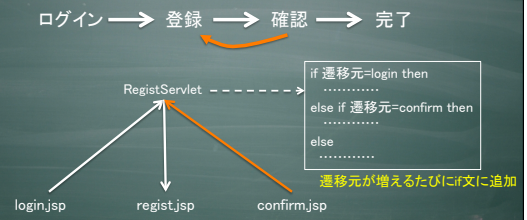
高度情報処理演習B

アーキテクチャ－MVC

ここまでの住所登録システム

- Servlet + JSPの導入
  - ビジネスロジックはServlet
  - 画面(ビュー)はJSP
- これで十分か？
  - リンクの追加や変更に関する問題
  - 仕様変更に関する問題
  - 実装方式/品質のバラつき問題
    - 大規模開発では致命的

リンクの追加や変更に関する問題



仕様変更に関する問題

- セッション管理を追加
  - 従来ならそれぞれの画面にプログラムを追加
  - 専用のメソッドを作って呼び出す
  - すべて開発者の責任
- 画面を追加したい
  - すべて新規作成
    - データの取得，確認，ロジックや画面(JSP)の作成

実装方式/品質のバラつき問題

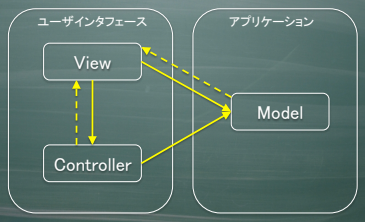
- 各画面の品質，実装方式が開発者に依存
  - 運用・保守で理解が困難
    - どうやって動かそうとしていたかを理解しなければならない
  - 開発者によって完成度が異なる
    - スキルの有無
    - 規約で決めても守られる保証はない
  - 変更時に参照すべきコードが不明
    - 最悪はコードを読み解く必要がある

ソフトウェアアーキテクチャ

- アーキテクチャ
  - 構造 (e.x., 建築物)
- ソフトウェア開発では
  - 責務分散モデル
  - 抽象化と問題の分割によって複雑性を減らすこと
  - オブジェクト同士の協調関係で処理を実行
  - Ex. BCE(Boundary-Control-Entity)

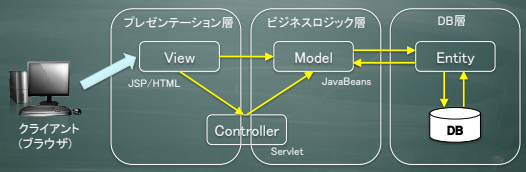
MVCアーキテクチャ

- 対話型アプリケーション

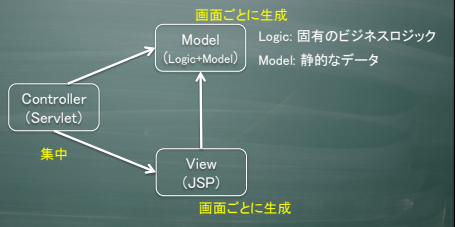


MVC2 (J2EE-MVC)

- Webアプリケーションのスタンダード

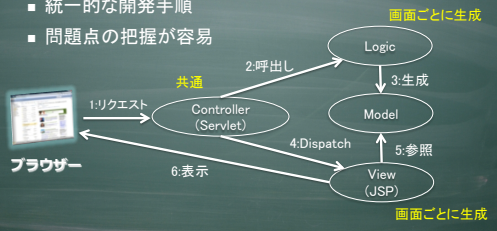


例題システムにMVCを適用

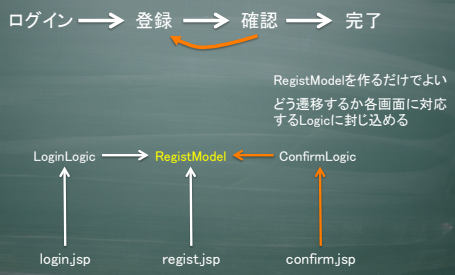


処理の流れ

- 必ずこの手順で処理する
  - 統一的な開発手順
  - 問題点の把握が容易

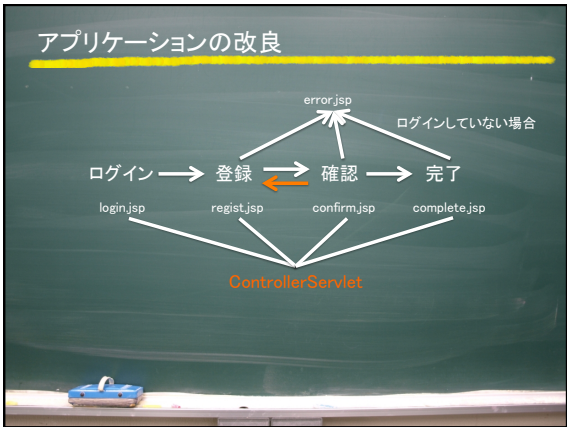


リンク構造を改良



問題点の解決

- リンクの追加や変更に関する問題
  - リンク構造の改良
- 仕様変更に関する問題
  - MVCアーキテクチャの導入
  - 何をどこで行うかを明確化
- 実装方式/品質のバラつき問題
  - MVCアーキテクチャによる処理手順の統一化
  - オブジェクト指向による差分プログラミング
    - 難しい処理を隠蔽する


アプリケーションの改良
error.jsp
ログインしていない場合
ログイン
登録
確認
完了
login.jsp
regist.jsp
confirm.jsp
complete.jsp
ControllerServlet