

3 階層 Web アプリケーションシステムの構築とその評価

菊池 史典[†] 小泉 寿男[†]

[†] 東京電機大学大学院理工学研究科

Web 技術を基盤とするシステムにおいては、アプリケーションサーバの登場により、システムロジックをより分散させようという形態が注目され、その契機となったのが J2EE である。J2EE はサーバ・サイドに特化した技術・環境を提供するもので、これを用いたシステムの構築が多く行われている。しかしながら、J2EE を用いた開発手法には様々な形態があり、応用システムに対応した構築方法が重要になってきている。

本論文では、J2EE に準拠した Servlet, JSP, EJB などの技術を用いた Web アプリケーションシステムを 3 層クライアントサーバシステム上に構築し、いくつかの典型的なシステムによる評価を行った。この評価として、ソフトウェア作成手法の違いによる比較評価、異なるアプリケーションサーバ上での構築における評価および考察を行う。

An Integration and Its Evaluation of Three-Tier Web Application Systems

Fuminori Kikuchi[†] Hisao Koizumi[†]

[†] Department of Computers and Systems, Graduate School of Tokyo Denki University

In systems based on web technology, the introduction of the application server has attracted attention as a means of ensuring wider distribution of system logic encouraged by JAVA 2 Platform Enterprise Edition (J2EE). J2EE provides an engineering environment specifically for the server side, and is being widely used to configure systems. However, many different approaches to J2EE development are being adopted, and it is vital that the methods should be appropriate for the systems being configured.

This paper describes the use of Servlet, JSP, EJB and other techniques under J2EE to configure web application systems on a three-tier client server system, evaluating it for a number of representative systems. Also this paper considers both comparative evaluations of alternative approaches to software production and configuration on different application servers.

1. はじめに

Web 技術を基盤とするシステムにおいては、アプリケーションサーバの登場によりシステムロジックをより分散させようという形態が注目されてきており、その契機となったのが J2EE である [1][2]。J2EE はサーバ・サイドに Java を活用することで特化した API や技術・環境を提供するもので、これを用いたシステムの構築が多くなりつつある。J2EE を用いた開発手法には Servlet, JSP, JavaBeans, EJB といった複数の形態があり、Web アプリケーション構築目的により適用形態やプログラミングの仕方

が変わってくるので、応用システムに対応した構築方法が重要になってくる。

本稿では、J2EE に準拠した技術 (Servlet, JSP, EJB など) を用いた Web アプリケーションシステムを 3 層クライアントサーバシステム上に構築し、典型的な商品購買システムのプロトタイプで評価した結果を述べる。システムの評価として、そのシステムの作成手法の違いによる比較評価、異なるアプリケーションサーバ上での構築における評価および考察を目的とする。

2. アプリケーションサーバの構築方法

3層クライアントサーバシステムを基盤とする Web アプリケーションシステムの構築には、図1に示すようにいくつかのパターンが考えられる。すなわち、開発環境であるアプリケーションサーバには開発用から大規模システム向けのものまで提供されており、さらにその上で動作するアプリケーションの作成方法の違いにより構築方法が変わってくる。

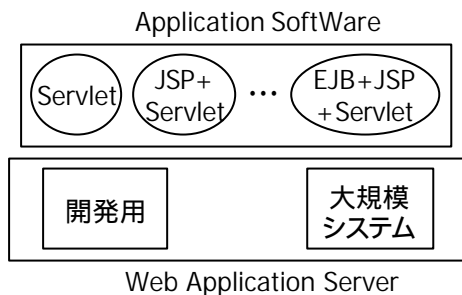


図 1 構築方法

2.1 J2EE による構築手法の違い

Web アプリケーションサーバ上にシステムを構築するとき用いる J2EE の違いによるパターンが考えられる。J2EE では、JavaBeans (EJB) + JSP + Servlet という Web アプリケーションモデルが採用されており、その根拠となっているのが Model-View-Controller パラダイム、すなわち MVC モデルである。MVC モデルは、Smalltalk を用いてプログラムを開発するために提唱されたフレームワークで、Model-View-Controller にそれぞれ、Beans + JSP + Servlet を用いることが多くなっている。全ての処理を Servlet や JSP で行う、または JSP と JavaBeans を用いる、さらにそれに EJB を加える、EJB にするなど構築方法はいくつかのパターンがある。これらの手法の中で、Servlet のみでシステムの構築を行っても、例えばデータベースに接続してその情報を利用するような Web アプリケーションを開発することは可能である。しかし、同様な処理が可能であっても、複雑で理解しにくいプログラムと、道筋がわかりやすく管理しやすいプログラムとでは、実践的に大きな違いがある。そこでアプリケーションの作成手法に以下のような違いを持たせたシ

ステムを構築し、比較評価を行う。

- (1) JavaBeans + Servlet
- (2) JavaBeans + Servlet + JSP
- (3) EJB + Servlet + JSP

2.2 異なるアプリケーションサーバでの構築

J2EE に準拠したアプリケーションサーバを用いる場合、以下のような機能・特徴をつかむことが重要となる。

・マルチプラットフォーム：対応可能な OS・Web サーバの比較。

・アプリケーションサービス機能：トランザクション処理、CORBA・RMI といった分散オブジェクト技術のサポート。

・システムサービス機能：処理が増えた場合にいかに容易にかつどの規模まで対応できるかの拡張性とフェールオーバーなどの信頼性。

・開発環境：シームレスな設計、グループ開発を含む統合環境をいかに提供するか。

そこで、同じアプリケーションソフトウェアの作成手法で、これらの動作環境の異なるアプリケーションサーバ上での構築を行う。アプリケーションソフトウェアの部分は、J2EE における MVC モデルに対応した手法を用いることにし、EJB + Servlet + JSP で構築する。

2.3 評価方法

2-1, 2-2 で示したシステムの構築方法での評価を以下の内容で行う。

・記述したプログラムの役割の違い：Web ブラウザの表示と業務ロジックの処理をどのように分担して開発するかは、生産性・メンテナンス性向上の要因である。

・ロジックの記述の違い：構築方法によってプログラムの記述すべきタグの分量が変わり、メンテナンス性・保守性に影響する。

・開発、構築のしやすさ：各アプリケーションサーバの特徴機能、実行・開発環境には違いがあり、構築の容易性に影響する。

3. システムの構築パターン

図1で示したシステム構築のパターンに基づいた構築方法を以下に示す。

3.1 アプリケーションソフトウェアの作成手法

アプリケーションソフトウェアの作成手法に違いを持たせた構築を行う。構築環境となる

WebサーバにはApache1.3,アプリケーションサーバにはアイ・ティ・フロンティア社のJRun3.0 Developer版を用いて構築を行う。

(1) Servlet + JavaBeans

入出力および制御の部分を Servlet で,ロジックの部分に JavaBeansを用いて構築を行う。

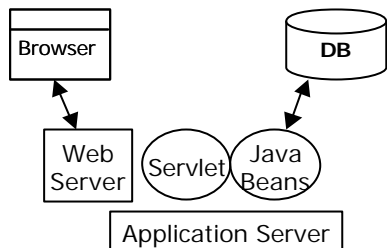


図 2-1 Servlet + JavaBeans

(2) Servlet + JSP + JavaBeans

入力および制御部分を Servlet,ロジックの部分を JavaBeans,出力の部分に JSPを用いて構築を行う。

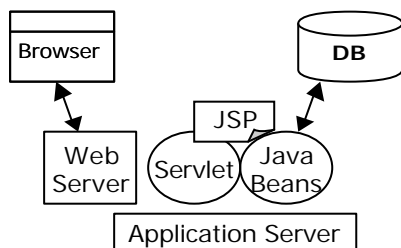


図 2-2 Servlet+JavaBeans+JSP

(3) Servlet + JSP + EJB

入力および制御部分を Servlet,ロジックの部分を EJB,出力の部分に JSPを用いて構築を行う。

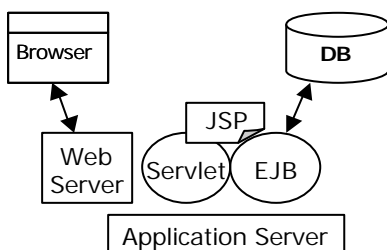


図 2-3 Servlet + EJB+JSP

3.2 アプリケーションサーバの環境

同一のアプリケーションソフトウェアの動作

に対し,構築環境となるアプリケーションサーバを変えてシステムの構築を行う。Apache 1.3, JRun3.0による構築と Oracle Web Listener, Oracle Application Server4.0.8 による構築を行う。アプリケーションソフトウェアはServlet + JSP + EJB の組み合わせでシステムの構築をする。

JRun3.0 は企業基幹システムや,E コマースアプリケーションを構築・運営するためのJavaアプリケーションサーバであり,J2EE に準拠している。モジュール化された構造を持ち,開発者は必要なコンポーネントのみを使用することができる。開発用のため,ServletおよびJSP,EJB の同時接続数は制限内で使用する。

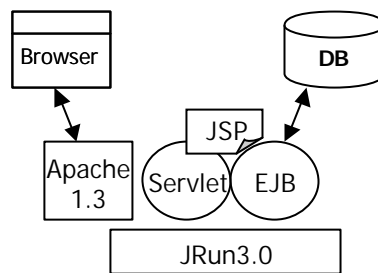


図 3-1 JRun による構築

Oracle Application Server 4.0.8 は,データベース「Oracle」との親和性を持っている。カートリッジ機能により複数言語でアプリケーションを開発する際に必要な共通基盤を提供しており,J2EE にも準拠している。

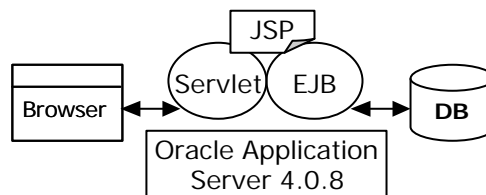


図 3-2 Oracle Application Server による構築

4. 評価システムの構築

4.1 システムの実行環境

J2EE を用いた応用システムの構築として,商品購買システムのプロトタイプ構築を行った。実行環境を図4に示す。クライアントは各種ブラウザを用いてサーバにアクセスをする。データベースには Oracle を用い,データベースとの連携は JDBC ドライバを用いた。

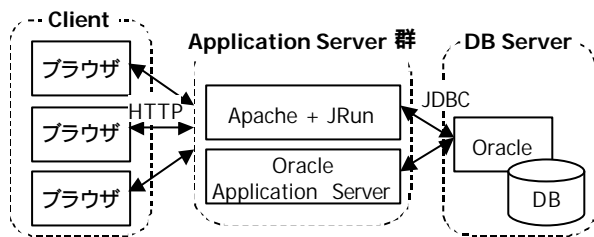


図 3 実行環境

4.2 評価システムの機能

今回構築した商品購買システムは「商品カタログから商品の選択」、「商品情報の表示」、「ショッピングカート」の機能を持ったサイトを構成する。DB サーバには、ユーザ情報、商品カテゴリ情報およびカテゴリ毎に分けられた商品情報が格納されている。

商品情報の表示機能：ユーザのログイン完了後、商品カテゴリ一覧を表示する。

ショッピングカート機能：ユーザが選択した商品の一覧を保持し、処理の終了とともに破棄する。選択した商品の数量変更や削除を可能にする。

ユーザの状態管理：複数ページから構成される Web アプリケーションの場合、所定の画面遷移のみ動作するようにする。

システムの画面遷移を図 5 に示す。一連の動きは次のようになる。

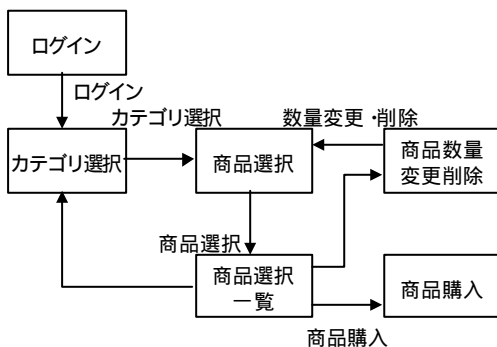


図 4 画面遷移

ユーザはログインページからユーザ ID、パスワードを用いてログインを行う。商品カテゴリの一覧が表示され、その中から 1 つを選択する。選択した商品カテゴリの商品の一覧が表示

されるので選択する。

選択完了後、ショッピングカート内の情報が表示され「商品の追加」、「数量変更」、「購入」のいずれかを選択する。

「追加」は商品カテゴリ一覧ページへ、「数量変更」は数量変更ページへ、「購入」は購入ページへ移動する。

4.3 クラス構造

図 5 に示した画面遷移に沿ったシステムのクラス構造を図 6 に示す。JavaBeans + JSP + Servlet の構成になっている。

まずユーザ情報を Session オブジェクト (ShopCart) に格納し、ログインが成功したらカテゴリ選択画面が表示される。選択したカテゴリにある商品の詳細な情報が出力され、欲しい商品を選択すると ItemToCart が、Session オブジェクトを取得する。ItemListDBBean のインスタンスを作成し、選択された商品情報を ShopCart に格納する。Confirm でショッピングカート内の情報を表示後、変更・追加・購入の処理を振り分け、追加・変更の場合は Session オブジェクトを取得後数量の変更をする。購入の場合はカート内の情報を表示し、Session オブジェクトを破棄する。

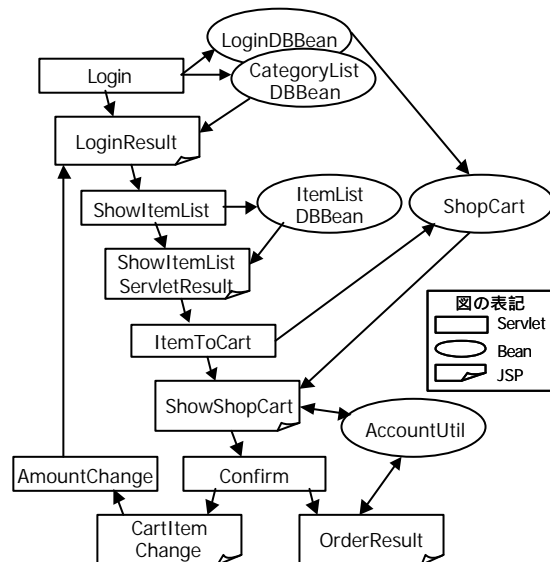


図 5 システムのクラス構造

次にそれぞれのクラスやファイルの機能について述べる。

(1) Servlet

- ・ AmountChange : カート内のアイテム数量を変更, アイテム項目の削除をする. カートの情報を表示する JSP の呼び出し.
- ・ Confirm : ショッピングカート内の情報を表示した後の処理の振り分けをする.
- ・ ItemToCart : アイテム一覧を再取得し, 選択したアイテムをセッション・オブジェクトに格納する.
- ・ Login : セッション・オブジェクト作成後ユーザ情報を取得する. ユーザが存在すれば, セッション・オブジェクトに格納, カテゴリ一覧を取得する.
- ・ ShowItemList : 選択された商品カテゴリから商品一覧を取得し, 商品一覧を表示する JSP を呼び出す.

(2) JavaBeans

- ・ AccountUtil : 金額・総量を計算し, その結果を JSP に渡す.
- ・ CategoryListDBBean : 商品のカテゴリ一覧をデータベースから取得し JSP に渡す.
- ・ LoginDBBean : ユーザ ID, パスワードからデータベース検索を行う.
- ・ ShopCart : ユーザ情報の保持やショッピングカート機能の実装をする.

(3) JSP

- ・ LoginResult.jsp : ログインしたユーザ名と商品カテゴリ一覧からカテゴリを選択するフォームを表示する.
- ・ ShowItemListServletResult.jsp : 選択されたカテゴリの商品一覧と商品を複数選択するフォームを表示する.
- ・ ShowShopCart.jsp : カートの情報と「追加」「数量変更」「購入」の選択肢を持つフォームを表示する.
- ・ CartItemChange.jsp : カートのアイテム数を変更するフォームを表示する.
- ・ orderResult.jsp : 購入した商品の一覧を表示し, セッション・オブジェクトを破棄する.

5. システムの実証と評価

5.1 実行結果

ブラウザからログイン画面にアクセスしログインを行ってカテゴリ一覧を表示する(図 7-1). カテゴリを選択し商品一覧ページが表示され

(図 7-2), カート内情報から数量の変更・商品追加等を行い, 購入されると購入した商品の情報と支払う合計金額が表示される(図 7-3).



図 7-1 Category 選択画面

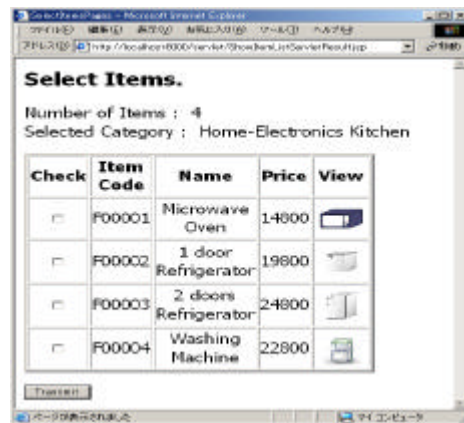


図 7-2 商品選択画面

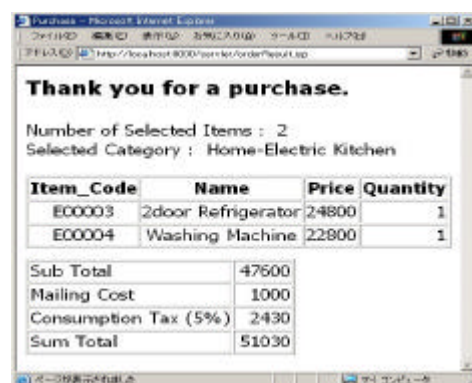


図 7-3 商品購入画面

5.2 評価および考察

(1) 作成手法の違い

図 2-1 と図 2-2 で比較した場合の違いはブラウザへの結果表示部分に JSP を用いたかどうかである. 図 2-1 ではクライアントからの入力

処理および結果出力処理を Servlet のみで行うため、1つの Servlet ファイルの記述量が非常に多くなる。Servlet のみで結果表示部分を記述する場合、出力用の命令が必要になる。したがって、結果出力部分が多ければ多いほど書くプログラムの量は増え、プログラムを作る際の大きな手間になり、また修正したいときのメンテナンス性にも欠ける。それに対し図 2-2 では結果出力部分は JSP が行うため、Servlet に JSP を呼び出すための記述が必要となるので記述量が増えるが、入力出力処理を分けることができ修正を行うなどのメンテナンス性が高いことが言える。処理を分けることで分散開発にも対応可能であるが、プログラムを行う上で明確な開発モデルが必要であると言える。

図 2-2 と図 2-3 で比較した場合の違いはロジックの部分が JavaBeans か EJB であるかである。トランザクション処理に適している EJB の方が業務形態の変化やアクセスの増加に対して柔軟に対応でき、Servlet から呼び出される部品を別のサーバに配置しセキュリティやスケラビリティを向上できる。プログラムの記述に着目すると、JavaBeans から EJB に移行する場合にある一定の制約があるため、設計の段階から EJB を考慮した設計が必要である EJB の Bean の利用には、これを呼び出すインタフェースなどの外部プログラムと、これらをアプリケーションサーバ上に配置するためにデプロイする必要があり、プログラムの記述量や作業が増える。しかしながら、インタフェースの中身は決まった手順で作成可能であり、必要な制御部分を記述した EJB 雛型を用意しておけば、外部プログラムとやり取りをするためのプロパティ、業務ロジック、プロパティに対して値の設定や参照をするためのアクセサ・メソッドを流用することで、EJB の Bean が容易に作成することが可能となる。

(2) アプリケーションサーバの違い

JRun には Apache を、Oracle Application Server には Oracle Web Listener をそれぞれ Web サーバに用いて構築を行ったが、両者複数の Web サーバをサポートしており構築環境に合わせた選択が可能である。アプリケーションサーバには、2.2 で述べた構成・機能上の比較

点がある。今回評価するために構築した商品購買システムの範囲では、両者の間での目立った性能の差は特に検出されなかった。クライアントからの大量の同時アクセスが発生した場合の応答時間、およびデータベースへアクセスするトランザクション数が同時に増えた場合にはシステムの応答時間に差が出るはずであり、現在その比較を進めている。また、複数のアプリケーションサーバで負荷分散したときの実行時間の比較および効果の測定も進めている。

6. まとめ

3階層システムを基盤とする Web アプリケーションシステムとして、商品購買システムを構築し、アプリケーションの作成手法の違いによる比較評価、異なるアプリケーションサーバ上での構築における評価および考察を行った。アプリケーションソフトウェア作成手法による特徴を確認でき、アプリケーションサーバ構築方法による特徴は確認できなかったが、今後は大量負荷同時発生時におけるアプリケーションサーバ構成およびオブジェクトの配置によるシステムの性能評価、負荷分散について評価・検証を行っていく予定である。また、SOAP をベースとした Web サービス[3]に対応した応用システムと、そのインフラである 3 層構造との関連性について評価していくこととしている。

<参考文献>

- [1] JavaTM BluePrints :
<http://java.sun.com/blueprints/>
- [2] 宮崎肇之他著, “Web アプリケーション開発スタイル標準化”, IPSJ 第 63 回全国大会講演論文集(1), pp133-134.
- [3] 青山幹雄他著, “e-ビジネスを実現するソフトウェア”, IPSJ Magazine, Vol.42 No.9 Sep. 2001 pp855-895.
- [4] VladaMatena・BethStearns, “Enterprise JavaBeansTM 開発ガイド”, PEARSON EDUCATION JAPAN, pp1-50(2001).
- [5] Tong Weigin, Meng Rui, Dong Jingyi, “Experience in Pure Java Implementation of Parallel Libraries”, 15th ICOIN, pp307-312.