

# ウェブアプリケーションの可能性を広げる Java™ 新技術 — FlyingServ® J-Frame Server®

FlyingServ® J-Frame Server®: New Java™ Technology Expanding Scope of Web Applications

宮崎 正志 舟城 亮一

■ MIYAZAKI Masashi

■ FUNAKI Ryoichi

現在、多くの業務システムが、ウェブプラットフォーム上に構築されるようになってきている。ところが、ウェブプラットフォームには業務システムに必要な画面や帳票の処理がサポートされておらず、開発者にとって大きな負担となっている。

FlyingServ® J-Frame Server®は、このような業務システム開発者に対して、Java™<sup>(注1)</sup>ベースのウェブアプリケーションと連携しながら、システムの画面操作性を向上させる手段を提供するミドルウェアである。この製品の持つ帳票サブシステム連携機能とCOBOL (COmmon Business Oriented Language) 連携機能により、ウェブアプリケーションの適用領域を広げることができる。

Many business application systems are now being built on the Web platform. The graphical user interface (GUI) capabilities of the Web platform are limited, however, making it difficult for developers to create rich forms that are typically used in conventional systems.

FlyingServ® J-Frame Server® is a framework that provides a rich, interactive GUI to the client. It is used in conjunction with Java™-based Web application programs, and helps developers to concentrate on the business logic rather than the GUI operation. By using its form function and COBOL interface function, it is possible to expand the use of the Web to a variety of applications.

## 1 まえがき

現在、企業の業務システムの多くがウェブ技術を利用したプラットフォーム上で構築されている。

その理由としては、ウェブプラットフォームで動作する業務システムは、クライアントにウェブブラウザだけがあればよく、従来のクライアント／サーバシステムに比較してTCO (Total Cost of Ownership) 削減に効果的であることが挙げられる。また、米国Sun Microsystems社が、1999年に、ウェブプラットフォームとしての標準仕様Java 2 Enterprise Edition™ (以下、J2EE™<sup>(注2)</sup>と言う)を発売し、システムの開発者が、プラットフォームベンダーに依存しないアプリケーションを開発することができるようになったことも大きい。

しかしながら、ウェブプラットフォームにも、実際に業務システムを開発する際には、いくつかの問題がある。

第一に、業務システムのユーザーインタフェースに関する問題である。ウェブブラウザは情報閲覧ツールとしては優れているが、入力ツールとしての操作性が悪い。そのため業務システムの開発者は、クライアント／サーバシステム並みの操作性を求めるシステムの利用者を満足させるために、苦勞を強いられている。

第二に、ウェブプラットフォームは、業務システムで大きなウェイトを占める帳票処理機能を提供していない。そのため、ウェブアプリケーションの開発者は、自分で帳票処理機能を開発する必要がある。

第三に、既存の業務システムとの共存を考える必要がある。現在、汎用機やオフィスサーバなどで構築された業務システムが多数稼働しており、これをウェブプラットフォームと共存させながら、どう発展させていくかが課題になっている。

ここでは、このような課題に対応する技術として東芝が開発したFlyingServ® J-Frame Server® (以下、J-Frame Server®と略記)について述べ、その仕組みを解説する。最初に、ウェブプラットフォームの課題と、それを解決するJ-Frame Server®のアーキテクチャを説明し、続いてJ-Frame Server®とウェブアプリケーション、帳票処理、及び、COBOLアプリケーションの連携機能について述べる。

## 2 ウェブプラットフォームにおける 業務システム開発の課題

ウェブプラットフォーム、特にJ2EE™の仕様は、トランザクション処理やデータアクセスといった、主に業務システムのビジネスロジックをサポートした内容となっている。ところが、実際の業務システムは、画面処理や帳票処理が大きなウェイトを占めており、これらは開発者が自分で作る必要がある。

初めに画面処理について説明する。画面処理は通常ウェブ

(注1)、(注2)、(注4) Java, J2EE, EJB 及びその他の Java を含む商標は、米国Sun Microsystems, Inc.の米国及びその他の国における登録商標又は商標。

ブラウザ上のスクリプト言語で開発する。前述のとおり、ウェブブラウザは入力ツールの操作性が悪く、それを補うために開発者が作成するプログラムの量はたいへん大きなものになる。それでも、かつてのクライアント/サーバシステムで多く使用された Visual Basic<sup>(注3)</sup> プログラムのような画面操作は実現できない。

最近、この種の課題に対応するため、ウェブブラウザに特殊なプログラムを組み込み、ウェブサーバからクライアントプログラムを自動的に送り込む、リッチクライアントと呼ばれる仕組みも登場している。しかし、これにも次のような問題があり、ウェブアプリケーション開発における課題を解決できていない。

- (1) クライアントに、リッチクライアント専用の実行環境が必要となり、運用管理負荷を高めてしまう。
- (2) リッチクライアントの画面アプリケーションとウェブアプリケーションは、かつてのクライアント/サーバシステムと同じ関係にあり、連携処理の開発や組合せ試験に工数がかかる。
- (3) クライアントの画面アプリケーションにビジネスロジックが組み込まれてしまうおそれがあり、これがプログラ

ムの保守性を悪くしてしまう。

業務システムで求められる画面操作例を図1に示す。

次に帳票について説明する。通常、Java<sup>TM</sup> プログラムで帳票を作る場合は、けい線の描画や印刷フォントの大きさなど、すべてをプログラム処理として行う必要がある。これではシステムのメンテナンス性が悪く、帳票のフォーマットが変わるたびにプログラムの変更が発生してしまう。

また、帳票処理はサーバで動作するので、いかにクライアントプリンタへ出力するかという課題もある。通常ウェブプラットフォームでは、直接、閲覧やプリント出力ができる PDF (Portable Document Format) 形式などのデータをサーバで作成し、ダウンロードすることが多いが、この方式だとクライアントにデータが残ったり、業務画面とは別に PDF ビューアが起動してしまい、業務システムで画面制御できなくなる問題がある。

最後に、汎用機やオフィスサーバ資産とウェブプラットフォームの共存について考える。

業務システムを新しいプラットフォームへ順次移行していく手法としては、新たに要件定義から始めて、システムを最初から作り直すスクラッチビルドの手法や、業務の見直しと

DB: DataBase

図1. ビジネス処理に必要な操作性の例 — 従来のクライアント/サーバシステムではあたりまえの処理が、ウェブプラットフォームでは難しい。  
Example of operations required for business processing

(注3) Visual Basicは、米国 Microsoft Corporation の米国及びその他の国における商標又は登録商標。

標準的なビジネスプロセスをパッケージ化したERP (Enterprise Resource Planning) の適用を組み合わせる方法などもある。しかし、新しいプラットフォームは必要でも、業務の中核の流れ自体は従来と変わらず、業務ロジックを変更する必要がないことも多い。こうした場合に、独自の仕様を盛り込んだ基本的な業務ロジックを生かし、これと新たなウェブアプリケーションをつないで業務を拡張するという手法が有効である。

ところが、COBOLアプリケーションは、ファイルへのアクセス方法や排他処理、及びデータの受け渡しなど、ウェブプラットフォームのアーキテクチャと異なる点が多く、プラットフォームのサポートなしにこのような処理を作り込むことは、開発者に大きな負担をかけてしまう。

### 3 J-Frame Server<sup>®</sup>のアーキテクチャ

J-Frame Server<sup>®</sup>の基本的なアーキテクチャは、“画面アプリケーションをサーバ側で動作可能とする”実行環境を提供することで、リッチクライアントに相当する機能を実現するというものである。

J-Frame Server<sup>®</sup>では、サーバ上で動作する画面アプリケーションの画面情報をクライアントに転送し、また、クライアントで操作したキー入力やマウスイベントをサーバ上の画面アプリケーションに転送する。これにより、ブラウザだけではできない、画面の部分更新や、操作に対するリアルタイムな応答が可能となる。クライアントでは、画面の描画とマウスやキーイベントを処理するだけのアプレットが動作する。したがって、クライアントには、アプリケーションロジックやデータはいっさいダウンロードされない。

J-Frame Server<sup>®</sup>は、クライアント環境としてリッチな操作性を提供しながら、アーキテクチャとしては、シンクライアントを実現したユニークな製品となっている。

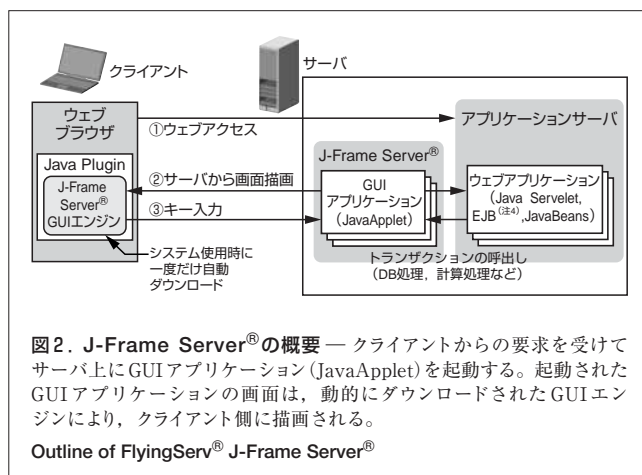
このように、J-Frame Server<sup>®</sup>は高度な操作性を実現するためのソフトウェアであるが、他社のリッチクライアント製品とは、次の点で異なる。

- (1) サーバサイドアーキテクチャである アプリケーションコードがクライアントで動作しないため、クライアントの環境依存が少なく、クライアントの管理が容易である。また、アプリケーションが、サーバ側にすべてあるため、セキュリティに対する配慮も最小限で済み、データベースへのアクセス機能や他ホストとの連携機能も実装しやすい。デバッグやトラブル解決もサーバ側ですべて行える。
- (2) 独自の画面開発言語を必要としない 他社のリッチクライアント製品は、独自の画面用API (Application Programming Interface)を使用したり、独自のビルダ

を使って画面を定義するが、J-Frame Server<sup>®</sup>は、Java<sup>™</sup>で作成したアプリケーションを実行することができる。Java<sup>™</sup>で開発できるメリットとして、豊富なJava<sup>™</sup>開発ツールが使用できることが挙げられる。GUI (Graphical User Interface)ビルダやパフォーマンス解析ツール、デバッガ、統合開発環境など、Java<sup>™</sup>プログラマーのための開発環境は、有償・無償を問わず非常に多くの製品が提供されており、その豊富で強力な開発環境を使用してGUIの開発を行うことができる。また、サードパーティ製の部品なども、Java<sup>™</sup>で作られたものであれば流用することが可能である。

- (3) 操作性を考慮した部品群を提供している 画面を設計するにあたり、オフィスコンピュータなどで培ったビジネス処理に基づく、操作性に優れた部品群を提供している。簡単なエントリ業務であればノンプログラミングで実現できる。

J-Frame Server<sup>®</sup>の概要を図2に、また、Java<sup>™</sup>の標準的な開発ツールEclipseを使った画面アプリケーション開発のようすを図3に示す。



### 4 J-Frame Server<sup>®</sup>とウェブアプリケーションの連携

通常、ウェブプラットフォーム上で動作するJ2EE<sup>™</sup>プログラムは、トランザクションやデータアクセスといった処理を行う。J-Frame Server<sup>®</sup>の画面アプリケーションはサーバで動作しているため、これらJ2EE<sup>™</sup>プログラムを直接呼び出すことができる。例えば、画面アプリケーションの検索ボタンを押すと、J2EE<sup>™</sup>として作られた検索処理を呼び出し、結果を画面表示するといった処理が簡単に作成できる。このような処理は、リッチクライアントの環境をクライアントに持つ製品では実現できない。

J-Frame Server<sup>®</sup>とウェブアプリケーションの連携イメージを図4に示す。

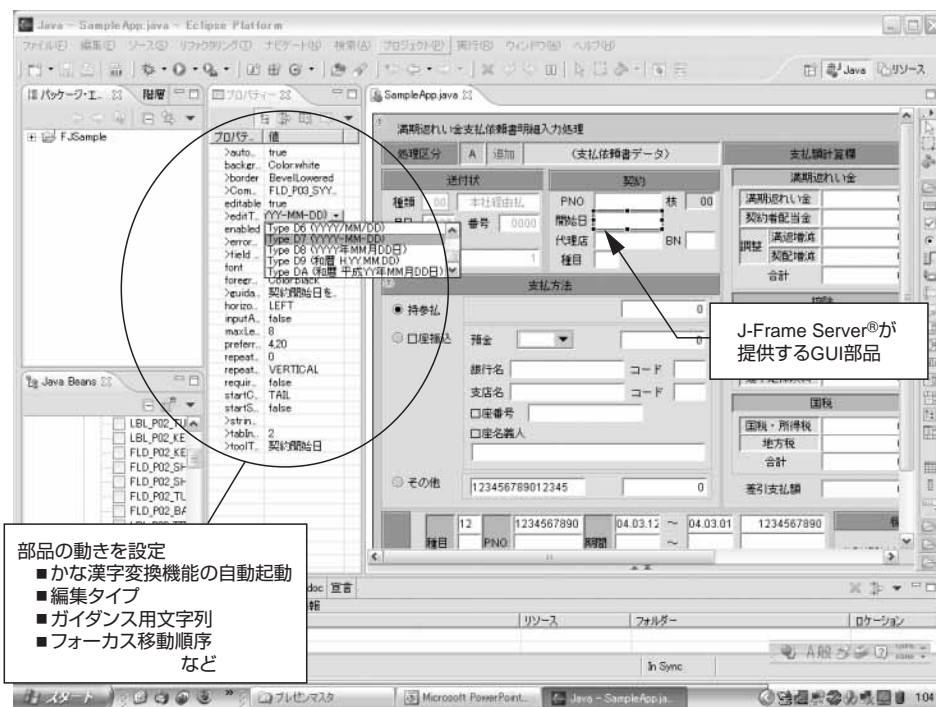


図3. Eclipseを使った画面アプリケーション開発のようす — Java™ 開発ツールの標準であるEclipseを使った開発の例である。GUI部品を画面におき、プロパティを設定することで画面アプリケーションを開発できる。

GUI application development using "Eclipse"

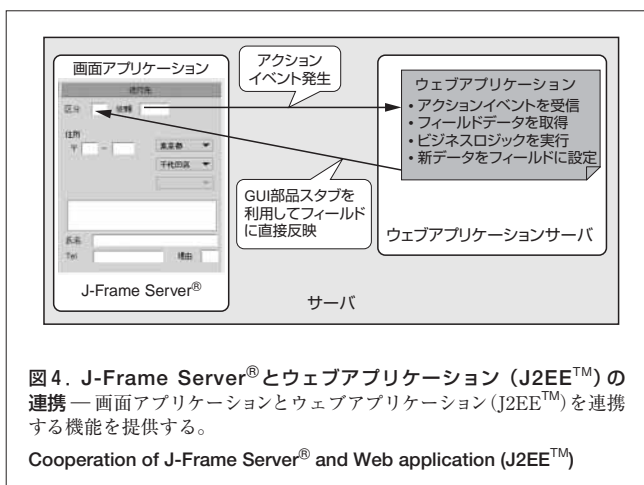


図4. J-Frame Server®とウェブアプリケーション (J2EE™) の連携 — 画面アプリケーションとウェブアプリケーション (J2EE™) を連携する機能を提供する。

Cooperation of J-Frame Server® and Web application (J2EE™)

## 5 J-Frame Server®と帳票処理の連携

J-Frame Server®では、画面処理と帳票処理を一貫して開発できる帳票オプションを用意している。専用の帳票エディタで帳票のレイアウトを作成し、帳票出力時にアプリケーションからデータをオーバーレイする。こうすることによって、帳票レイアウトが変更になっても、出力データが変わらなければ、アプリケーションを変更する必要はなくなる。

この帳票オプションは、当社がオフィスコンピュータの時代から培ってきた業種アプリケーション向け帳票ノウハウを集

めたもので、画面処理との一貫した開発や、クライアントプリンタへの出力など、次のような特長を持つ。

- (1) 画面と帳票処理を一貫した開発 Java™による画面と帳票の処理を一体で開発することができ、業務アプリケーションの開発生産性を高めることができる。また、業務画面への帳票プレビューの表示や、印刷指示ボタンの実装など、帳票とより高度に連携した業務システムを実装することができる。
- (2) クライアントにデータを残さないセキュアな印刷機能 開発された業務アプリケーションは、J-Frame Server®によってサーバ側で実行されるため、印刷やプレビューに使用される帳票データがクライアントに保存されることがなく、高いセキュリティを保つことができる。
- (3) クライアント側プリンタへのダイレクト印刷機能

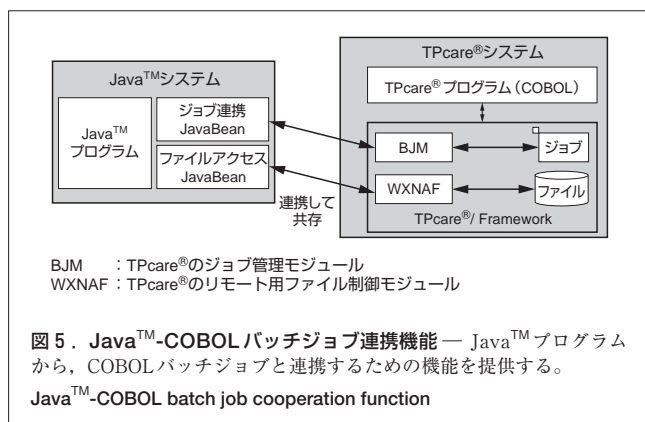
クライアントに接続されたプリンタへ、サーバから直接印刷するための機能をサポートする。この際、帳票データはサーバで各種プリンタ言語 (LIPS<sup>(注5)</sup>, Post-Script) やPDF形式に変換され、クライアントにキャッシュされることなく、ダイレクトにクライアントプリンタに出力されるので、セキュリティの高いシステムを実現できる。

(注5) LIPSは、キャノン(株)の商標。

## 6 J-Frame Server®とCOBOLアプリケーションの連携

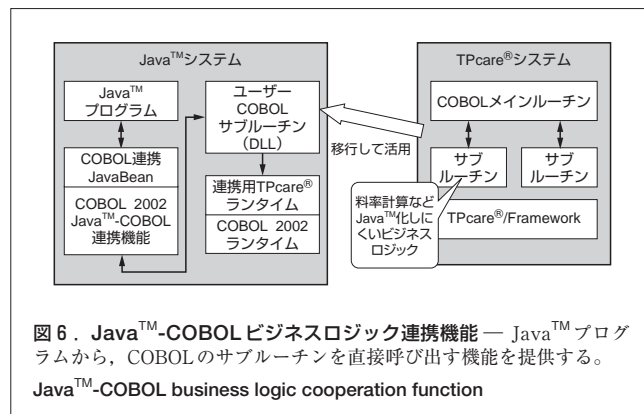
J-Frame Server®を使えば、ウェブプラットフォーム上で、使い勝手の良い画面アプリケーションとCOBOL言語で書かれたアプリケーションを連携させることができる。当社製品のユーザーにおいても、オフィスサーバTP90シリーズ上のシステムや、そのオープン環境への移行ソフトウェアであるTPcare®により移行されたシステムなど、COBOL言語によるシステムが多数稼働しており、J-Frame Server®を使えば、こうしたシステムをウェブプラットフォームへ移行することができる。

まず、Java™とCOBOLバッチジョブの連携について、具体的に説明する。一般に、バッチ的に行われる処理は、業務の中身が変わらないかぎり、従来のシステムを利用することが望ましく、コストをかけて移行する必要はない。J-Frame Server®を使用する場合でも、バッチ処理については、既存のシステムをそのまま使用することができる。変える必要がないものはそのまま流用する、という考えである。この既存のバッチジョブのシステムと連携するのがJava™-COBOLバッチジョブ連携機能であり、その構成を図5に示す。



次に、Java™アプリケーションと、ビジネスロジックを実装したCOBOLサブルーチンとの連携機能について述べる。これは、Java™アプリケーションから直接COBOLのサブルーチンを読み出す機能を提供するものである。既存のCOBOLアプリケーションに対し、Java™のインターフェースでラッピングすることにより、ビジネスロジックのJavaBeans化、EJB化、ウェブサービス化などを可能とする。

Java™からCOBOLを読み出す場合には、Java™とCOBOLのデータ形式の変換を行わなければならない。この機能では、COBOLサブルーチンをDLL(Dynamic Link Library)化する機能や、Java™とCOBOLの間のデータ変換を行うた



めのJavaBeansの生成機能などの開発環境を合わせて提供している。また、実行時にJCL(Job Control Language)を指定することが可能で、ファイルのアサイン処理など、従来のTPcare®で実現していた機能と同様の処理を実行することができる。

この機能の構成を図6に示す。

## 7 あとがき

ここでは、ウェブプラットフォームが業務システムの標準プラットフォームになってきた背景とウェブアプリケーション開発における課題、更にそれらの問題を解決するJ-Frame Server®のアーキテクチャについて述べた。

今後ともオープンなウェブプラットフォームの特長を生かしつつ、業務システムにとって必要なプラットフォームになるよう製品を発展させていく。



宮崎 正志 MIYAZAKI Masashi

東芝ソリューション(株) プラットフォームソリューション事業部 ソフトウェア開発部参事。FlyingServ®及びJava™関連製品の開発に従事。

Toshiba Solutions Corp.



舟城 亮一 FUNAKI Ryoichi

東芝ソリューション(株) プラットフォームソリューション事業部 商品企画部参事。FlyingServ®及びJava™関連製品の商品企画に従事。

Toshiba Solutions Corp.