

# COBOL 資産を活用した Web アプリケーション構築技術 \*

1Q-02

米光 哲哉 鶴澤 亨 只野 完二 宮崎 肇之 小室 彦三 †

日立製作所 システムソリューショングループ 生産技術統括本部 §

## 1 はじめに

近年における電子商取引や電子申請システムなどといった Web システム導入はますます拡大している。また、Web システムを迅速に構築することが求められている状況であり、いかにして生産性を上げ開発するかが重要でありまた課題にもなっている。その解決案の一つとして、既存の COBOL 資産を流用しかつ COBOL で業務ロジックを組み、今まで蓄積してきた開発ツールまたは手順を有効に活用することで生産性を向上させることができると考える。

そこで本稿では、Web アプリケーション構築時に何故 COBOL を活用するか理由を述べ、COBOL 資産を活用した Web アプリケーション構築事例を紹介する。また今後この構築例を弊社の情報部門に展開する上での課題について報告する。

## 2 COBOL の必要性

Web アプリケーションの開発は、Web サーバ上に Java™ (Servlet, JSP, EJB 等も含む) で構築することが中心となってきた。その中で COBOL 資産を活用する理由として以下があげられる。

第一の理由として、業務ロジックとして既に COBOL 資産が存在することである。業務処理の多くは COBOL 資産として既存のロジックが存在し、また新規に作成する場合でも、CASE ツール

を適用することで、ほぼ COBOL の業務ロジックが自動生成できる開発環境が既に整っている。同じ言語で再構築する場合でも著しく品質が低下する (バグ密度で 10 倍程度) のに、経験の少ない Java™ での再構築は現時点ではリスクが高い。

第二の理由として、基幹業務の中での COBOL の割合が多いことである。基幹業務を迅速に Web システムに移行するには、COBOL で作られた資産を流用し、Web にラッピングした方が生産性が高く Web システムに移行することができる。

第三の理由として COBOL プログラマーが Java™ プログラマーよりも未だに多いこともあげられる。今後 Web アプリケーションの中心が Java™ になりつつあることは否定できないが、早急に COBOL プログラマーが Java™ プログラマーに転換することも難しい。

## 3 Web アプリケーション開発スタイル

弊社の生産技術部門では、Web システムの開発において基準となる Web アプリケーション開発スタイルを設定している。Web アプリケーション開発スタイルは P 層、F 層、D 層からなるアプリケーションの 3 階層構造をどのように構築するかという観点で整理されており、5 つのパターンが用意されている。F 層の構築については顧客要件に基づいて選択肢を設け、P 層の構築については、JavaApplet を採用する 1 パターンを除いて、全てのスタイルに JSP/Servlet で構築することとしている。

以下に 5 つのパターンを示す。

- ① JSP/Servlet + JavaBeans
- ② JSP/Servlet + EJB (Enterprise JavaBeans)
- ③ JSP/Servlet + COBOL (CORBA を使用する)
- ④ JSP/Servlet + COBOL (CORBA を使用しな

\* Web Application Development by Utilizing Legacy COBOL Program

† Tetsuya Yonemitsu, Tooru Uzawa, Kanji Tadano, Tadashi Miyazaki, Hikoza Komuro

§ Hitachi Ltd. Solution Systems. Engineering Support Management Center

い)

#### ⑤JavaApplet+COBOL

### 4 構築事例

図1に今回構築した事例のシステム構成を示す。この事例は、上記 Web アプリケーション開発スタイル④のスタイルに属している。従来から適用実績のある OLTP(On-Line Transaction Processing)環境を活用して Web システムのトランザクション処理を可能とした構成である。

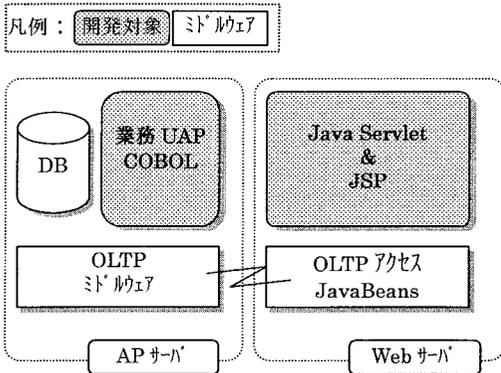


図1：システム構成図

AP サーバには、OLTP 上に COBOL 業務ロジックを配置する。AP サーバのハードウェアは、メインフレーム、ワークステーション、PC サーバを問わないが、今回の事例では PC サーバを利用した。

Web サーバには、フロントエンドの JSP、その JSP を制御する Servlet を配置する。また、Java™ と COBOL の間に OLTP ミドルウェアがあり、OLTP と通信するための JavaBeans を Web サーバ上に用意した。ブラウザから送られた入力情報を Web サーバが受け取り、OLTP 通信用 JavaBeans を経由して AP サーバの OLTP と通信を行うことで、Java™ アプリケーションは、COBOL とデータの受け渡しを可能とした。JSP では業務ロジックは記述せず、画面の入出力編集処理のみ記述している。

AP サーバの COBOL 業務ロジックは、実績のある既存の CASE ツールを適用し新規に作成し

た。この CASE ツールは、入出力条件、データ項目、レコード・ファイル・プログラム仕様を定義して COBOL プログラムを自動生成している。実績ある CASE ツールを利用することで、基幹となる業務プログラムの品質が向上しかつ高い生産性で開発することができる。JSP についても弊社の JSP 作成を支援するツールを使用し生産性を向上させている。

COBOL 業務プログラムの開発は従来からの開発手法をそのまま適用し、F 層の担当者は業務仕様のみ考慮し、P 層の担当者は画面レイアウト、画面からの入力編集処理、処理結果の出力編集処理のみ考慮し作業を分担したことも生産性を向上させている。

### 5 今後の課題

今回、OLTP 上の COBOL 業務ロジックを従来の開発手法で開発し、Web アプリケーションを構築することができることを示したことで、Web システムを構築する開発者へノウハウを提供できたと評価している。また同様のシステム構成で既存の OLTP 上で稼動している COBOL の基幹業務を Web システムに構築することが確認することができると考えている。

今回は OLTP 経由で COBOL プログラムとデータの送受信を行っているが、OLTP を介さずに COBOL プログラムとアクセスする JavaBeans を用いた評価を実施している。業務特有の入力チェックで既に COBOL プログラムのサブルーチンとして存在する場合、JSP から COBOL プログラムとアクセスする JavaBeans 経由で、COBOL のロジックを呼ぶことができる。既存の COBOL で作成された業務特有のサブルーチンを有効活用する開発手順も今後確立する必要があると考えている。

### 参考文献

[1]宮崎肇之 他：アプリケーション開発スタイルの標準化，情報処理学会第 63 回大会，2001.