

MVC モデルを適用した Web アプリケーション構築技法の評価

2J-04

齊藤直樹 菊池史典 小泉寿男
東京電機大学理工学部

1 はじめに

近年 Web アプリケーションの開発は、J2EE の提唱する環境が用いられるケースが多くなってきている。また、J2EE アプリケーションの特徴として、コンポーネントの使用、既存のシステムをそのまま利用できるなどがあり、開発する上で有効であると考えられる。構築の方法には、すべてを JSP、Servlet で処理する方法、または JSP、JavaBeans を用いる方法、それに EJB を加える方法などいくつかの構成がある。

J2EE では、Web アプリケーション開発のデザインパターンとして MVC アーキテクチャが採用されており [1]、Model-View-Controller にそれぞれ、EJB-JSP-Servlet を用いるケースが多い。

本稿では、J2EE における MVC アーキテクチャに基づいたシステム構築について評価を行った。

2 J2EE における MVC

Model : ビジネス・ロジックとデータへのアクセス方法を記述する。実装は EJB(又は JavaBeans) でおこなう。

View : クライアントからのリクエストに応じて、Model を可視化する。実装は JSP でおこなう。

Controller : Model のデータ管理と View の表示を管理・制御する。実装は Servlet でおこなう。

2.1 MVC によるシステム構成

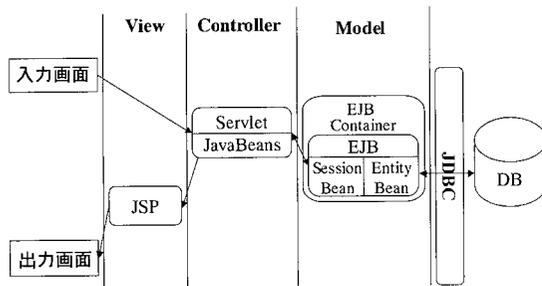


図 1 MVC によるシステムの模式図

図 1 は J2EE の MVC アーキテクチャに基づいた、

An evaluation of Web application construction technique using the MVC model
Naoki Saito, Kikuchi Fuminori, Koizumi Hisao
Department of Computers and Systems Engineering Tokyo Denki University

クライアントからのリクエストに応じレスポンスを返すシステムの概略図である。まずクライアントからのリクエストが、Controller に対して送信される。Controller はリクエストに応じ、適切な処理を行う Model を呼ぶ。Model は EJB でなく、JavaBeans によって実装する構成もある。呼ばれた Model は、受け取った値をロジックにかける。Controller は Model から処理結果を受け取り、それを View に反映させる。View は Web ページを生成し、クライアントへレスポンスを返す。MVC によって構築されたシステムはこのように Servlet を基点として動作し、クライアントからのリクエストに対して応答を返す。

3 J2EE のアプリケーションモデル

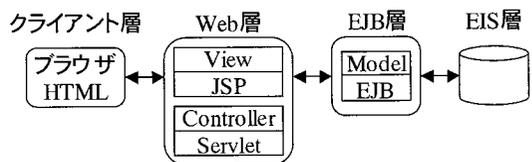


図 2 J2EE のアプリケーションモデル

クライアント層 (Client-Side Presentation) : ユーザーからの入出力の受け付け、ロジックの実行結果の表示をする。

Web 層 (Server-Side Presentation) : クライアントからのリクエストを受け、EJB 層へその内容を通知し、ビジネスロジックの実行を依頼する。そして、レスポンスをクライアントに返す。ビジネスロジックは実装しないため、バックエンドへの直接的な接続は行わない。

EJB 層 (Server-Side Business Logic) : Web 層の依頼に応じてビジネスロジックを実行する。必要に応じて、EIS 層へのアクセスを行う。

EIS 層 (Enterprise Information System) : ビジネスロジック実行に必要なリソースを提供する。Web システムにおいては DBMS、Web システム以外の既存の情報システム、CORBA オブジェクトなどのバックエンドを指す。

J2EE では、3 層構造における中間層を Web 層と EJB 層に分けることで、入力情報だけを EJB

層に与えることができ、ビジネスロジック部分のコンポーネントである EJB の独立性が高められている。これによりクライアントへの依存度が低下し、EJB の再利用性を高めることができる。

4 評価システム

MVC、及び J2EE のアーキテクチャに基づき、システムを実装した。各実装は、クライアント：Web ブラウザ、コンテンツ管理をおこなう Web サーバ：Apache、Web アプリケーションの実装をおこなう AP サーバ：Tomcat、DB サーバ：Oracle8i をそれぞれ使用した。

まずクライアントは Web ブラウザを用いて Web サーバにアクセスし、入力フォームからリクエストを送信する。リクエストを受け取った AP サーバ内の Servlet は、適切な EJB コンポーネントを選択する。EJB は受け取ったデータに応じた処理をおこない、値を Servlet へ返す。実装において分散トランザクションが必要な場合は EntityBean、そうでない場合は SessionBean を用いた。EJB より値を受け取った Servlet は、フロントコンポーネントを介して JSP へ値を渡す。JSP は値に応じて動的に HTML ファイルを生成する。クライアントは生成された HTML ファイルを、Web サーバを通じて受け取る。システムの構成図を示す。

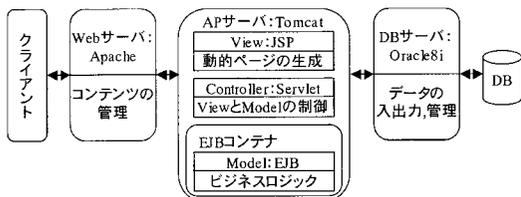


図3 実装システムの構成

5 MVCの評価・考察

5.1 MVCモデルの課題点

- (1) MVC の構造を常に構築するため、簡単なシステムの場合、必要以上に複雑な実装となる
- (2) Model が内部状態の更新・伝播を頻繁に行うと、画面の変更を頻繁に行う必要がある
- (3) View と Controller の依存度が高く、個別の再利用が難しい
- (4) View と Controller の Model への依存度が高く、Model のインタフェースの変更が View と Controller に影響を及ぼす

5.2 J2EEにおけるMVCの課題点

SessionBeanからEntityBeanを呼ぶものが存在すること、ControllerがViewとModelの二つの部分を扱う構造であることにより、Controllerをロジックの依頼者とする場合、ControllerがWeb層とEJB層に分割されるため、ModelとControllerの境界が明確ではなくなる。

5.3 考察

5.1で挙げたViewとControllerの依存関係と5.2で挙げた課題点を総合して考えると、MVCを適用したアプリケーションの開発においても、あるリクエストに対し応答を返す対話型のシステムにおいて、一連の構成要素それぞれがある程度の依存関係に置かれている。これにより、M-V-Cの各部の開発時において、Viewの開発者がControllerに加えModelの仕様も知っている必要がある。この場合、MVCモデルの適用によって得ようとした機能の分割による開発・管理の容易性の確保やModel(EJB)のコンポーネント化は十分に実現されていないといえる。現在これらのような問題の解決のため、開発におけるルールの制定が進められている。

そこで、システムの構築においてM-V-Cの各依存関係は、完全になくなるものではないとした場合、これらを分割して開発する方法を常に選択する必要はないと考えることもできる。プログラムとしてはM-C、V-Cの分離をおこないつつ、それぞれの仕様を知る人間が各部の開発をおこなうことで開発の効率は上がる。作成されたものはCの部分のみを各システムのCに組み込むことでM、Vを用いることができる。

6 まとめ

MVCの適用によるM-V-C各部の独立した開発法は以上に挙げた理由により、モデルの適用によって得ようとした独立性の確保や依存性の低下などが十分に達成されていないとした。

そこで開発方法の一つとして、依存関係にあるModelとController、ViewとControllerの各要素を同時に作成した場合について考察した。今後は各々の課題について、考察・検証・評価をおこなう。

<参考文献>

- [1]Java™ BluePrints :
<http://java.sun.com/blueprints/>
- [2]Special Issue on Object-Oriented Technology,
Communications of the ACM, Vol.38, No.10, 1995