Merrill Lynch Connects Past and Future Technology
CASE STUDY

Merrill Lynch is a worldwide leader in financial management and advisory services, employing 50,600 workers in 36 countries and territories. The company and its subsidiaries provide brokerage, investment banking, financing, wealth management, advisory, asset management, insurance, lending, and other related products and services to private, institutional, and government clients with assets of $1.6 trillion. In 2005, Merrill Lynch posted a record $5.1 billion in net earnings, a 15 percent increase over the previous year, on net revenues of $26 billion.

One of the most critical components of Merrill Lynch's operations is its information technology infrastructure. Over the last five years, that IT infrastructure has played a major role in the company's gains. Like many financial institutions, Merrill Lynch has had to modernize its technology infrastructure in order to remain competitive.

Merrill Lynch considered its IBM mainframe installation, which was one of the largest in the world, to be a strategic asset. The mainframe ran in the neighborhood of 23,000 programs to process the firm's 80 million daily online transactions for accessing customer accounts online or making stock trades.

In modernizing its technology, Merrill Lynch had to make choices regarding its legacy computers and applications. Internet-based applications that gave customers access to their portfolios and tools to work with them were a key to remaining competitive. But these applications did not use mainframe-based software. How could Merrill Lynch develop such applications while leveraging the processing power and wealth of data in its mainframe?

The answer appeared to be Web services and a service-oriented architecture (SOA). Most corporations developing a SOA typically use commercially available platforms such as those from BEA Systems and webMethods instead of creating their own development platforms. They rely on the vendor's expertise and access to consultants familiar with integrating mainframe and Web applications.

Project leader Jim Crew, then head of database infrastructure for Merrill Lynch, determined that on the surface, purchasing an SOA platform was much easier than building one, and would have enabled the firm to deploy its Web services relatively quickly. However, no SOA vendors that Crew researched offered products that met Crew's requirements for the project. They were offering SOA platforms that were geared toward distributed programming and recent development tools such as Java and .NET.

Merrill Lynch's 1200 mainframe programmers did not have experience with these tools. Retraining this huge staff did not make sense economically, nor did purchasing new workstations required for running the development software. According to research from Gartner Group consultants, retraining Merrill Lynch's mainframe programmers could have taken as much as a year and cost more than $80 million. To Crew, it was obvious that the firm should pursue a more unconventional approach; construct a proprietary Web development platform from the ground up to extend the capabilities of its legacy mainframe systems.

Merrill Lynch had initially tried to avoid these costs by copying the data stored in its mainframe installation into Oracle, Sybase, or Microsoft SQL Server databases. In those formats, the data were compatible with server-based applications. However, that technique was not entirely satisfactory. Copying large quantities of data often introduces errors based on disk failures and space issues. Furthermore, some data can become obsolete as soon as they are copied. For instance, a client who made several stock trades would have to wait until the next day to see an accurate balance in his or her account. Crew noted that the firm was spending money on copying data that could quickly be out-of-date while the accurate data were always residing on the mainframe.

Instead, Merrill Lynch created its own set of in-house proprietary tools that enable its mainframe legacy programs and the functions they perform to be exposed as Web services. XML tags are used to describe the data for other applications that are equipped to interpret XML. SOAP makes it possible for programs running under different operating systems to communicate with each other. Together, the two standards made it possible for online applications to communicate effectively with the mainframe without an additional layer of middleware.

Merrill Lynch's Web services toolset was called X4ML, which stood for XML for Modernizing Legacy

Crew challenged his team to increase the firm's savings from Web services ten-fold to $20 million. Crew's team established five criteria for the Web services project:

1. No new programming languages for the mainframe programmers to learn.
2. No new software tools for development that would require expensive workstations; tools would be accessible from a Web browser.
3. A central storage directory for the Web services that would be developed so that programmers could easily reuse and repackage them with each other.
4. Web services developed as a result of the project had to conform to the existing mainframe security standards as well as Web security standards for encryption, authentication, and authorization.
5. Inclusion of budding Web services standards in the Web services architecture to ensure future viability.

The project team prohibited the new platform from requiring changes to program code on the mainframe or hindering its operation in any respect. The team did not want to alter the mainframe in any way because of its track record, its complexity, and the fact that there was likely no one on staff who knew the inner workings of its deep-rooted code.

To maximize simplicity and speed, the team did not install a middleware server to translate requests made to it in other languages, such as Java, into instructions that could be understood by the mainframe applications. Instead, the translation software was written in Assembly Language (a programming language dating to the 1950s that is rarely used today for business applications) and installed directly on the mainframe This strategy reduced the number of things that could go wrong during translations and promised better performance.

The lack of middleware meant that the system's users, such as Merrill Lynch financial advisers, could request information directly from the mainframe from their desktops. For example, an adviser could use a Web browser to request a list of all clients who owned shares of a certain stock, such as General Electric (GE). The request arrives at the mainframe using SOAP, which instructs the mainframe to perform a particular operation, and the search is translated by XML.

A Merrill Lynch mainframe programmer can access the X4ML development tool from a desktop Web browser. Using X4ML, the programmer can create and name a new Web service, import the necessary application from the mainframe, and then pick and choose which parts of the operation in the legacy application to include in the Web service. Thus, a programmer is able to produce a Web service that pulls out all of the personal data for a client, or only the less sensitive data, such as name and address. Once a programmer creates a Web service, it is listed in a Universal Description, Discovery, and Integration (UDDI) directory, where it can be accessed by other programmers. The X4ML development tool also includes a testing capability, which enables programmers to correct errors before deploying a service, as well as utilize trial-and-error to perfect combinations of applications for new services.

Merrill Lynch earmarked $1 billion over a three-year period to use X4ML to provide its 14,000 financial advisers with a new suite of wealth management applications. For this initiative, the firm teamed with Thomson Financial and Siebel Systems (now owned by Oracle), which offered financial data and research services and client management expertise, respectively.

Merrill Lynch's investment in Web services saved the company $41 million in application development costs. The company wrung even more value out of X4ML by selling it in December 2005 to Web services vendor SOA Software Inc. of Los Angeles. As part of the deal, Crew and three other key members of the X4ML team shifted their employment to SOA Software to continue enhancing the tool, which was renamed Service Oriented Legacy Architecture (SOLA). Merrill Lynch had a long history of selling internally developed technology, and it viewed the sale of X4ML as a way of optimizing its investment.

Chief Technology Architect Andrew Brown did not think that turning the technology over to another company would hurt his firm's competitive advantage. He needed six months to convince management that selling to a software vendor was the right move. After the fact, management appreciated the value of the sale and the space that it created in the IT budget. At the time of the sale, X4ML was utilizing 600 Web services for 40 different core applications at Merrill Lynch and processing 1.5 million transactions daily. The price of the X4ML sale to SOA was not disclosed, but SOA Software began selling SOLA to customers in 2006 for $125,000. Purchasers of the tool were poised to gain unmatched scalability. Meanwhile, the success of X4ML gave a second life to Merrill Lynch's mainframe programmers and their work.

Sources: Mel Duvall, "Merrill Lynch & Co.: Web Services, Millions of Transactions, All Good," *Baseline Magazine*, February 2006; Charles Babcock, "Merrill Lynch Sells Its Web Services Vendor a Web Services Tool," *InformationWeek*, December 6, 2005; Elena Malykhina, "Merrill Lynch Embraces SOA," *InformationWeek*, November 8, 2005; China Martens, "SOA Software Snaps Up Blue Titan," IDG news service through www.techworld.com, May 8, 2006; Ann Bednarz, "SOA Software Buys Blue Titan," NetworkWorld, May 8, 2006.

CASE STUDY QUESTIONS

1. Why did Merrill Lynch need to update its IT infrastructure?
2. What is the relationship of information technology to Merrill Lynch's business strategy? How was its Web services initiative related to that strategy?
3. Evaluate Merrill Lynch's approach to Web services development. What are the advantages and disadvantages? Is it a good solution? Explain your answer.
4. Do you think that Merrill Lynch's decision to sell off its successful technology initiatives was a good idea? Why or why not?