SOAP formats and sent back to the X-Bank computer in real time.

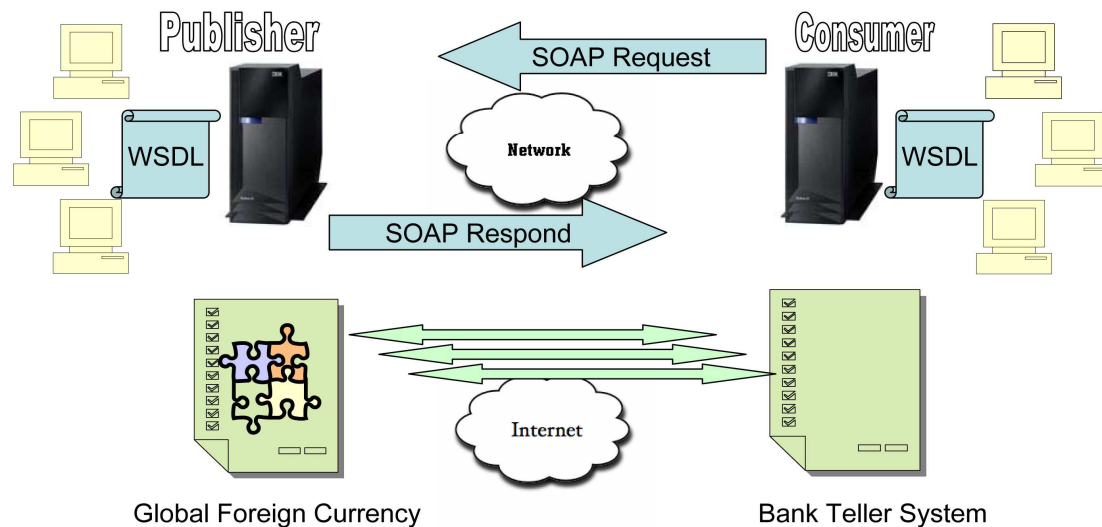


Figure 5-1. Bank of America's Solution to the Problem