



最新Java技術による ECサイト構築の要点 Vol.2

WebSphere Commerce Suite V5.1 Customization

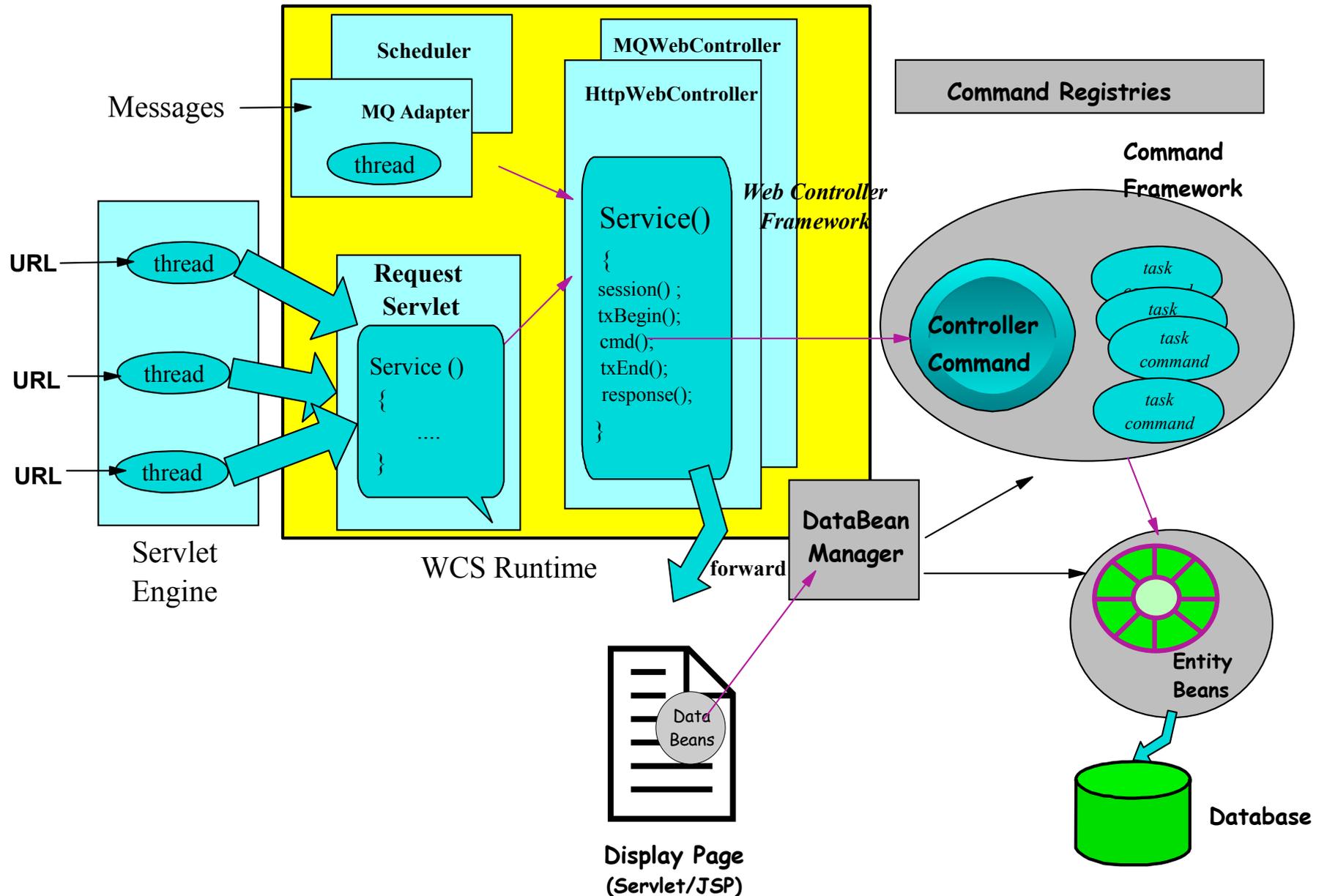
WebSphere Commerce Suite V5.1

Agenda

- ▶ WCS V5.1アーキテクチャー
- ▶ カスタマイズ
- ▶ 参考資料

WCS V5.1 アーキテクチャー

Websphere Commerce Server Components



WebSphere Commerce Site V5.1

Server Components (1)

▶ RequestServlet

- `com.ibm.commerce.server.RequestServlet`
- リクエストをディスパッチするだけのServlet
 - `/webapp/commerce/wcs/stores/servlet/`によって起動し、URLから実行すべきパラメーターとコマンドを受け取り、使用すべきController(`HTTPWebController`)の`service()`メソッドを起動するServlet

▶ WebController

- WCSサーバーにアクセスするコンポーネント、プロトコルの違いにより分かれる
 - httpリクエストは`HTTPWebController`にディスパッチされる
 - `com.ibm.commerce.webcontroller.HttpWebController` extends `WebController`
- 実行する機能は以下のとおり
 1. HTTP/HTTPSのチェック
 2. URLパラメーターとコマンド引数のマップ
 3. JTAトランザクションの開始
 4. `ControllerCommand`の実行
 5. Exception発生時のRetry
 6. JTA Commit/Rollback

Server Components (2)

▶ Controller Commands

- 実際のビジネスロジックを実行するJavaプログラム
- URLに指定されるコマンドと1:1のクラスのマップは以下の表に記載されている
 - URLREG URL-INTERFACE
 - CMDREG INTERFACE-CLASS
- 中でTaskCommandsを呼ぶこともあり、呼ばないこともある
- 結果表示用のViewCommandsをWebControllerに返す。

▶ Task Commands

- 従来のOFに相当する詳細のビジネスロジック
- ControllerCommandsから呼ばれる
- 同じくCMDREGに登録されている

▶ View Commands

- 通常及びエラー表示用のコマンド
- 以下の三種類がある
 - Redirect View Command
 - Direct View Command
 - Forward View Command
- VIEWREG表にControllerCommandに指定されているViewNameと呼ばれるクラス、JSPのマップがある
 - Product/CategoryDisplayコマンドに関しては、別の表(DISPENTREL、DISPCGPRREL)に記載されている

Server Components (3)

▶ EJB

- WCSでのデータベースアクセスは全てEJBを介してなされる
 - WASのEJBコンテナによりパーシステンスを管理
 - データアクセスコードとビジネスロジックを分離
- 基本は各テーブルにマップされたCMP Entity Beanを使用
 - 大量データ検索時のSessionBeanも多少あり

▶ AccessBean

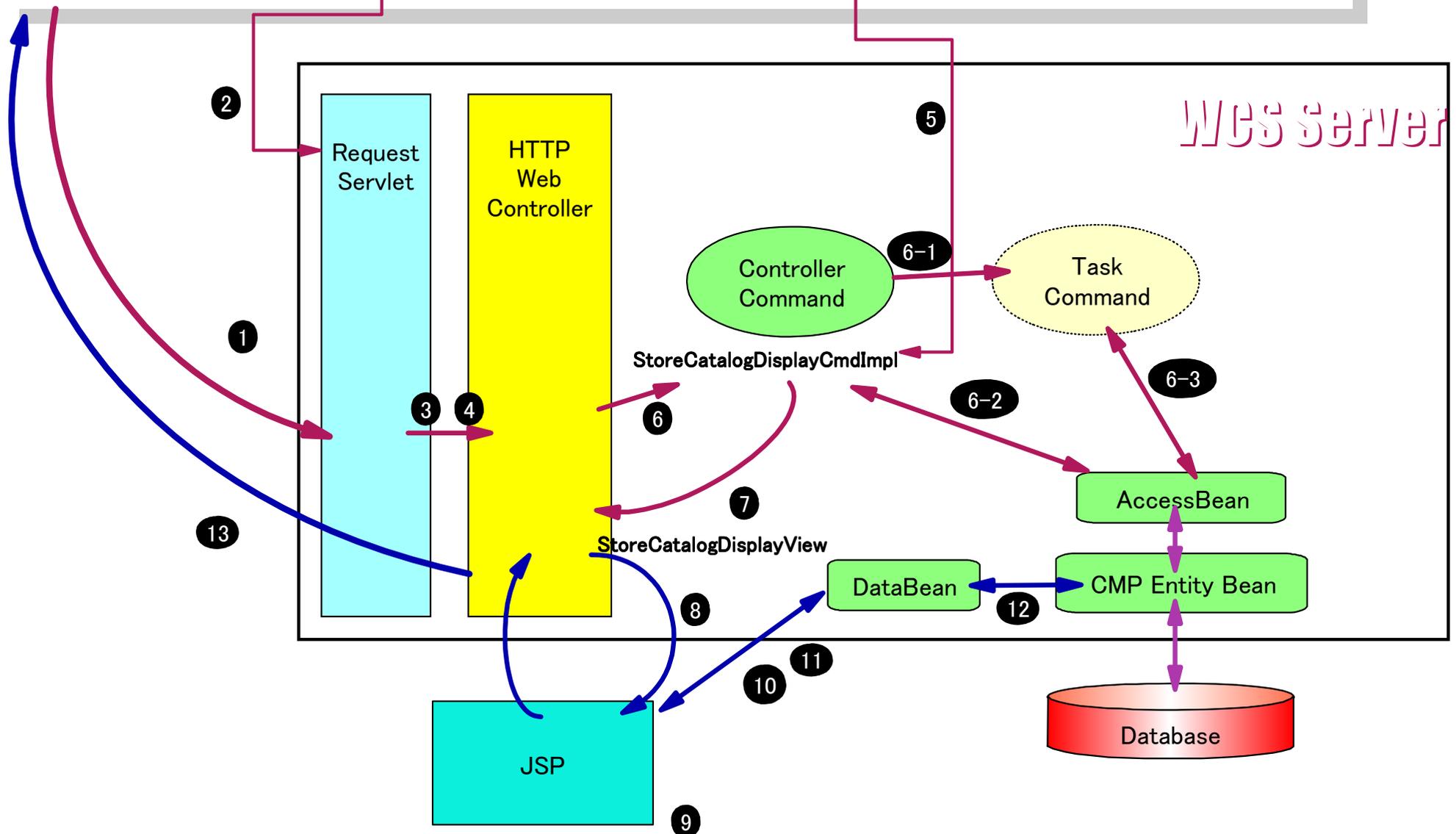
- WCSの各種コマンドはEJBを直接扱わず、AccessBeanを使用する
 - JNDIやHome/Remote InterfaceといったEJB独自の複雑なプログラミングを簡略化
- AccessBean自体はVAJのツールを使用した作成が可能

▶ Databean

- JSPからのアクセスはDataBeanを介してなされる
 - EJBへのアクセスもAccessBeanをextendしたSmartDataBeanを介する

Command Flow

http://wcs.ibm.com/webapp/wcs/stores/servlet/StoreCatalogDisplay?storeId=10001&catalogId=10001



■ URL Command : StoreCatalogDisplayコマンド

- ストアのトップカテゴリを表示するコマンド
- パラメータとしてはstoreId(ストアID)、catalogId(カタログId)、langId(言語Id)をHTTP Requestより取得

■ Command Flow

1. ショッパーのBrowserよりURLでコマンドが呼ばれる
 - http://wcs.ibm.com/webapp/wcs/stores/servlet/StoreCatalogDisplay?storeId=10001&catalogId=10001
2. Web ServerはWASのプラグイン経由でWASはServlet Engineを呼び出し、そこからRepository情報よりwebapp/wcs/stores/web/servletをRequest Servletに関連付け処理を渡す
3. RequestServletはHTTP Adapter Managerに要求を渡し、Browserからの要求であることを解析しHTTP Browser Adapterに処理を渡す
4. HTTP Browser AdapterはWeb Controllerに処理を渡す
5. HTTP Web ControllerはCommand Registry (URLREG、CMDREG) よりどのコマンドを実行すべきか(この場合、StoreCatalogDisplayからStoreCatalogDisplayCmdImplクラス)を取得
6. 適切なControllerCommandが実行されるがその際には以下のようなオプションがありうる
 - ControllerCommand自身がAccessBean-EJBを介してDB処理を実行
 - ControllerCommandが1つもしくは複数のTaskCommandを実行
 - TaskCommandもAccessBean-EJBを介してDB処理を実行
7. ControllerCommandは実行終了に伴い、呼び出すべきViewNameをWeb Controllerに返す
8. HTTP Web ControllerはCommandRegistry (VIEWREG) よりDevice Typeに応じたViewCommandを実行
9. ViewCommandはJSPに処理をForward
10. JSP内のDataBeanより動的ページを作成するためにDataBeanManagerがDataBeanをActivateする
11. DataBeanManagerは必要に応じてDataBeanCommandを実行する
12. DataBeanがExtendされたAccessBeanを介してEJB経由でデータベース情報を取得
13. 結果の動的ページをブラウザーに返す

Command Registry

▶ URLREG

STOREENT_ID	URL	HTTPS	INTERFACENAME
0	InterestItemAdd	1	com.ibm.commerce.interestitems.commands.InterestItemAddCmd

▶ CMDREG

STOREENT_ID	INTERFACENAME	CLASSNAME
0	com.ibm.commerce.interestitems.commands.InterestItemAddCmd	com.ibm.commerce.interestitems.commands.InterestItemAddCmdImpl

▶ VIEWREG

STOREENT_ID	VIEWNAME	DEVICE_FMT_ID	INTERFACENAME	CLASSNAME	PROPERTIES
1	StoreCatalogDisplayView	-1	com.ibm.commerce.command.ForwardViewCommand	com.ibm.commerce.command.HttpForwardViewCommandImpl	docname =StoreCatalogDisplay.jsp

Command Registry

- WCSのコマンド定義情報は以下の表内に保管され、取得される
 - 新規にコマンドを作成した場合、JSPをカスタマイズした場合などは下記の情報を挿入／更新する必要がある

▶ URLREG

- Browserから投げられるURL CommandとController CommandのInterfaceのマップ
 - StoreIdをキーとするので、Storeごとに別ControllerCommandを定義することが可能
 - HTTPかHTTPSか、認証を必要とするコマンドかなどの定義も

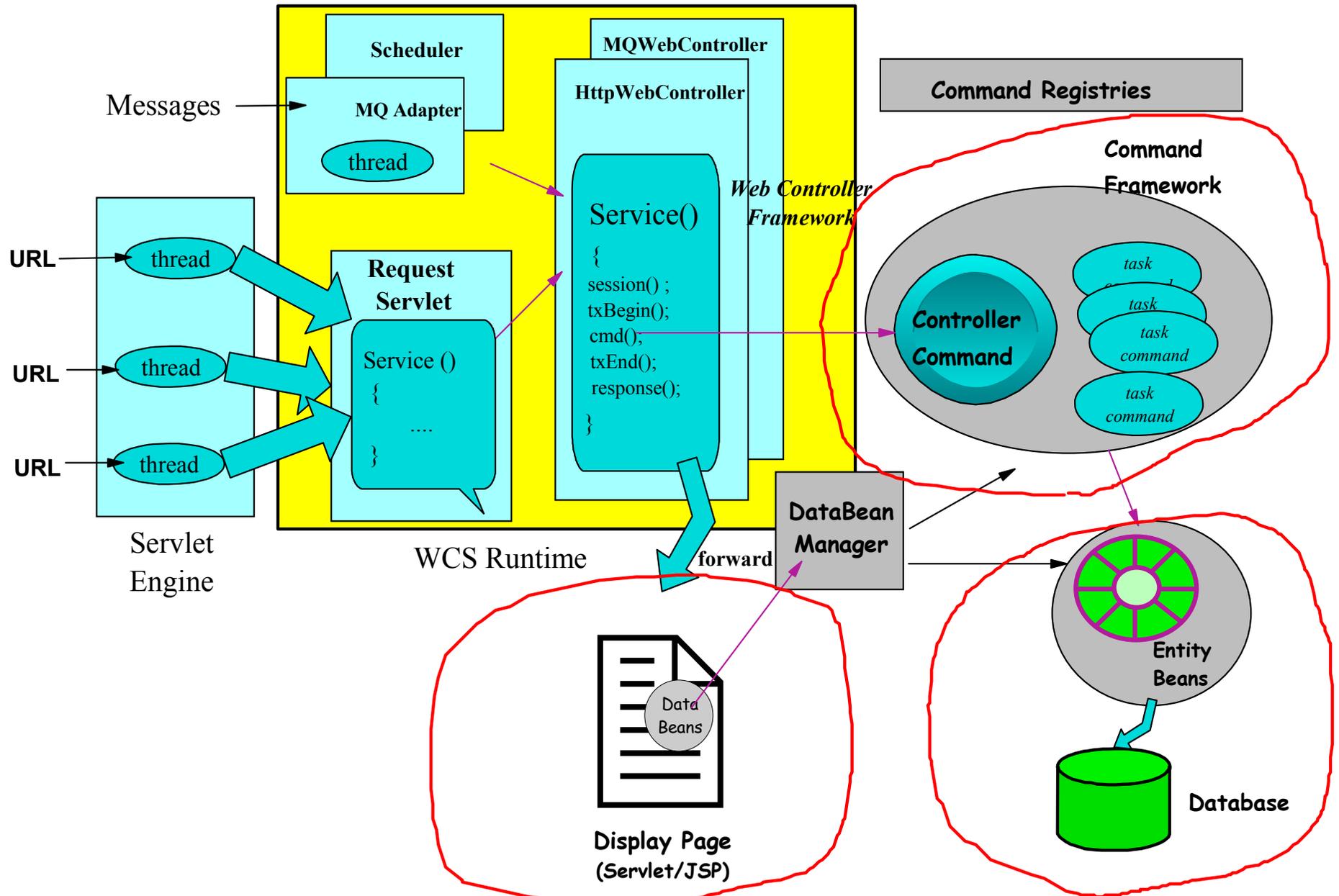
▶ CMDREG

- Controller CommandのInterfaceとClassのマップ
 - Interfaceにdefault classが定義されていれば定義は必要ない

▶ VIEWREG

- Controller Commandから返されるViewNameとViewCommand、JSPのマップ
 - ストアごと、デバイスフォーマットごとに別のViewCommand/JSPを定義可能(携帯端末対応)

カスタマイズ対象



カスタマイズ対象

▶ プレゼンテーションロジック

- JSP (Java Server Pages)のカスタマイズ
 - HTMLでのLook&Feelの変更
 - Data BeanのカスタマイズによるDB表示項目の変更
 - プロパティファイルなどを利用したマルチカルチャー環境の実現
 - パーベイスブ機能のための、携帯用ページの追加作成

▶ ビジネスロジック

- Javaコマンド群のカスタマイズ
 - Controller Commands カスタマイズ対象
 - Task Commands カスタマイズ対象
 - View Commands カスタマイズ対象ではない
- オリジナルのクラス群をVAJプロジェクトファイルの形で提供
 - 既存ビジネスロジックの変更
 - 新規ビジネスロジックの追加
- オブジェクト
 - Databaseの拡張に伴い、EntityBean、AccessBean、DataBeanなどをカスタマイズ

カスタマイズ

カスタマイズ手順

▶ 手順

- 要件定義、ストアフローの検討、カスタマイズ機能の抽出
 - スタアフローの検討
 - 使用機能の取捨選択
- プレゼンテーションロジックの変更
 - 基本デザインの取り込み
 - 表示項目の変更
 - 新規ページ作成
- ビジネスロジックの拡張
 - 新規コマンド作成
 - 既存コマンドの変更
- データ入れ込み
- テスト、修正

▶ 対象

- プレゼンテーションロジック
 - JSPのカスタマイズ
 - HTMLのカスタマイズ
- ビジネスロジックのカスタマイズ
 - EJBのカスタマイズ
 - Commandのカスタマイズ

カスタマイズに必要なもの

▶ スキル

- Java
- Enterprise JavaBeans
- Database
- JavaServerPages
- HTML
- WebSphere Application Server V3.5
- VisualAge for Java V3.5

▶ リソース

- WebSphere Commerce Studio V5.1
 - PageDesignerをJSP作成に使用
- VisualAge for Java Enterprise Edition V3.5
 - EJBの修正を行わない場合、WCStudio Developer's EditionにバンドルのVisualAge for Java Professional Editionを使用することも可能

JSPのカスタマイズ (1)

▶ 使用されているJSPファイルを特定する

- CommandRegistryより特定ページで使用されているJSPファイルを取得
 1. 使用するURL Commandをオンラインヘルプより取得
 - Topics → Reference → Commands → URL Commands
 - 使用コマンドのBehaviorの項で呼び出すViewNameをチェック
 - サンプルストアのJSPに関してはオンラインヘルプに詳細の説明あり
 2. VIEWREG表よりViewNameをキーにpropertiesからJSPファイル名を取得
 3. WAS WebPathから指定JSPファイルを選択
 - JSPファイル内でINCLUDEの形でヘッダーやフッターなどのファイルを呼んでいる場合もある
 4. さらにJSPによっては内部ロジックにより複数JSPへ分岐することもある
 5. 必要に応じ、新たなJSPファイルを作成／コピー
 - その場合VIEWREG表の定義を更新する
- 商品ページ、カテゴリーページなどは別の表にJSPとの関連を保管
 - DISPENTREL(商品)、DISPCGPRREL(カテゴリ)に保管
 - ユーザーに応じた細かなJSPの対応が必要なため

トップページ

- URLからStoreCatalogDisplayコマンドを呼んでいることが分かる
- StoreCatalogDisplayの説明にはStoreCatalogDisplayView ViewCommandを呼んでいるとある
- VIEWREGにはViewName=StoreCatalogDisplayViewのJSPはStoreCatalogDisplay.jspが定義されている

Example

JSPのカスタマイズ (2)

▶ JSPファイルを修正する

- デザイン(HTML)の修正
 - 各ページごとにベースデザインに基づきHTML部分の修正を行う
 - INCLUDEページもあわせて修正
 - 提供Sample(InFashion)はStyleSheet(css)を使用しているのでCSSの変更も含む
- 入力、表示項目の修正
 - 結果表示画面であれば、必要データのDataBeanを取り込むことで表示項目をカスタマイズ
 - 入力画面でも使用コマンドが受け付けうるパラメーターなどをHTMLに追加
- その他
 - 条件分岐
 - 条件によって呼び出すJSPファイルやURLCommandを変えるといった若干のプログラムロジックがあるのでそれらの修正を行う
 - JavaScriptなどの処理変更
 - Cache機能によりサーバー側に特定ページはキャッシュされるので、変更の反映が無い場合はキャッシュを消去

Example

トップページ

- Look&Feelを変えるためには、StoreCatalogDisplay.jspのほかにそこからINCLUDEされるheader.jsp、sidebar.jsp、footer.jspの修正が必要
- さらにInFashion.cssのStyleSheetの定義項目も必要に応じて変える

ショッピング登録ページ

- Register.jspからは入力項目を元にUserRegistrationAddコマンドを呼び出すが、必須パラメーターのうち不要なものをHiddenタグで渡している
- この必須部分を入力項目に追加する、その他必要な入力項目を追加するためにはRegister.jspファイルのFORMの修正を行う

▶ EJBを追加する

- WCSの表に関してはEntityBean、AccessBean、DataBeanなどはあらかじめ提供されているので追加の必要は無い
- カスタマイズ上、新規表を追加した際にCommand構造からアクセスさせるためにEntityBean、AccessBean、DataBeanなどを作成する必要がある
- EJBのカスタマイズを行う場合、原則としてVAJ Enterprise Editionが必要

■ アプローチ

1. EntityBeanを作成し、VAJにより表作成を行う
 2. 表を作成し、VAJによりEntityBean作成を行う
 3. 表とEntityBean両者を作成し、VAJ上でマップさせる
- 推奨は3のケース

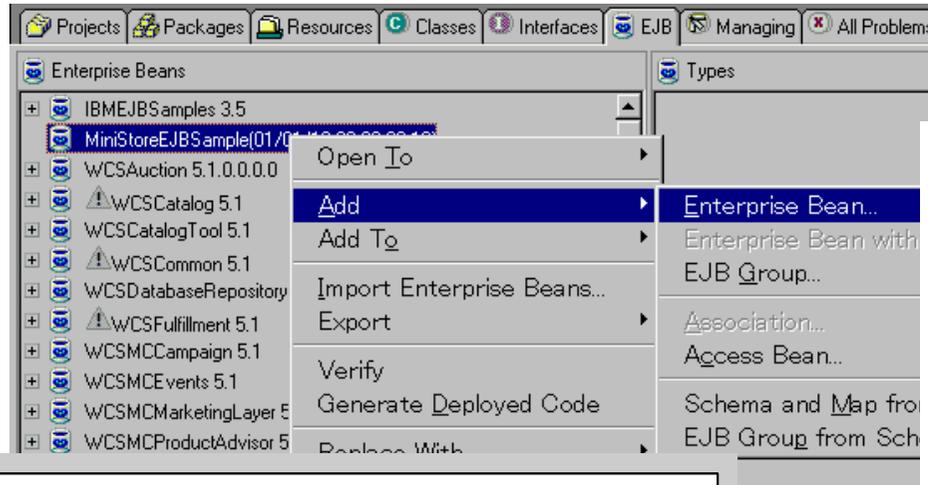
■ テーブルの作成

- アプリケーション定義に基づき、テーブルをデザイン、WCS DB上に作成

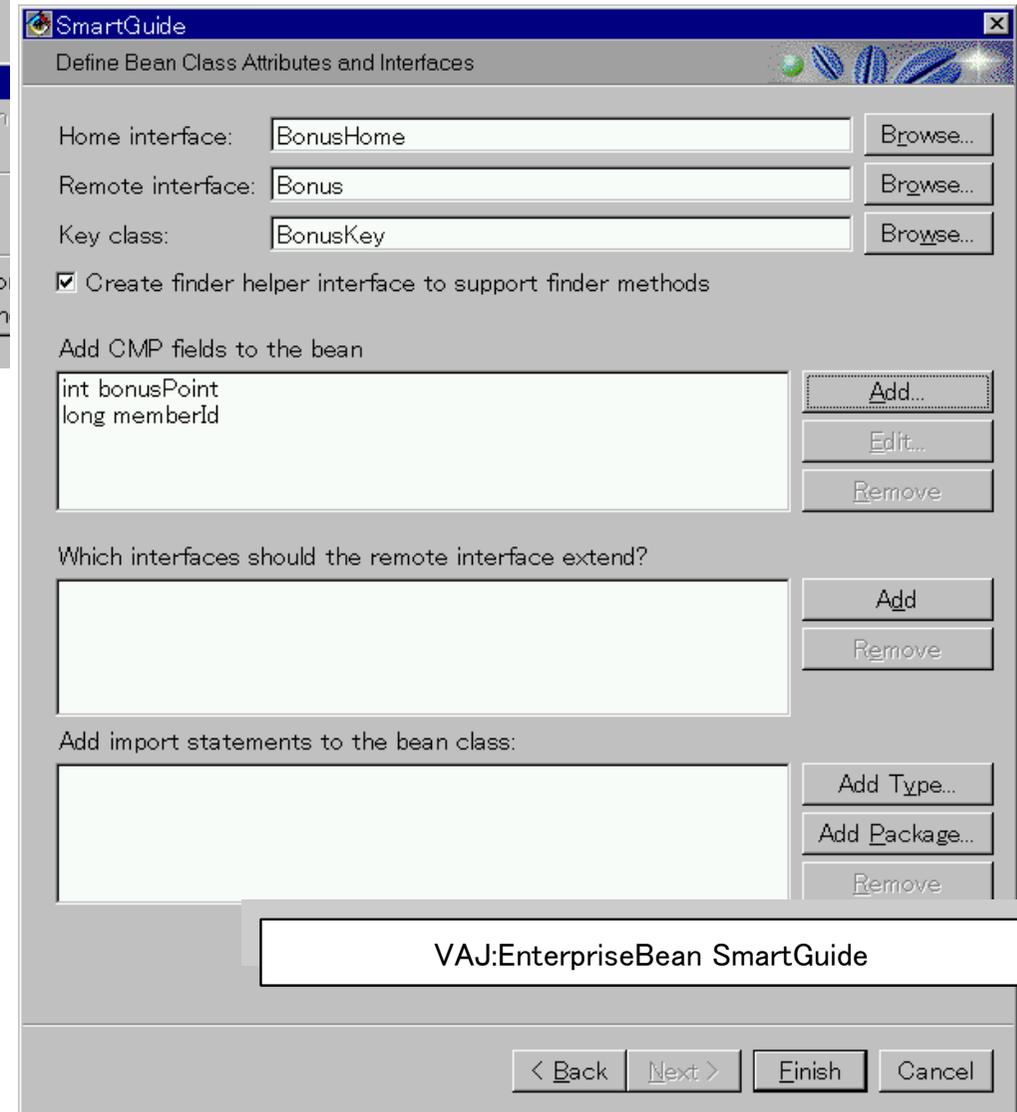
■ EntityBeanの作成

- VAJ Smart GuideからAdd → EntityBeanの作成を選択
- Bean TypeにCMP EntityBeanを選択し、CMP Fieldsにカラムに対応するFieldを追加する
- 一部コードを修正
 - finderHelper, Homeインターフェースなどを必要に応じ修正
- 表とEntityBeanのマップ
 - VAJ Schema BrowserおよびMap Browserを使用
- AccessBeanの作成
 - VAJからEntityBeanを選択、Add → AccessBean
- WAS環境へDeploy
 - VAJでJARファイルの形式でExport
 - WASのWebSphere Commerce ServerのClassPathを通す
 - WAS AdminClientからEnterpriseBeanを登録

EJBのカスタマイズ: サンプル画面 CMP Entity Beanの作成

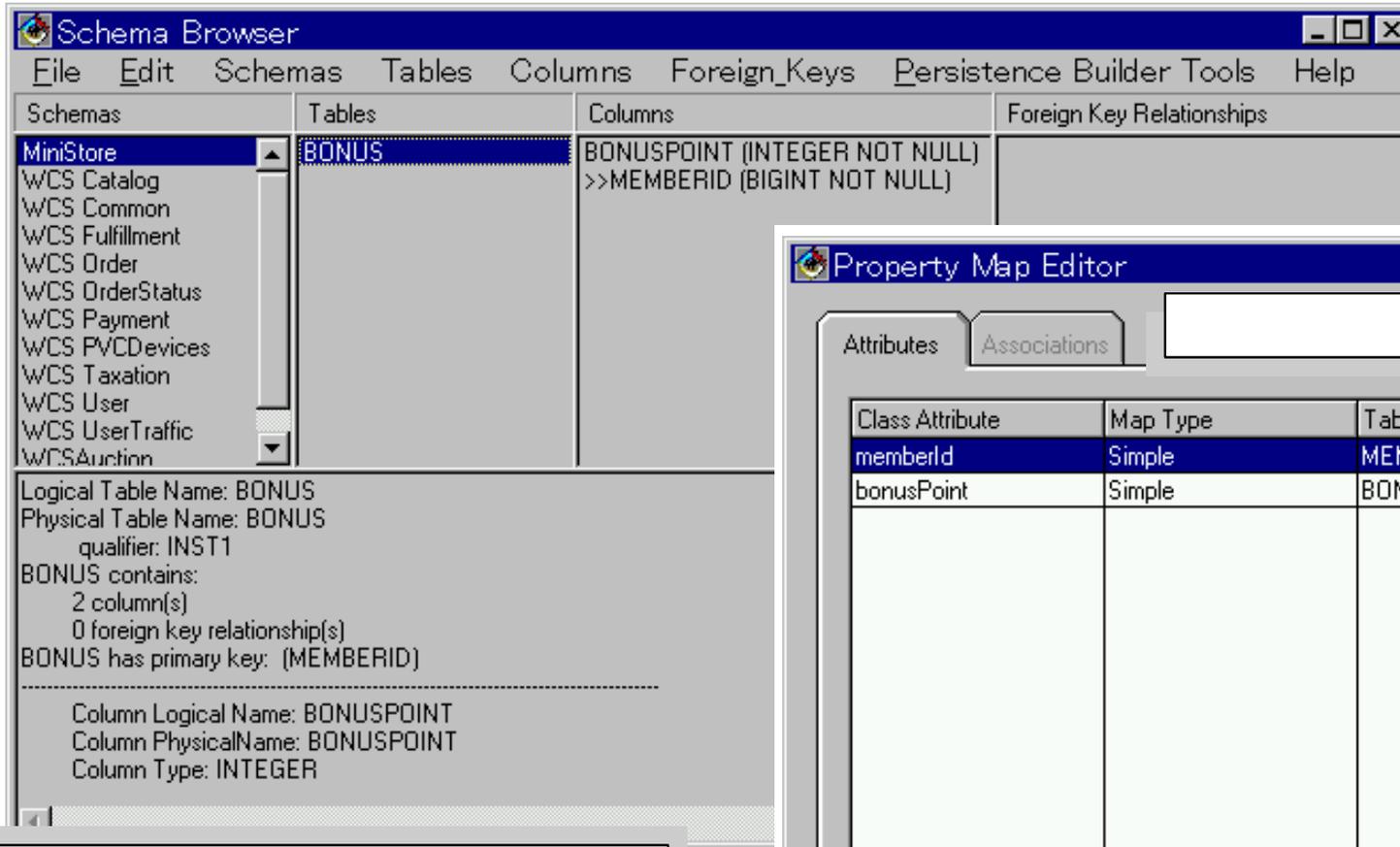


VAJ:EnterpriseBeanの作成

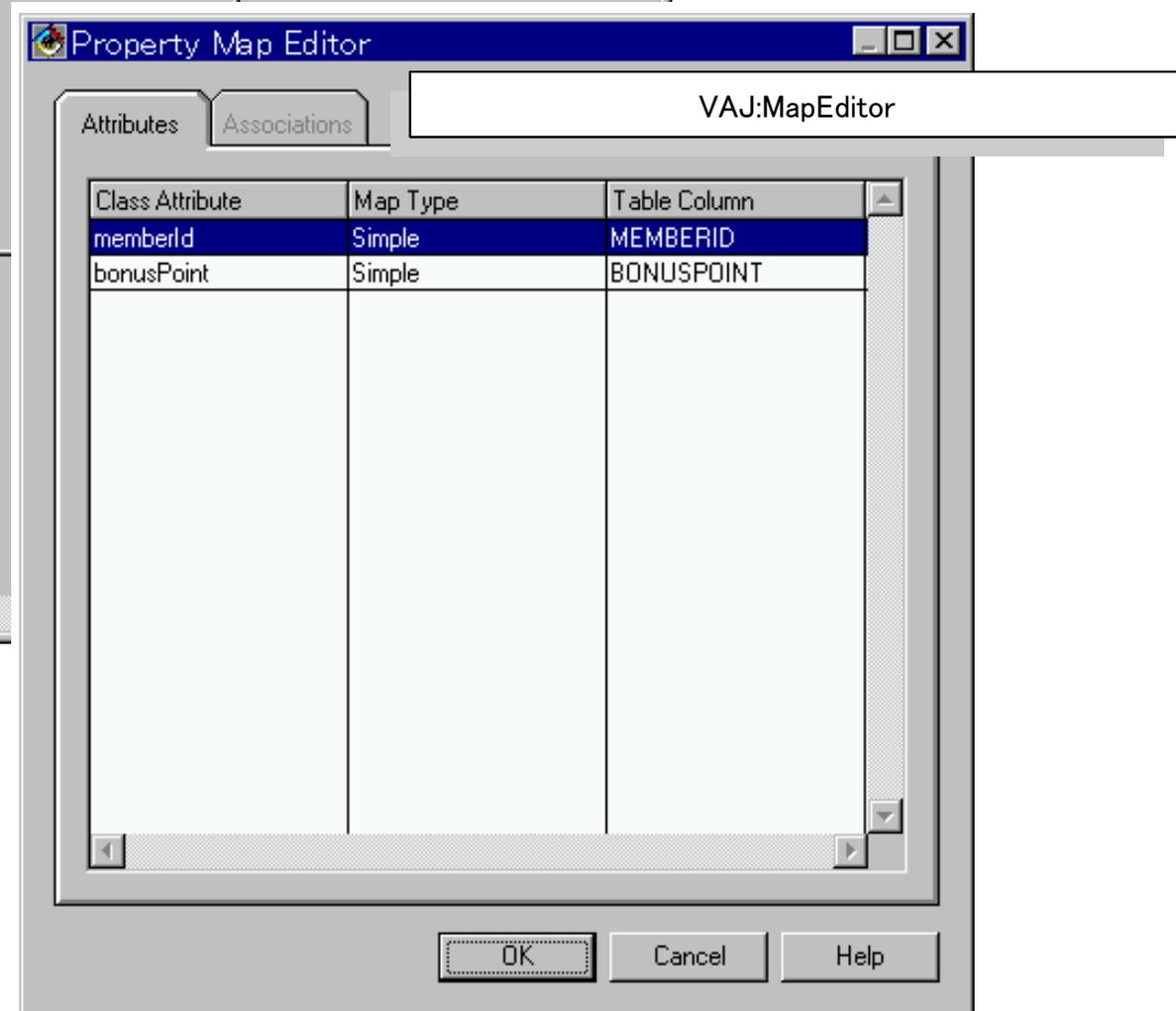


EJBのカスタマイズ: サンプル画面

Entity BeanとDBテーブルカラムのマッピング

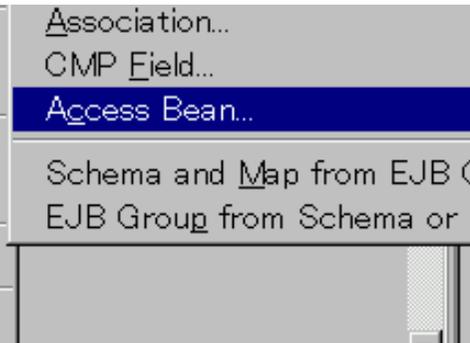
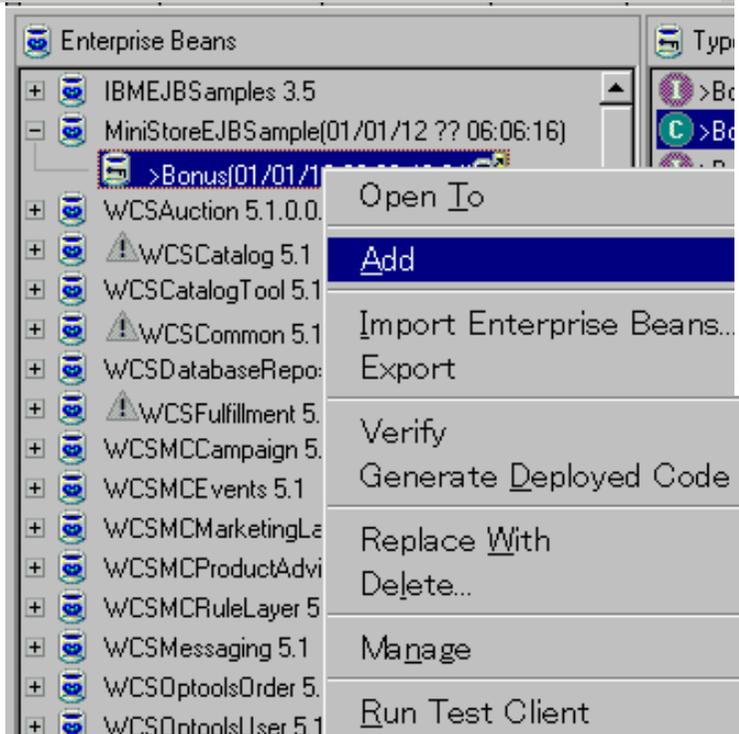


VAJ:SchemaBrowser



EJBのカスタマイズ: サンプル画面 AccessBeanの作成

VAJ:AccessBeanの作成



コマンドのカスタマイズ

▶ ControllerCommandの作成

- CommandRegistryに登録
 - URLREG、CMDREGに登録
- ControllerCommandの作成
 - URLREG、CMDREGの登録情報に基づき、
 - インターフェースcom.ibm.commerce.commands.ControllerCommandをextendした新規インターフェースを作成
 - クラスcom.ibm.commerce.commands.ControllerCommandImplクラスをextendした新規クラスを作成
 - performExcute()メソッド内のビジネスロジックを要件に応じカスタマイズ
 - 必要なデータベースアクセスは作成したEntityBeanのAccessBeanを介して行う
 - TaskCommandを呼び出す場合、コード内に記載

▶ TaskCommandの作成

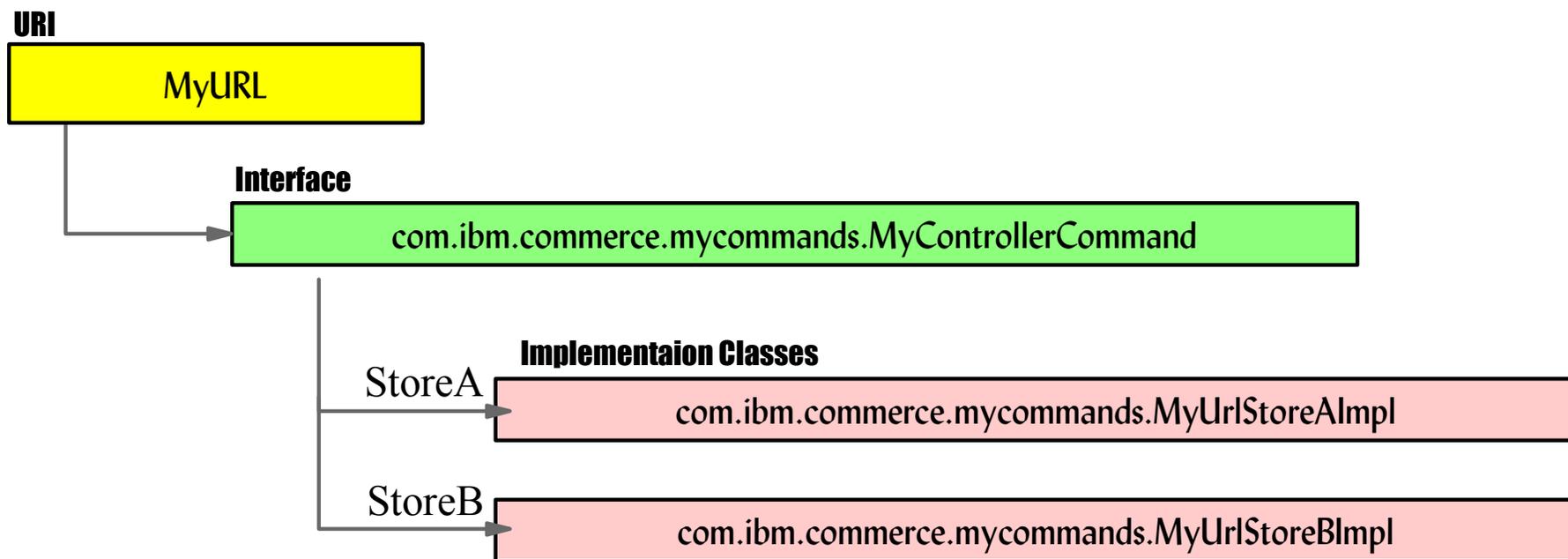
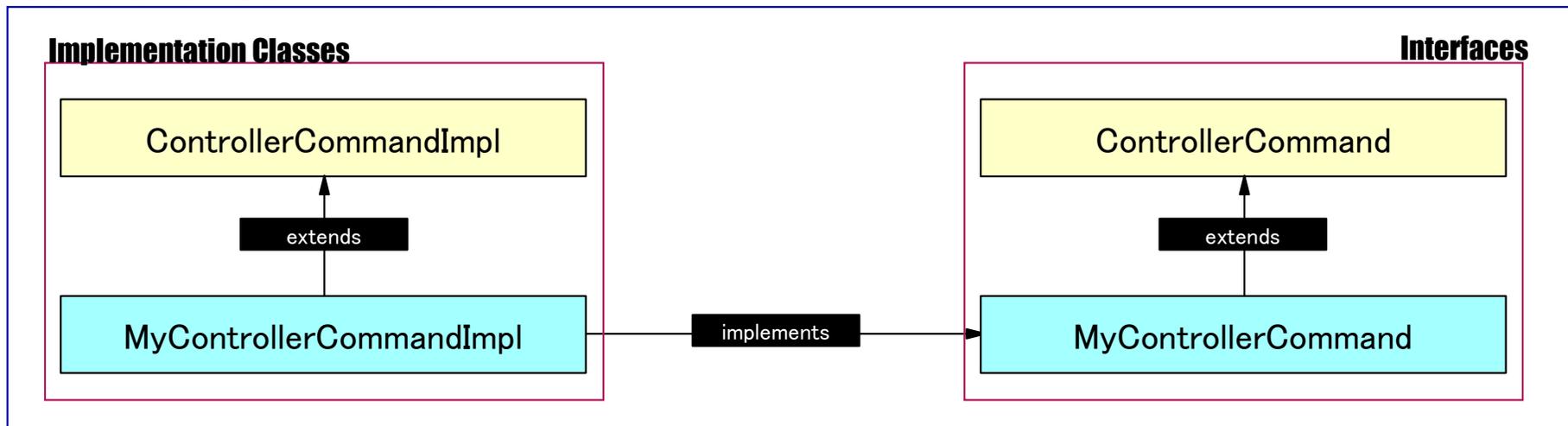
- TaskCommandの作成
 - 細分化できるビジネスタスクはTaskCommandとして分化
 - インターフェースcom.ibm.commerce.commands.TaskCommandをextendした新規インターフェースを作成
 - クラスcom.ibm.commerce.commands.TaskCommandImplクラスをextendした新規クラスを作成

▶ CommandのDeploy

- 作成したコードをJAR形式でWCS環境にExport
- WAS AdminClientよりClasspathを通し、ApplicationServerを再起動

コマンド構造

Controller Commands



カスタマイズの統合

▶ 各種カスタマイズを統合

- 新規ビジネスロジックを実現するためのURL Commandを作成
- URLコマンドを呼ぶべきJSPページを作成、FORMなどでURL Commandに渡すパラメーターを収集
- URL CommandとControllerCommandの関連をCommandRegistryに記載
 - URLREG/CMDREG
- 新規表の作成
- 新規作成表に関してはEntityBean、AccessBean、DataBeanを作成する
- ControllerCommandを作成し、ビジネスロジックを実行
 - 基本はDBの操作であるため、AccessBeanを介してDBの処理を行う
- 必要に応じてTaskCommandを呼び出す
- 細分化されたロジックをTaskCommand作成で実装
- 返すべきページのViewNameをControllerCommandに記載
- 結果表示用のJSPファイルを作成
 - DataBeanを介してDB情報を取得
- VIEWREGにViewNameとJSPとの関連を記載

参考資料

カスタマイズ時に参照すべき資料

▶ マニュアル

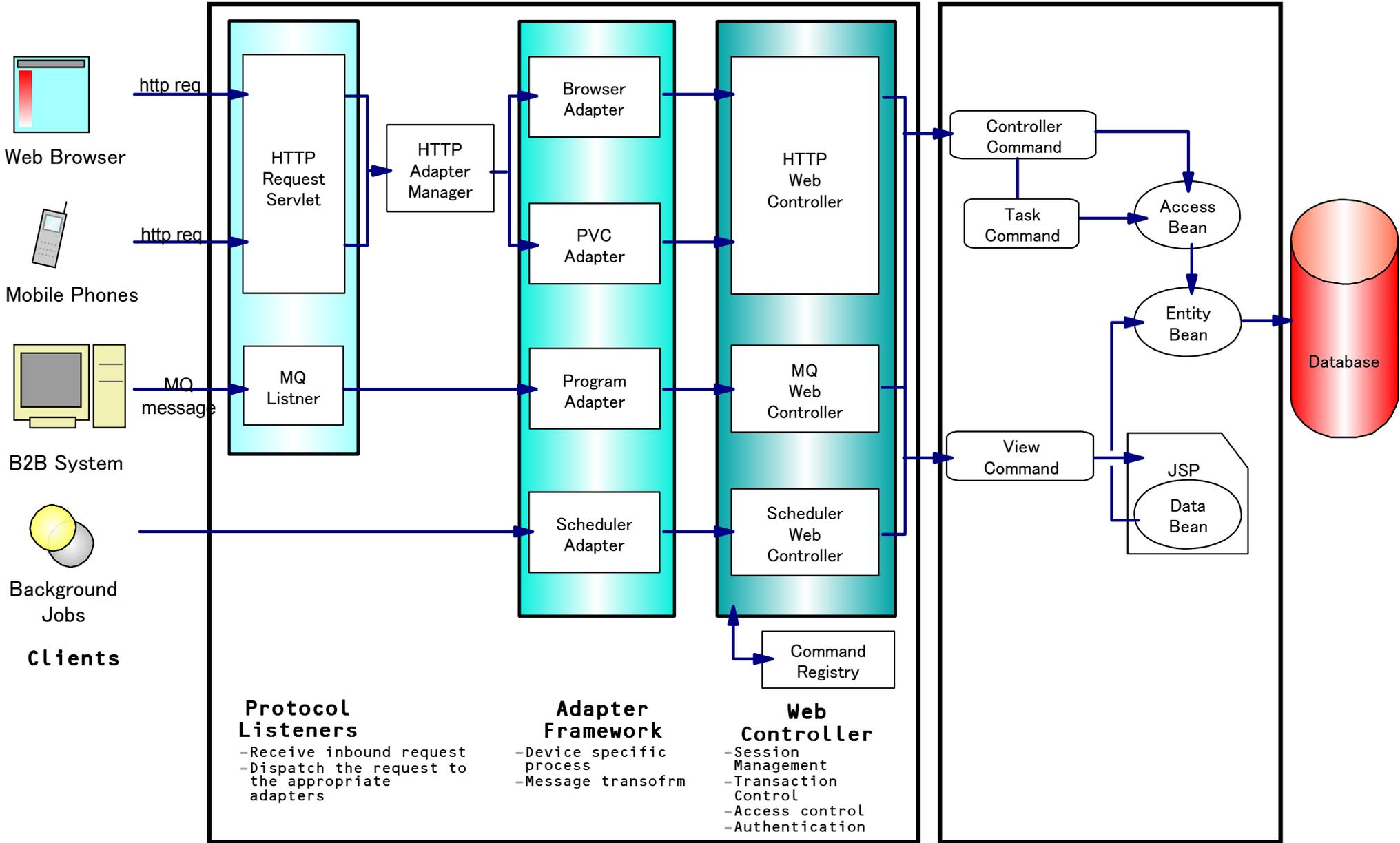
- WCS 5.1プログラマー・ガイド
 - 各コンポーネントの詳細説明、カスタマイズ方法
 - ロジックを作成するTutorialを提供
 - サンプルスケルトン・コードを含む

▶ オンラインヘルプ

- Referenceとして以下の資料を提供
 - Topic → Reference → Command
 - Topic → Reference → Database Schema
 - Topic → Reference → Sample Store

JavaCommand、URL Commandの詳細、JavaDoc
ObjectModel、DataModel、DBテーブル詳細
サンプルストアフロー、各ページ詳細

Server Components - Detail

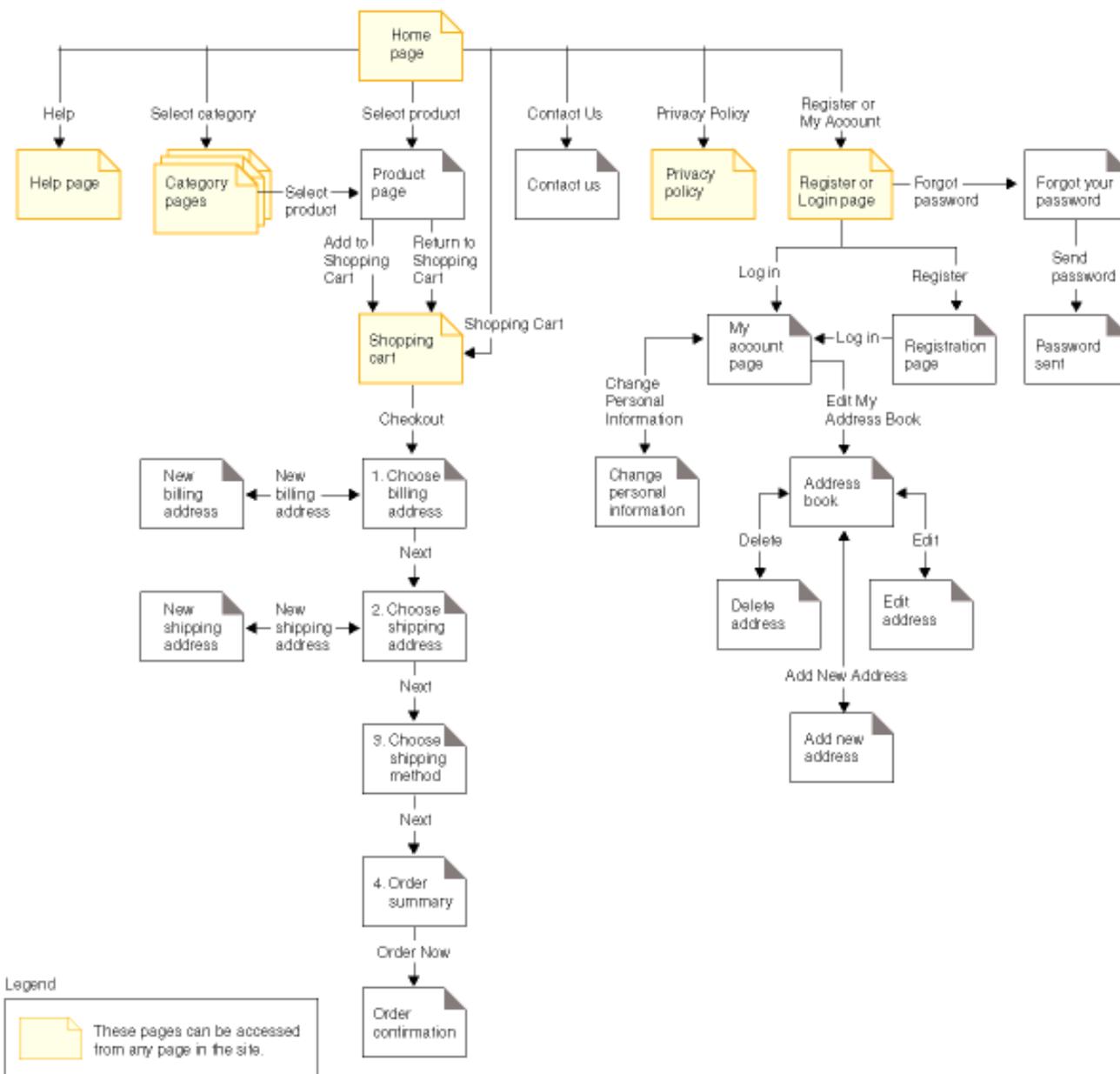


WebSphere Commerce Site V5.1

▶ WCS Servlets

- WebApplication (WCS Stores / WCS Tools) にはそれぞれ以下のServletが定義されている
- WCS Request Servlet
 - com.ibm.commerce.server.RequestServlet
 - RequestServletは処理ディスパッチ用のServlet、WCSユニーク
 - WebPath: default_host/webapp/wcs/stores/servlet
- WCS JSP Compiler
 - com.sun.jsp.runtime.JspServlet
 - JSP1.0用のコンパイラ、WCS以外でも使用される
 - WebPath: default_host/webapp/wcs/stores/*.jsp
- WCS File Serving Servlet
 - com.ibm.servlet.engine.webapp.SimpleFileServlet
 - AppServerを介してHTMLやGIFなどを取得するためのServlet、WCS以外でも使用される
 - WebPath: default_host/webapp/wcs/stores

WCS Sample Store (InFashion) Flow



WebSphere Commerce Suite V5.1



Take e-commerce to the next level—

fast flexible
fearless

New WebSphere Commerce Suite,
Version 5.1