

Case Studies of System Architectures That Use COBOL Assets

● Haruhiro Okishima

(Manuscript received March 27, 2006)

The Common Business Oriented Language (COBOL) was developed as a programming language for business transactions. For a long time, it has been a major programming language for mission-critical systems because of its high reliability and compatibility ensured through international standards. NetCOBOL is a COBOL development environment offered by Fujitsu that supports quick construction and stable operation of mission-critical systems in cooperation with Java and Visual Basic. It achieves this by using existing COBOL assets as well as many supporting tools and standard development styles based on technologies that comprise Fujitsu's Systems Development Architecture & Support facilities (SDAS). In this paper, we introduce case studies of the architectures of three systems that use COBOL assets. The first system, at Seikodo Co., Ltd., is based on a combination of Java and COBOL. The second, at Meitetsucom Co., Ltd., is a .NET Framework-based system linked with a combination of Visual Basic and COBOL. The third, at Presidential Life, is a .NET Framework-based system built using COBOL.

1. Introduction

The Common Business Oriented Language (COBOL) guarantees high reliability and compatibility through the COBOL85 and COBOL2002 international standards. COBOL offers the ideal language specifications for business processing, with advantages such as mass data processing, document processing, and reporting, as well as high speed and high performance.

Fujitsu develops and supplies the following software products: COBOL85 for mainframes; COBOL-G for business computers; and NetCOBOL for Solaris, Linux, Windows, and .NET Framework (**Figure 1**). These products are evolving and maintain compatibility with each other except for OS-specific functions. They enable business logic created as much as 10 or 20 years ago to be executed in the latest environments. By replacing mainframes and business computers with the latest models, migrating to open systems, upgrading old Windows systems, and expanding

from Solaris to Linux, we are transferring the high compatibility of COBOL to various system configurations within a short period of time.

It has also become possible to use COBOL to build systems that connect with mainframes and business computers. For example, COBOL applications can now be connected with mainframes and open systems, and COBOL applications can use READ/WRITE statements on open systems to access mainframe databases.

In addition, with NetCOBOL it is now possible to build systems that cooperate with COBOL and Java using Interstage, which is an application framework that supports object orientation and is based on Fujitsu's J2EE.

NetCOBOL is the world's first COBOL development environment to support .NET Framework, which is a base framework for Microsoft's .NET environment. It can cooperate with various languages such as Visual Basic (VB) and C# and therefore enables construction of

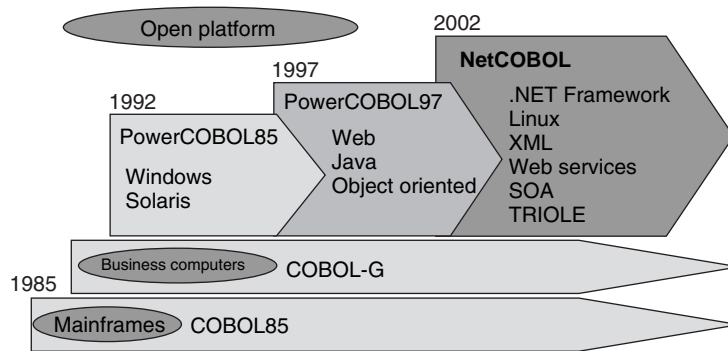


Figure 1
Evolving NetCOBOL.

systems using Web forms, Web services, and Windows forms.

This cooperation enables the construction of an optimal system that uses the most appropriate languages for the system configuration and existing assets. Languages such as Java and VB can be used for front systems (including GUIs), and COBOL can be used for back systems such as business logic (Figure 2).

The style of COBOL development is also evolving. NetCOBOL provides many full-featured development environments for the whole development cycle, from designing and programming to testing and maintenance, based on SDAS's high productivity, quality, and maintainability.

NetCOBOL provides far greater productivity and maintainability through features such as testing support tools and document production support tools. These tools have features from the SIMPLIA series of SDAS tools built in, including specialized editors, builders, debuggers, and screen form editors.¹⁾

In this paper, we present case studies of the architectures of three systems that use COBOL with advanced technologies. The first system was built through cooperation with Interstage and Java, and the other two were built with .NET Framework.

2. Case study of system built in cooperation with Interstage and Java in Japan

In this section, we look at the system of Seikodo Corporation (hereafter called Seikodo), in which COBOL, Interstage, and Java are cooperating.

Seikodo is a wholesaler of audio and video related products such as CDs, DVDs, and games. They have created ordering and inventory control systems, store merchandising support/information provision systems, and strategic information systems that are unique in the industry.

The existing mission-critical ordering and inventory control system had been in service for 20 years and was updated by expanding the base of a mainframe that was built in 1984. As a result, the system had become complicated and rigid, and the company needed a system that could handle the rapid changes in the company's business environment (e.g., the changeover of sales channels from private stores to corporations and the start of Internet sales), as well as changes in their product lineup. The company therefore decided to build a new mission-critical system based on an open system.²⁾

The goals and requirements for the new system were as follows:

- 1) Speedy adaptation to the business environment and stable operation
Seikodo provides 24/7 service at all of its

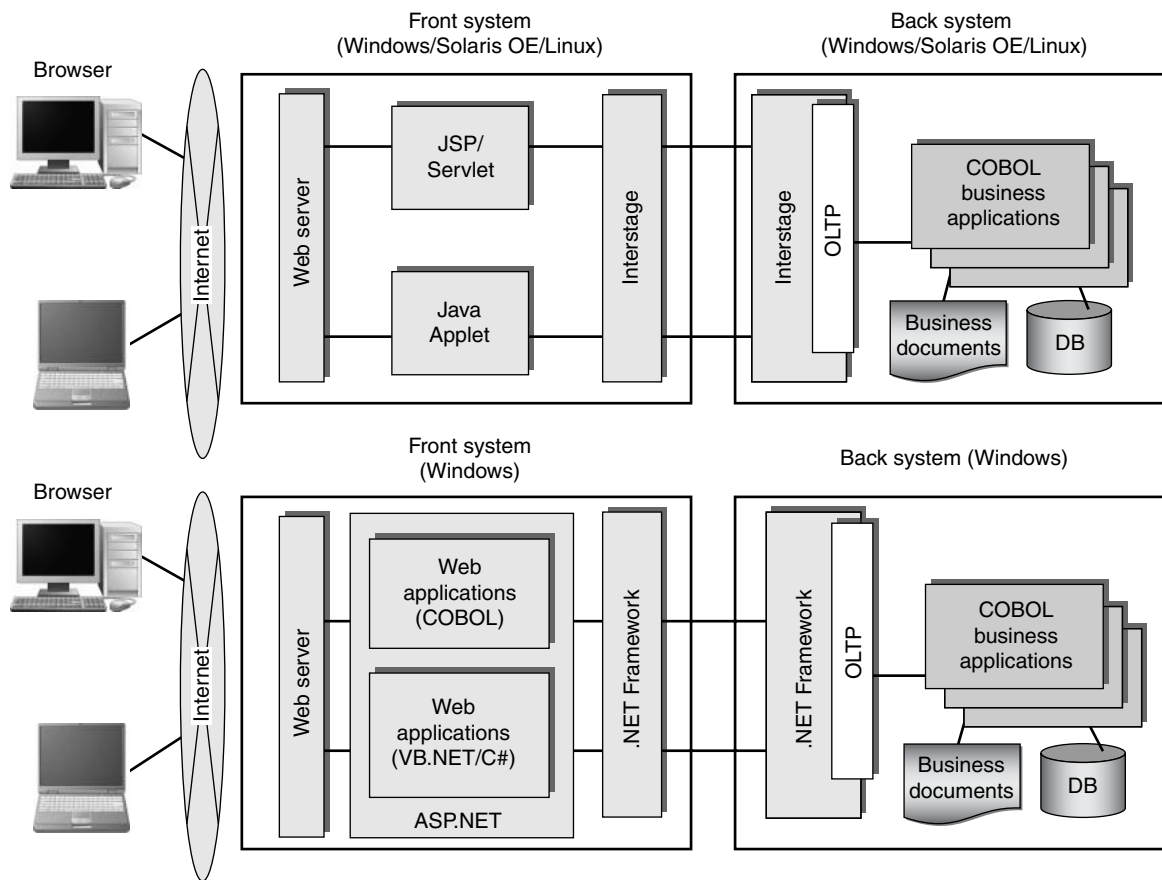


Figure 2 System that uses most appropriate languages.

10 000 nationwide retail stores. It has about a million CDs and 600 000 DVDs in stock, with about 600 000 unique products, and the data volume averages about 70 000 transactions per day (180 000 at peak times). Therefore, one of the key requirements for the new system was that it provide stable services.

2) Gradual migration and system expandability

Over the years, the company had used about 7500 COBOL programs, and about 4500 of these were marked for migration. Consequently, the goal was to gradually migrate the existing system to a new open system, while keeping the existing system in operation.

3) Cooperation with existing resources operated on the mainframe

In building the new system, Planet Co., Ltd.

and Fujitsu System Solutions Ltd. (Fsol) — to which Seikodo outsources tasks of its information division — worked with the company to quickly build a high-quality system.

2.1 New system configuration

It was decided to use CORBA for cooperation with the mainframe, COBOL for batch processing, and Java for the Internet. As a result, all Web/AP and DB servers became redundant and the system achieved the high levels of availability, performance, and expandability that were required (Figure 3).

COBOL was chosen for batch processing because it was necessary to make full use of the existing COBOL skills of Seikodo employees so the system could be quickly built and then

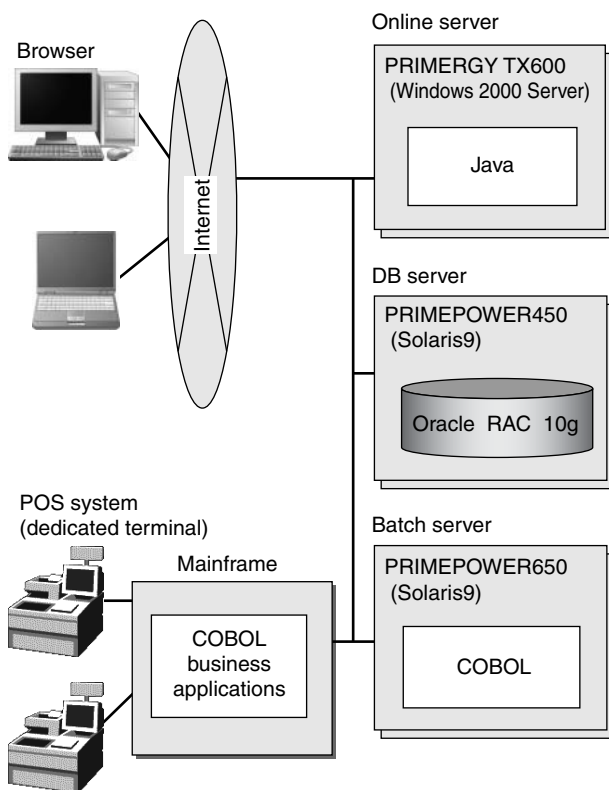


Figure 3
Seikodo's system configuration.

maintained by Seikodo without external help. Another reason for choosing COBOL was that the company could easily obtain outside help from people familiar with COBOL. There were also concerns that performance would be degraded if Java were adopted.

To handle Internet sales and inventory orders, the company originally planned to use Windows 2000 Server on the PCs in the company's head office and retail stores and reference and update them using Oracle Database (hereafter called Oracle DB) on Solaris. However, because some stores could not immediately replace their old mainframe terminals, a mechanism for accepting input from these terminals was needed on the mainframe. The company therefore decided to process the input from the stores using COBOL applications running on the mainframe, and a new COBOL application was developed for referenc-

ing and updating the integrated Oracle DB via the Interstage CORBA framework.

NetCOBOL was adopted in every aspect of the development of the COBOL applications on Solaris. Faster file and data processing were realized in the batch processing by using a high-performance data sorting and merging utility called PowerSORT.

The new batch processing system has a much better performance than the old one and can easily be expanded to cope with future business growth. To reduce risk, the migration of this system was done gradually (Figure 4).

2.2 Results of implementation

The new system surpassed the old one in terms of performance and was made highly expandable so it could cope with future business expansions.

The case study of Seikodo shows how the advantages of Java and COBOL can be used to build systems. There are also many architecture examples in which Java is only used for Web input and COBOL is used for the logic part.

Another major feature of the Seikodo system is that risk was low during its construction because the migration was done gradually through cooperation with the mainframe.

3. Case study of system built with .NET Framework in Japan

This section describes the COBOL system that was built on .NET Framework for Meitetsucom Co. Ltd. (hereafter called Meitetsucom).

Meitetsucom manages information systems for Nagoya Railroad Co., Ltd., which operates a wide variety of businesses (mainly in Nagoya), including railway, automotive, tourism, and airline agent businesses.

In 1986, Meitetsucom built a system on a mainframe that could search train timetables from within the company via a dedicated terminal. In 1998, the system was changed to UNIX and could be accessed by general users via the Internet.

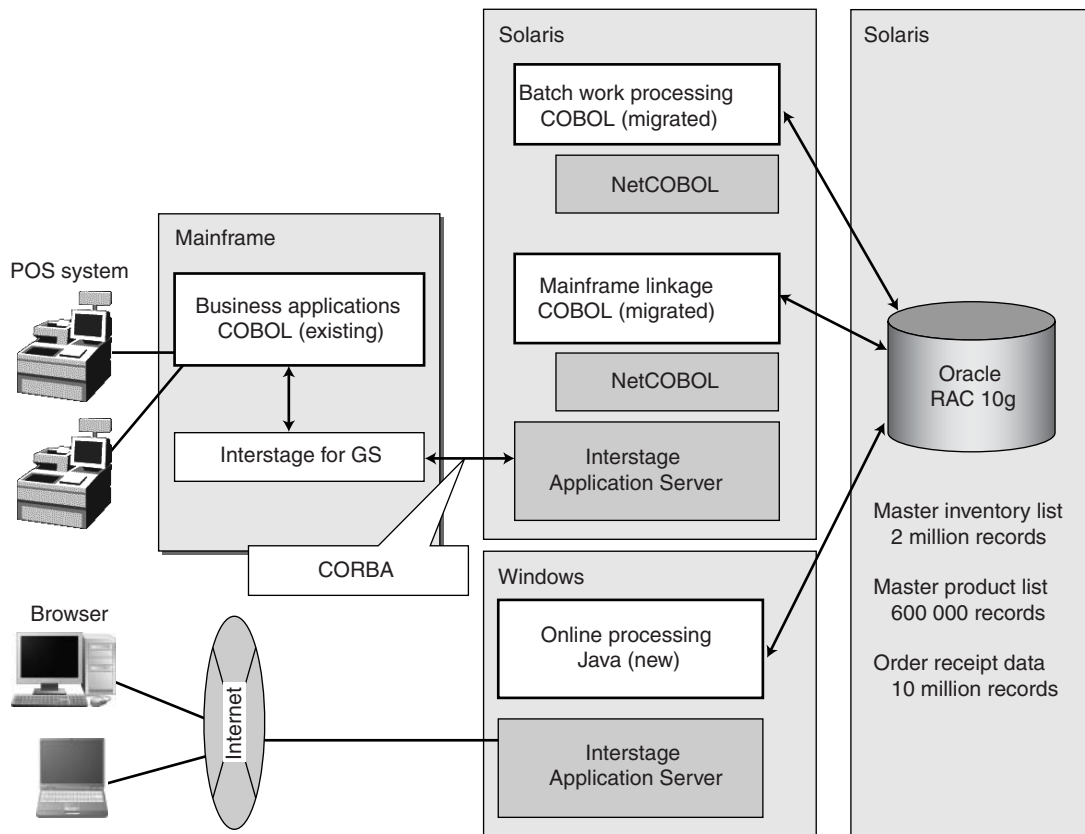


Figure 4 Migration of Seikodo's system.

Then, in 2004 the system was rebuilt using .NET Framework to increase its accessibility and improve its screen interface in response to the major timetable changes that occurred when the Central Japan International Airport was opened.³⁾ .NET Framework was chosen because it has the following features:

- 1) Outstanding performance and economy compared to Meitetsucom's current computer system
- 2) Microsoft's extensive expertise and the usage record of .NET Framework
- 3) Flexible development of Web and cellular phone screen interfaces with VB
- 4) Good use of existing COBOL assets
- 5) A common development environment, Visual Studio.NET, that enables multiple languages such as COBOL and VB to be used for increased development efficiency

The old system ran on a non-Fujitsu mainframe with a UNIX server using COBOL. However, Meitetsucom decided to use our NetCOBOL because they were impressed with its early support of .NET Framework.

3.1 New system configuration

The new system has two servers: an application/DB server, which uses COBOL and Oracle; and a Web server, which uses VB .NET. The two servers are connected by a Web service provided by .NET Framework (**Figure 5**).

Screen interfaces for three cellular phone companies were created with VB, and Macromedia's Shockwave Flash was used to improve the operability of the screens used to access the Internet.

About 30 000 steps of source code were migrated from existing COBOL programs for the

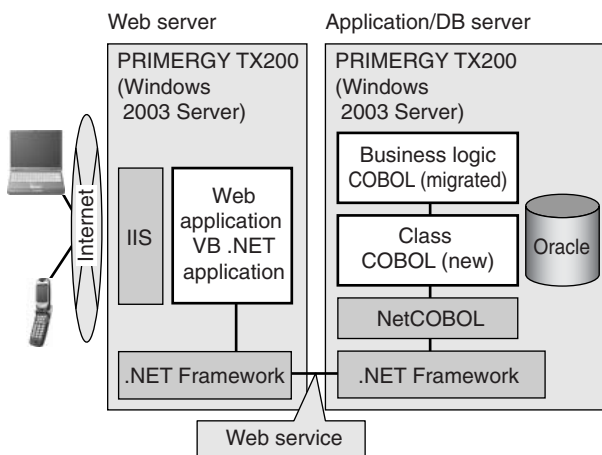


Figure 5
Outline of Meitetsucom's system.

business logic used in the application server. However, because NetCOBOL is highly compatible with the COBOLs of different companies and different operating systems, the system was developed in just five months (Figure 5).

3.2 Advantages of new system

By substantially reducing the hardware and software costs and simplifying the old system, the new system reduced the cost of operations management. In addition, services such as providing detailed information about train transfers and access via cellular phone were improved for customers. Performance, which had been an issue in the past, was improved to meet the goal of 160 calls per minute, leading to increased customer satisfaction.

The major feature of the Meitetsucom example is the use of different languages for different systems; namely, VB for the front system and COBOL for the back system.

NetCOBOL can be used with Visual Studio .NET to create flexible Web input screens, and this combination has been used to build many systems using only COBOL.

4. Case study of system built with .NET Framework in U.S.

Presidential Life Insurance Company was founded in 1965 and is located on the west bank of the Hudson River in Nyack, New York. It is a leading provider of fixed deferred and immediate annuities and life insurance products to financial service professionals and their clients across the U.S. The company has a revenue of \$344 million and employs about 100 people in its headquarters and regional offices.

Presidential Life recognized the benefits of moving away from the mainframe in the mid-90s and has been running the production CICS-based applications that support their annuity products, tax, and stockholder systems on Windows NT for several years. These applications comprise over 160 screens and over 650 COBOL programs.⁴⁾

4.1 Problems

Support for the operating system and software on which the applications depend was coming to an end. Therefore, Presidential Life had to find a solution that would keep them fully supported for the foreseeable future.

Presidential Life was also beginning to think it could take advantage of some of the GUI features and Web connectivity of newer releases of Windows.

4.2 Fujitsu solutions for new system

Resolving the support issues clearly required upgrading to the latest version of Windows and finding a newer product that would support CICS COBOL programs. Presidential Life was therefore pleased to find that NeoKicks^{note 1)} was ideal for CICS migration because not only did it work in Microsoft's strategic .NET environment but it also enabled them to upgrade their BMS^{note 2)} screens to ASP.NET pages.

note 1) NeoKicks is a tool that migrates mainframe CICS applications to the .NET environment.

note 2) Basic Mapping Support (BMS). A facility that handles data stream input and output from a terminal.

The COBOL part of the solution, NetCOBOL for .NET, which provides the foundation for NeoKicks, gave access to the extensive .NET Framework class libraries and also brought the benefits of Visual Studio .NET to Presidential Life's developers.

Using the Pervasive.SQL^{note 3)} for the COBOL data file support for migrating CICS code enabled Presidential Life to bring all of its data into a common file system because the company's life insurance systems were already using Btrieve.^{note 4)}

The primary reasons listed by Presidential Life for choosing Fujitsu Computer Systems^{note 5)} were:

- 1) Confidence in Fujitsu's software quality
- 2) Ease of migration provided by the NeoKicks Project Wizard, which converts COBOL programs and BMS maps into the new .NET format and migrates Visual Studio .NET projects.
- 3) Good availability of consultants to train and support staff during the migration gave assurance that the project would be completed successfully.

Presidential Life already had experience with Fujitsu's software products because they migrated to NetCOBOL for Windows in 2003. This gave them confidence in the reliability of Fujitsu's products and, after investigating other solutions that could support CICS in a Windows environment, they decided that NeoKicks with NetCOBOL would provide them with the agility they were seeking, both in terms of application development options and reasonable cost.

4.3 Final result

Clearly, the main benefits for Presidential Life was that they met their goals, especially the critical need to run their applications on NeoKicks and Pervasive.SQL. However, other benefits have

note 3) Databases of Pervasive Software

note 4) Btrieve is the predecessor of Pervasive.SQL.

note 5) Fujitsu Computer Systems offers a wide range of enterprise hardware and software products and services in the U.S.

become apparent during the course of the pilot projects, for example:

- 1) Certain applications can be adapted for Web use.
- 2) The performance of applications has increased.
- 3) All COBOL data files use a uniform file system (Btrieve).

5. Conclusion

NetCOBOL allows mainframe COBOL programs to be effectively utilized, whether working with Fujitsu's COBOL or one from another company. It can also be used with many system configurations and new technologies, for example, it can be used with Java and VB and cooperate with existing mainframes. The main advantage of using NetCOBOL is that it allows clients to develop and enhance their valuable COBOL software.

We hope that NetCOBOL will continue contributing to the next generation of business system architectures as a development environment for rapid, high-quality production of systems that are based on the SDAS architecture and have the reliability of mainframes, along with high compatibility and support for the latest technologies.

References

- 1) NetCOBOL.
<http://www.netcobol.com>
- 2) Case study of Seikodo Corporation. (in Japanese).
<http://www.cobol.gr.jp/>
- 3) Case study of Meitetsucom Corporation. (in Japanese).
<http://www.cobol.gr.jp/>
- 4) Presidential Life Refreshes CICS with NeoKicks.
http://www.netcobol.com/info/Presidential_Life_Case_Study.htm



Haruhiro Okishima, Fujitsu Ltd.

Mr. Okishima received the B.E. degree in Electronics Engineering from Nihon University, Tokyo, Japan in 1985. He joined Fujitsu Ltd. Kawasaki, Japan in 1985, where he has been engaged in development of middleware. Since 2001, he has also been engaged in support and development of NetCOBOL. He is a member of the COBOL Consortium in Japan.