



COBOL TECHNOLOGY AND CONTEMPORARY BUSINESS SYSTEMS

By
Paul Halpern, PhD
Director of Marketing
Micro Focus
701 East Middlefield Road
Mountain View, CA 94043

And

Wayne Rippin
WMR Consulting
5172 Tehachapi Way
Antioch, CA 94509

Business Imperatives

Among business executives there is broad agreement that effective implementation of technology is a significant factor in meeting new business challenges. This is born out by the essential role that IT plays in enabling business initiatives. There are several ongoing business challenges requiring near term resolution that are common to enterprises wanting to expand and grow their markets:

- Reducing internal business processes and costs
- Updating customer response management systems
- Improving the staff Knowledge Management system
- Attracting, training and leveraging staff with key technical skills

Legacy systems that currently service these business functions may need to be updated in order to increase their effectiveness and reduce their total cost of operation. The tasks before management are:

- Successfully leveraging these legacy systems by improving their execution efficiency through modernized development and maintenance processes
- Re-hosting these systems to non-proprietary platforms as suppliers move away from closed systems
- Transforming and integrating existing legacy systems across non-homogeneous platforms and making them Web ready, using evolving technologies such as Web Services.

Existing legacy systems are predominantly written in COBOL. It is estimated that there are 2000 billion lines of COBOL code growing at a rate of 5 billion lines per year over the next four years. Businesses have approximately five trillion dollars invested in these systems. There are approximately 90,000 COBOL programmers maintaining and writing over 9.5 million COBOL applications, representing a large proportion of the computer code in the world. (Gartner Inc., *From the Dustbin, Cobol Rises* - eWeek, May 28, 2001)

COBOL and Contemporary Systems

COBOL was first introduced in the 1950s for accessing and manipulating real world, data-centric business applications. The necessity to rapidly and securely retrieve from and update to databases led to the development of transaction monitors such as CICS and IMS. These, in combination with COBOL, exploded the use of commercial databases on mainframes in the banking and financial sectors.

In more recent years, COBOL has tended to be eclipsed by newer languages, such as C++ and Java. However, developers in these languages have faced the same problems faced (and solved) by COBOL programmers when developing transactional systems. New transactional frameworks have evolved based on open systems, such as Microsoft's COM+ and Enterprise JavaBeans (EJB). These provide interconnection, security and transaction-based services for objects created in multiple languages. Recent enhancements to COBOL have enabled it

to interoperate with these new languages. This technology enables a transformed and integrated COBOL legacy application to be part of contemporary system architectures.

Concurrently, data centric research moved forward with the development and introduction of XML. Fundamentally, XML is a set of rules and guidelines for describing structured data in a plain text format.

Using XML as the delivery vehicle, Web Services are being touted as the technology to elevate the Web to its next evolutionary level. Web Services, built on an XML base, promise to permit programmers to create application components that will be able to communicate with other components, regardless of implementation language or deployment platform. Until now, although it has been possible to build applications out of software components, they have often been restricted to running on one particular operating system or platform. Interoperability between the different component architectures has been difficult. On the Microsoft platforms, the dominant software component architecture has been the Microsoft Component Object Model (COM). On non-Microsoft platforms, the dominant component architecture used has been Common Object Request Broker Architecture (CORBA). More recently, there has been a focus on the Java-based Enterprise JavaBeans (EJB) model. Making these different models communicate with each other has often been a difficult and frustrating task. Web Services evolve the object-oriented vision of assembling software from component building blocks to the assembly of services that may or may not be built on object technology. (Rippin, 2001)

The Future of COBOL

Micro Focus has defined three pillars supporting a technology bridge that enterprises must cross to expand their markets, increase their profitability and reduce total cost of operation:

- COBOL Application Re-hosting and Mainframe Renewal
- Legacy Transformation and Integration
- Java and Web Services

Micro Focus provides customers with a complete infrastructure solution that is unique in the industry. Connecting these pillars is a roadway built upon Micro Focus tools and solutions. These contain a comprehensive, scalable suite of innovative COBOL development environments enabling businesses to address new competitive challenges by:

- Allowing building of scalable business systems
- Providing for more efficient processing of legacy applications
- Enabling legacy applications to be re-hosted to most hardware platforms
- Maintaining security levels across all ported platforms
- Permitting connection of data and applications that exist on different platforms

- Keeping cost down and productivity up by limiting the reinventing or converting of application code

COBOL Mainframe Renewal

The Micro Focus approach is to develop, with the customer's input, a well documented and unsurpassed plan for renewing mainframe business systems. The plan evaluates the current state of a customer's business system. Based on this analysis, strategy objectives are defined. The plan documents these objectives and incorporates them in a phased-in process to balance progressive change with business needs. The risk is thereby minimized. The plan has the following basic steps:

- Assess current system
- Develop a vision and strategy
- Create a proof of concept model
- Build, deploy and maintain the system
- Provide for continuous review and innovation of the plan

Implementation of the Micro Focus methodology is achieved with tools such as Micro Focus Mainframe Express®. It provides a fully integrated, IBM-compatible mainframe maintenance and development environment that runs on a PC on a programmer's desktop. Programmers can take advantage of modern, powerful development tools to edit, compile, debug and test mainframe applications on their workstations, all combined in a state-of-the-art integrated development environment. Mainframe Express combines leading-edge mainframe emulation, a comprehensive unit and integration test environment and unparalleled connectivity. Productivity across the entire development lifecycle is increased. A process-driven change management system tracks changes as the development cycle is completed.

Once the development process is complete, the deployment issues are handled by the solution component containing complete replication of OS/390 development on the desktop. This includes support for all major databases and transaction systems.

COBOL Application Re-hosting

This pillar provides customers the ability to re-platform, or re-structure COBOL applications on their platform of choice: UNIX or Windows. Micro Focus is able to offer this solution because it is the recognized industry leader by hardware manufacturers and brings to their platform the advantages of Micro Focus COBOL, the recognized industry standard.

The re-hosting task can be made significantly easier by the use of Micro Focus Revolve®. This tool helps developers understand, document and improve COBOL applications. Developers can work visually, improving time-to-market by accelerating understanding to a fraction of the time it would take using outdated technology or manual processes.

Micro Focus Net Express® and Micro Focus Server Express™ are used for developing and deploying the re-hosted application on to high-performance Windows or UNIX servers.

Micro Focus Net Express is a complete COBOL development environment for Windows platforms that offers an open, scalable alternative to running COBOL systems on the mainframe. In addition, for those people targeting UNIX platforms, Net Express can be used to quickly develop the applications in an intuitive, visual environment; then Server Express can be used to compile and deploy the application on the UNIX platform of choice.

This combination of tools allows applications to be developed easily then rapidly deployed. Remote debugging and production debugging enhancements enable advanced e-business and web applications to be debugged with ease, directly on the deployment server, thus making it easier to develop and deploy robust, high performing web applications.

Java and Web Services

Businesses are constantly seeking to improve access to their products and service offerings. Today, that translates into providing Web access to customers as both a sales and informational channel. A company developing an e-commerce site could use a string of Web Services to tie together the processing of a credit card, checking product inventory and determining the best possible shipper linking together the data each service requires, regardless of which back-end systems they are running.

It should be noted that just because these are called 'web' services, it does not mean that they can only be used in traditional web applications that use a web browser to communicate with the user. A Web Service can just as easily be used from an application that interacts with the user via a Windows graphical user interface, or even a character user interface. They are called Web Services because standard Internet protocols are being used for the communication between applications.

Web Service applications can vary in scope from simple requests (for example, the retrieval of a stock price) to complex business systems that access and combine information from multiple sources.

Web Services provide the following benefits:

- Organizations can make their services available, both internally and externally, to a wider range of platforms and clients, both increasing revenue and delivering more value to clients. In particular, Web Services provide a potential solution to the business-to-business (B2B) integration problems that have existed for years where different systems have provided incompatible, non-standard interfaces for accessing their functionality.
- Organizations can access a large number of Web Services provided by third parties to reduce their own development time and deliver more powerful applications.

- Businesses can develop applications using their favorite programming languages (COBOL, Java, C++, etc) and architectures (J2EE, Microsoft .NET, etc) without concern about which architecture their clients will use or the need to impose a particular architecture on their clients. This means that better use can be made of existing development staff, and the development costs incurred by having to develop multiple interfaces into systems can be reduced.

A recent paper by Rippin (2001) demonstrated that exposing COBOL applications as a Web Service is currently, at best, a complicated two-stage process which uses the functionality of Net Express and requires excessive use of third party software.

Micro Focus is working vigorously to make Web Services easy to use and create using Micro Focus tools. The objective is to enable existing COBOL applications to be exposed as Web Services. The technology will be complementary to Application Servers such as IBM WebSphere or BEA WebLogic. This will also enable organizations to take advantage of Web Services without the need to learn about other technologies such as Java or COM.

Micro Focus plans to implement COBOL as a first class Web Services language. This would enable applications to do the following:

- Invoke Web Services implemented in any language from COBOL using conventional syntax such as the CALL statement.
- Define a new COBOL subprogram (or use an existing COBOL subprogram) with any COBOL data definitions in its Linkage Section and simply publish this as a Web Service.
- Run the COBOL program as a Web Service on whatever platform it executes on today, such as Windows, UNIX, Linux or a mainframe under a TP monitor such as CICS.

The COBOL Web Service platform would provide HTTP protocol handling, security, SOAP, XML and a highly scalable, stateless server based on the current COBOL language and platform. All this support would be implemented with the high degree of performance and robustness that you have come to expect from COBOL.

Conclusion

Businesses that face IT challenges to their expansion and growth more than likely have a reservoir of COBOL legacy applications to upgrade. It is well recognized that a complete rewrite of these mission-critical applications is not financially or technically reasonable. Therefore, successfully leveraging these applications by improving their execution efficiency, or re-hosting them for quicker and more secure implementation and deployment, or exposing many of their core business functions as Web Services, thereby making them Web ready, is now the preferred approach.

The development of object language constructs for COBOL and access for COBOL to infrastructures such as COM+ and EJB provide interconnections enabling the transformation and integration of COBOL legacy applications with contemporary systems. Micro Focus offers a complete spectrum of tools and solutions to achieve these goals. Micro Focus plans to implement COBOL as a first class Web Services language. This, in theory, would permit 75% of the world's business applications that are written in COBOL to have access to the Web. Furthermore, it would enable a COBOL legacy application to invoke Web Services implemented in any language. It would allow any new or existing COBOL subprogram to be published as a Web Service and run any COBOL legacy application as a Web Service on whatever platform it executes on today, such as Windows, UNIX or a mainframe.

Rippin Wayne, 2001 *Web Services and Micro Focus COBOL* See www.microfocus.com White Papers