Seasar Conference 2006 Autumn

GUIプレゼンテーションフレームワーク

S2JFace

~なぜ、あなたはWebアプリケーションを作り続けるのか~

2006.11.12

エスエムジー株式会社　小森　裕介（komori@smg.co.jp）

株式会社グルージェント　亀谷　大樹（kameya@gluegent.com）

はじめまして！

- 名前：小森　裕介
- Blog：http://d.hatena.ne.jp/y-komori/
- 所属：エスエムジー株式会社（http://www.smg.co.jp）
- 主な仕事：
  - Javaによる集中監視制御システム設計・開発
  - 教育・各種執筆活動
- Seasar2とのかかわり
  - S2JFaceコミッタ、S2Containerコミッタ、S2JMSコミッタ

- 名前：亀谷　大樹
- Blog：http://d.hatena.ne.jp/dkameya/
- 所属：株式会社グルージェント（http://www.gluegent.com）
- 主な仕事：
  - OSS開発
- Seasar2とのかかわり
  - S2JFaceコミッタ

はじめに

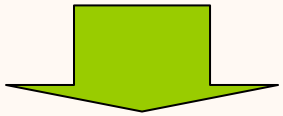
- Webアプリケーション開発でこんな思い、したことありませんか？

『え～！ なんでこんな複雑なGUI作らなきゃいけないの？』
『**Ajax**を広めた**G○○gle**なんか、大嫌いだ！』
『･･･そもそも、このシステム**Web**でやる意味あるの？』

そんな悩めるJava**エンジニアに福音です！**

Webアプリケーション開発の問題点

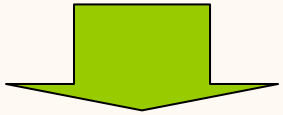
- Thinクライアントであるがために複雑な画面がつくりにくい
  - Ajaxの普及で技術的には可能性が増えた
- 案件の多くを占める業務システムは「Web」である必要性がない
  - 多くがイントラネット内で利用されるため、利用者が限られている
  - 「Web」である縛りが画面デザインの自由度を下げ、開発の難易度を上げている！
  - Webが必要なのは「クライアント管理が不要」だから
  - 今ならば JavaWebStart など、代替技術がある



なぜ、あなたは**Web**アプリケーションを作り続けるのか？

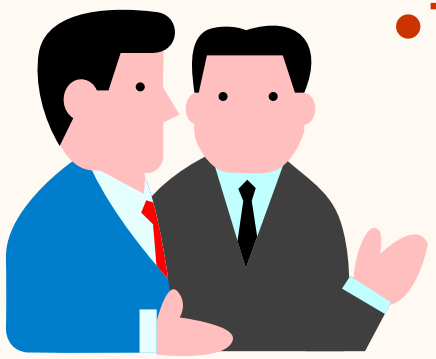
GUIアプリケーション開発の問題点

- (Javaによる)適当なフレームワークが存在しない
- 画面を作るのが面倒
  - GUIエディタも存在するが(特にJavaでは)使いやすいものがない
- 画面デザインとロジック開発の分担がしにくい
  - プログラムがわからないと画面がつくれない
    (HTMLならプログラムがわからなくても画面がデザインできる)



開発手法がある程度確立している
Webアプリケーションとして作らざるをえない

システム開発における作業分担



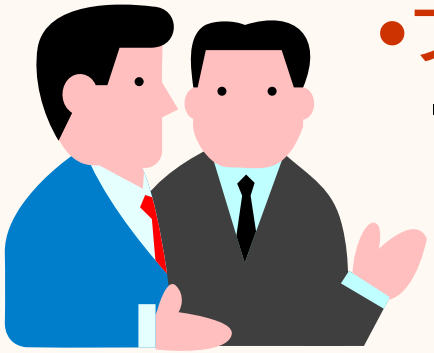
- フロントSE
  - 顧客と話し合って要件＆画面仕様を詰める
  - コーディングはしない
  - **HTML**モックなら書ける？
  - 画面仕様は**Word**や**Excel**で書くことが多い

大規模なシステムほど分担がはっきりする

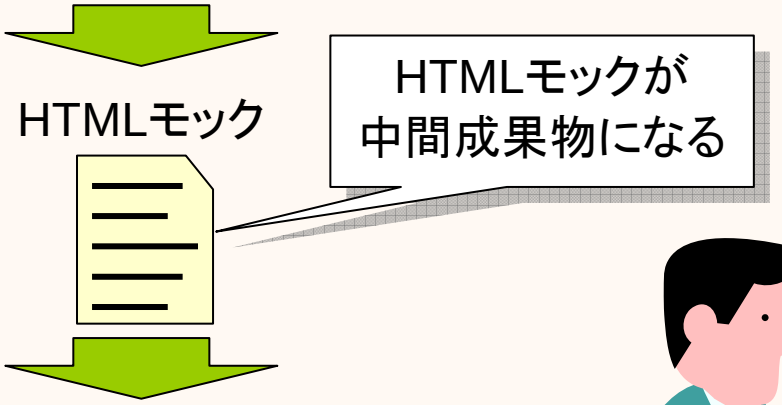
- 開発者
  - 画面仕様に沿って実際の画面を作成
  - 複雑な画面仕様に頭を抱える
  - バリバリコーディング！

プレゼンテーションとロジックの分離

- S2JSF(Teeda)におけるプレゼンテーションとロジックの分離



- フロントSE
  - **HTML**モックを作成する

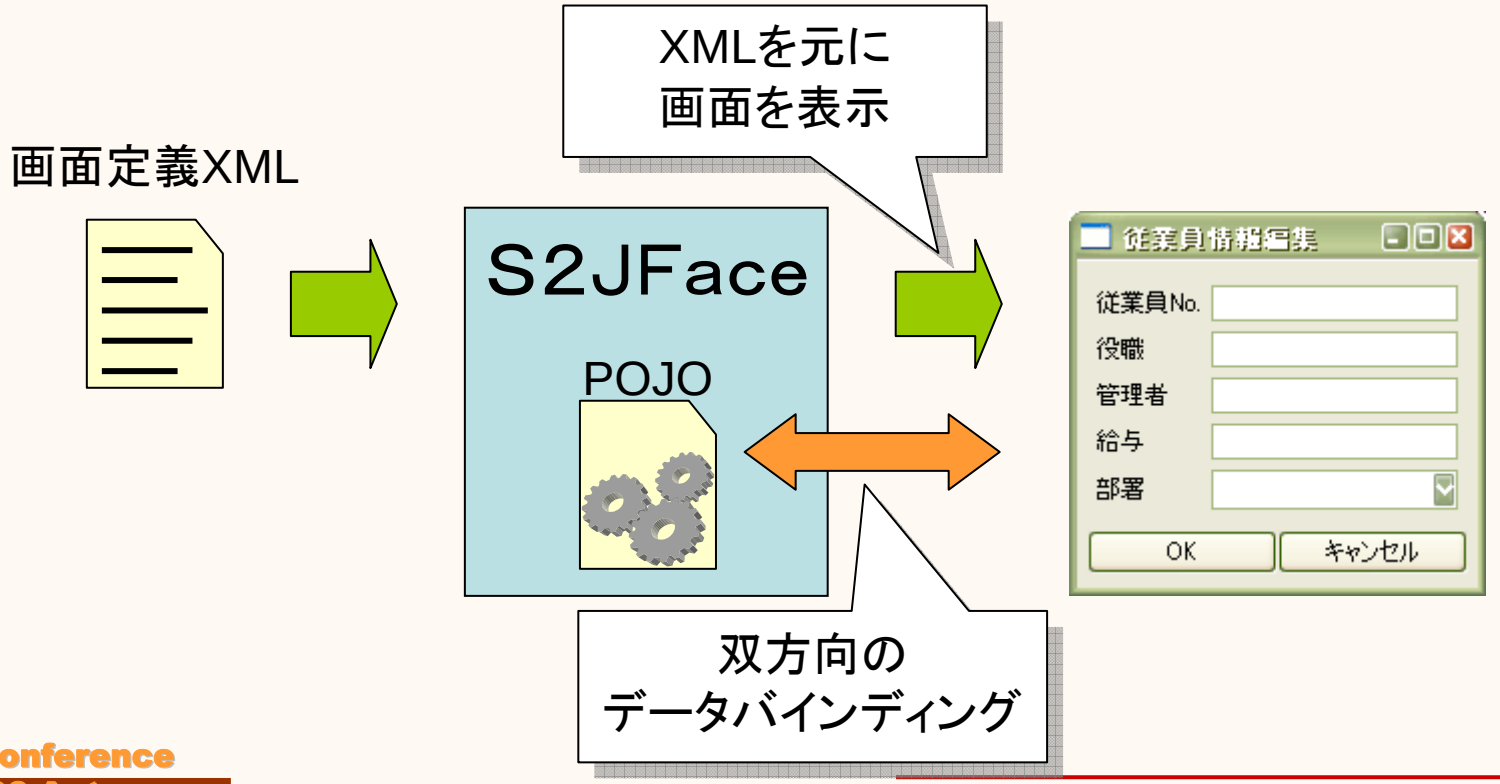


- 開発者
  - **HTML**モックを元に実際の動きをつける

GUIの世界でもこのような分離ができるようにしたい

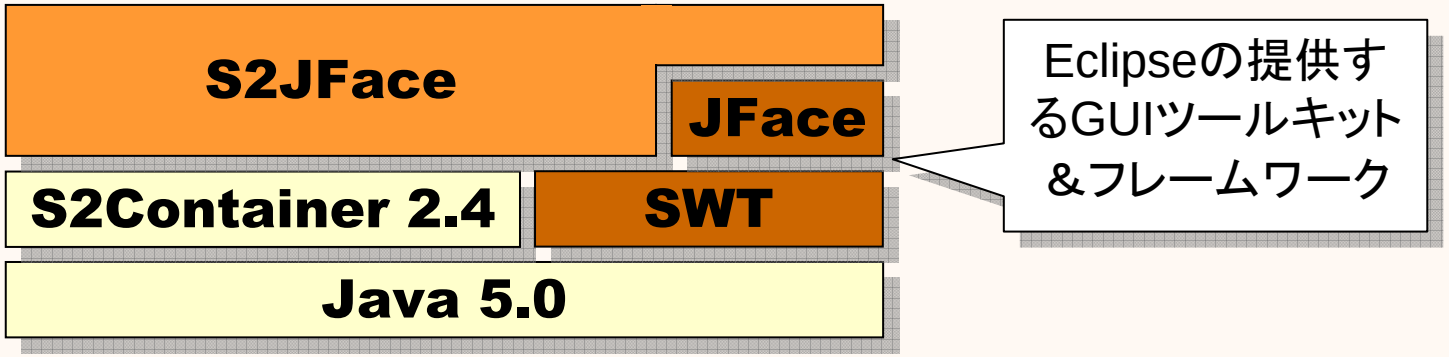
S2JFaceとは

- XMLによる画面作成を可能としたGUIフレームワーク
- S2JSFライクな画面とデータのバインディングをサポート
- もちろんロジックはPOJOでOK

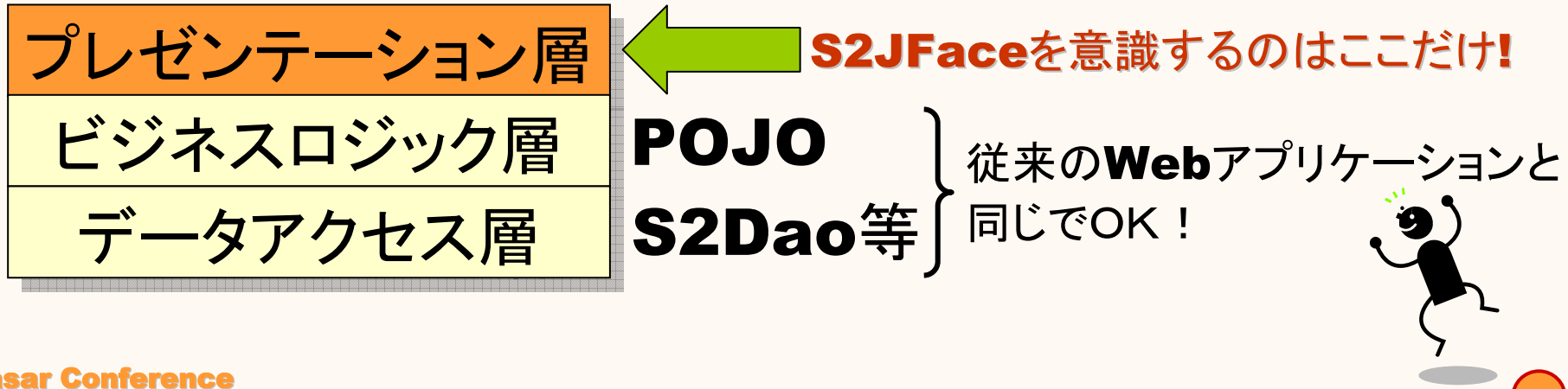


S2JFaceのアーキテクチャ

- ソフトウエア構成



- 論理構成



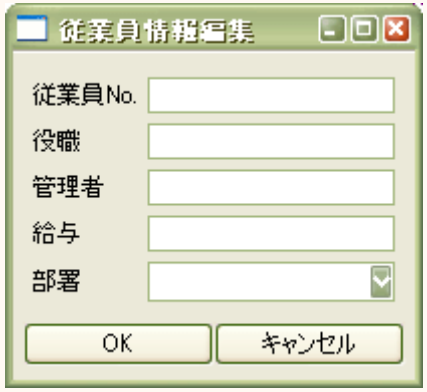
なぜSWT & JFace？

- **見た目**
  - Windowsネイティブのコンポーネントを利用しているため、画面上の違和感がない
- **速さ**
  - Swingに見られるもたつきが少ない
- **実績**
  - Eclipseで培われた実績
  - JFaceによる開発のしやすさ
- **プラットフォーム・ポータビリティ**
  - Swingに比べればポータビリティは多少下がる
  - 業務アプリケーションクライアントの大半がWindowsである現状を考えるとほとんど問題にならない

S2JFaceでの画面定義

- 独自のXML形式で画面を記述
- HTMLに似た構文のため、覚えやすい
- XMLには画面デザイン以外の情報を持たせないのがポリシー

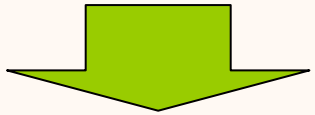
```
<?xml version="1.0" encoding="UTF-8"?>
<template xmlns="http://s2jface.sandbox.seasar.org">
  <window title="従業員情報編集" width="400" height="300">
    <gridLayout numColumns="1">
      <gridData horizontalAlignment="FILL" />
    </gridLayout>

    <composite>
      <gridLayout numColumns="2">
        <gridData horizontalAlignment="FILL" />
      </gridLayout>

      <label text="従業員No." />
      <text id="empNo" />

      <label text="役職" />
      <text id="job" />

      <label text="管理者" />
      <text id="manager" />

      <label text="給与" />
      <text id="salary" />

      <label text="部署" />
      <combo id="department" style="DROP_DOWN, READ_ONLY" />
    </composite>

    <composite>
      <fillLayout type="HORIZONTAL" />
      <button id="ok" text="OK“ />
      <button id="cancel" text="キャンセル“ />
    </composite>
  </window>
</template>
```



- **Eclipse-WTP**付属の**XML**エディタで補完しながら記述が可能
- 将来的には**GUI**エディタも提供予定

GUIエディタ vs テキストエディタ

- GUIエディタ
  - ☺ 直感的な操作で簡単に画面が作れる
  - ☹ 自動化しにくいので画面の大量生産には不向き
- テキスト(XML)エディタ
  - ☹ タグを覚えなければならず、敷居が高い
  - ☺ 独自ツールを作成すれば画面作成を自動化しやすい

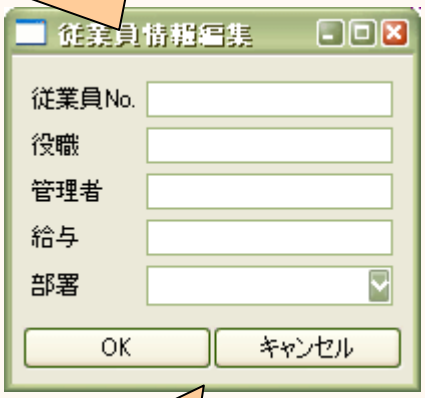


S2JFaceはテキストエディタによる画面作成と
GUIエディタによる画面作成の両方をサポート(予定)

S2JFaceでの画面とクラスの関係

- 1画面につき1個のJavaクラス（コンポーネント）が対応
- 「*画面名称*+Action」が命名規則
- いわゆる「ページ指向」のフレームワーク

「EmployeeEdit」
という名前の画面



S2JFace独自の
XMLで記述

EmployeeEditAction

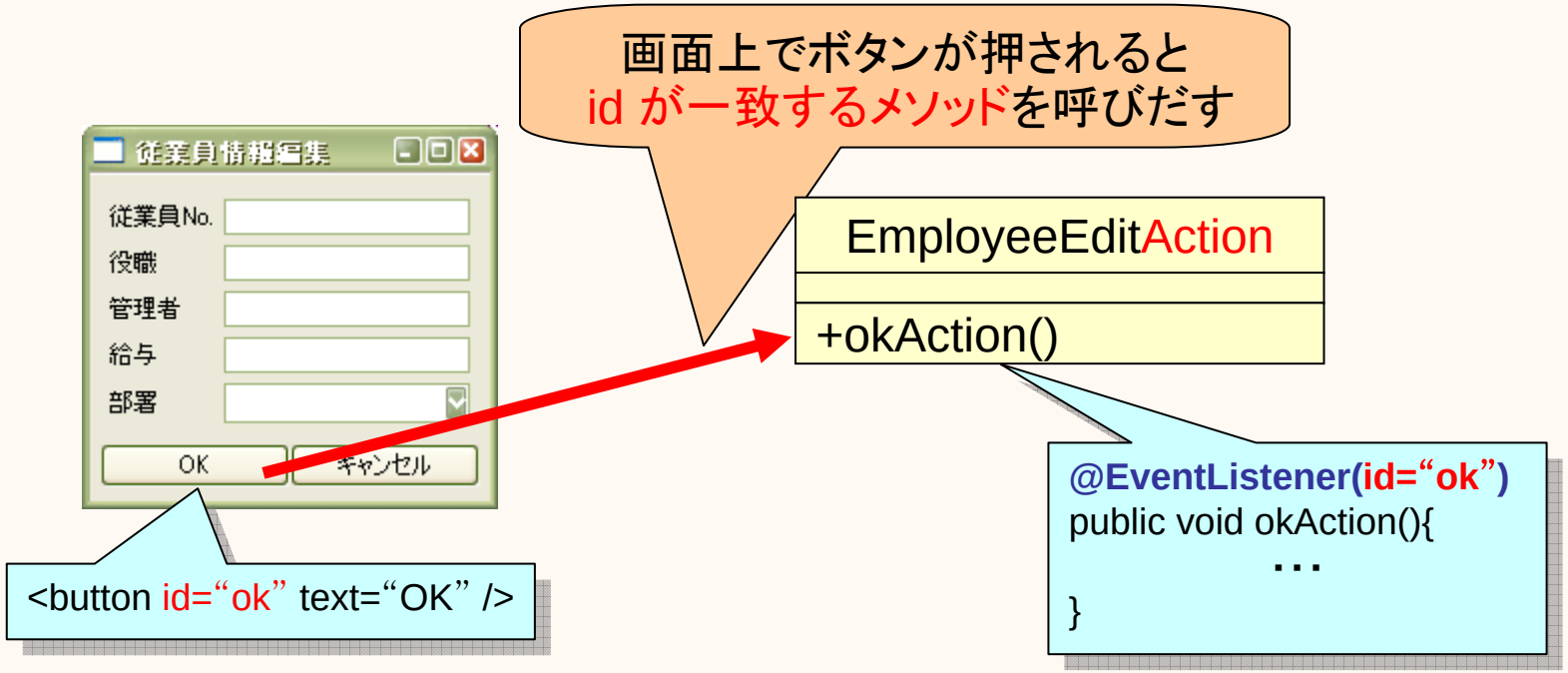
アクションクラスは
S2Containerへ登録しておく

```
<component name="EmployeeEditAction"
class="org.seasar.jface.example.employee.EmployeeEditAction" />
```

Methodバインディング（@EventListener）

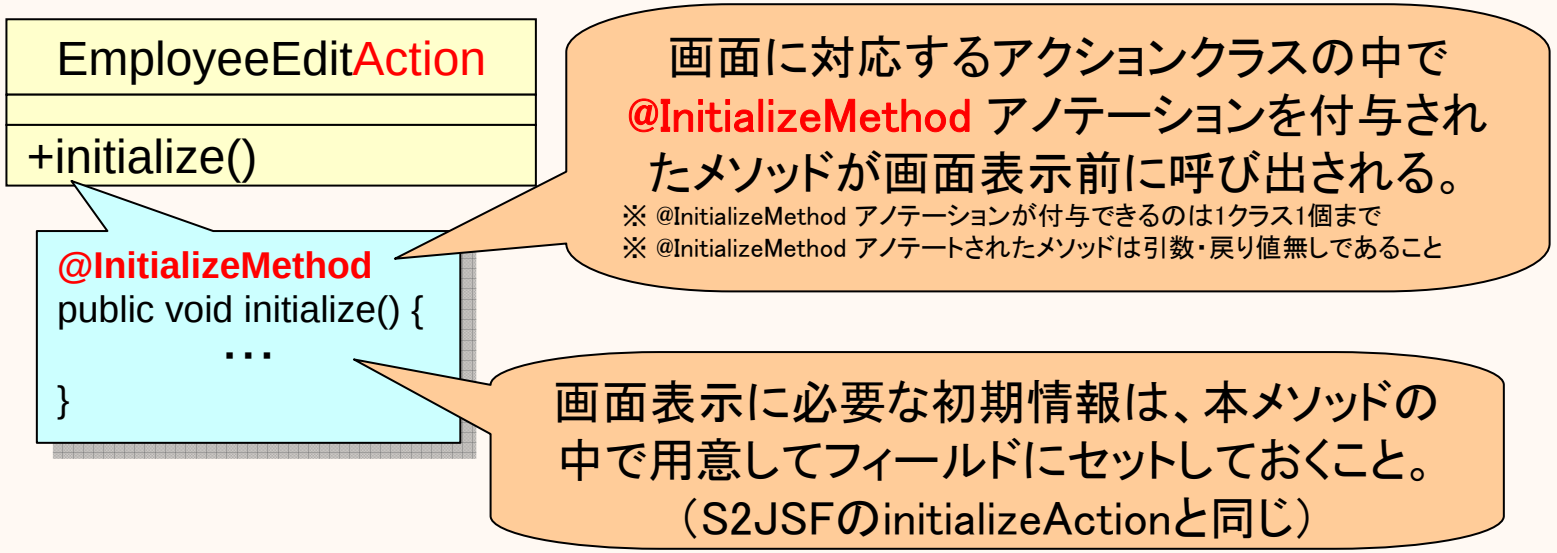
- 画面上のイベントに対して、アクションクラスのメソッドを呼び出せる
- イベントとメソッドの関係はアノテーションで記述

  → Listenerクラスを作成する面倒なコーディングは不要！



画面の初期化（@InitializeMethod）

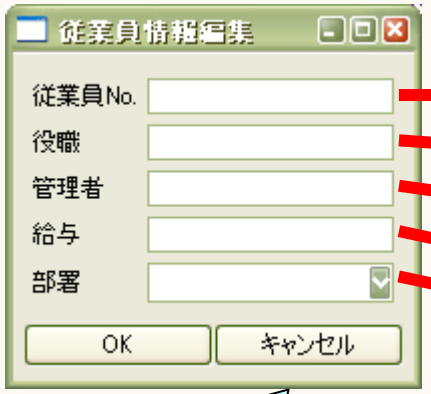
- 画面初期化時に呼び出されるメソッドもアノテーションで指定



Widgetインジェクション

- アクションクラスに対して、画面上のwidgetを自動的にインジェクションできる
  → widgetオブジェクトの取り回しを考える面倒から解放！

XML上の**id**とアクションクラスの
フィールド名が一致すれば
自動的に設定（setterメソッドは不要）



EmployeeEditAction
-Text empNo
-Text job
-Text manager
-Text salary
-Combo department
+okAction()

フィールドはWidgetの
サブクラスであること

```
<text id="empNo" />
<text id="job" />
<text id="manager" />
<text id="salary" />
<combo id="department" />
```

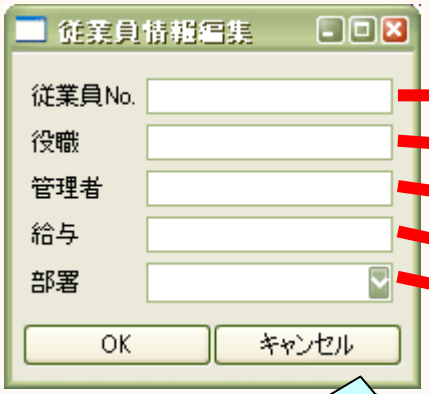
```
@EventListener(id="ok")
public void okAction(){
  String job = job.getText();
}
```

呼び出された
メソッドの中で
すぐに利用可能

Valueバインディング（値の取り出し）

- テキストフィールドなどの入力系コントロールへ
  入力された値を、アクションクラスへ直接バインド

@BindingValue(type=BindType.Import)
アノテーションをフィールドに付与すると
フィールドの型に入力値を変換してバインド



EmployeeEditAction
-int empNo
-String job
-String manager
-String salary
-int department
+okAction()

```
@BindingValue(id="empNo",
type=BindType.Import)
private int empNo;
@BindingValue(type=BindType.Import)
private String job;
...
```

Idとフィールド名が一致していれば
id引数は省略可能

```
<text id="empNo" />
<text id="job" />
<text id="manager" />
<text id="salary" />
<combo id="department" />
```

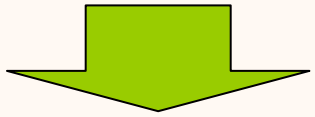
Valueバインディング（値の設定）

- アクションクラスのフィールドからテキストフィールドなどの入力系コントロールへ値を直接バインド

画面表示時に@BindingValue(type=BindType.Export)
アノテートされたフィールドの値をテキストフィールドへ設定

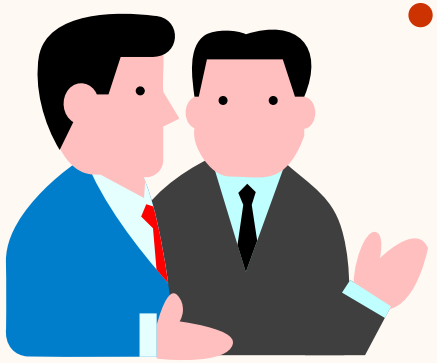
WidgetインジェクションとValueバインディングの使い分け

- Widgetインジェクション
  - ☹SWTのWidgetを直接意識する必要がある
  - ☺自由な画面操作が可能
- Valueバインディング
  - ☺SWTのWidgetを意識せずデータのみを考えれば良い
  - ☺バリデーションが可能
  - ☹細かな画面操作は不可能

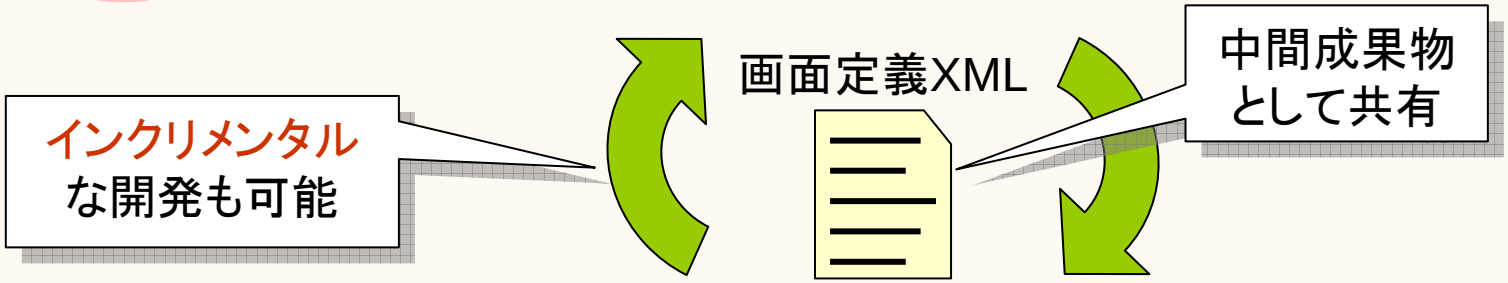


GUIの見た目を変更する処理が必要な場合は**Widget**インジェクション
データのみを扱う場合は**Value**バインディングを利用する

S2JFaceを利用した開発モデル



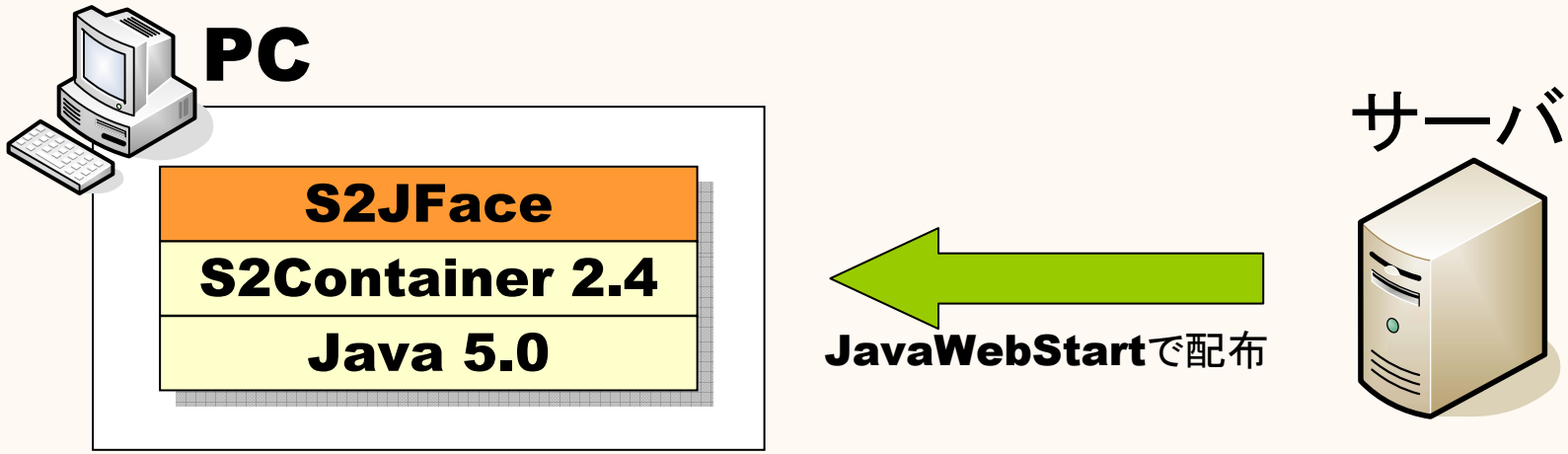
- フロントSE
  - 顧客と打ち合わせをしながらGUIエディタで画面モック作成
  - 慣れている人はXMLエディタで直接作成も可能
  - 基本的にロジックを意識する必要なし
  - その場でデモンストレーションも可能



- 開発者
  - 画面定義XMLにIDを追加
  - プレゼンテーション層以下を実際に作成

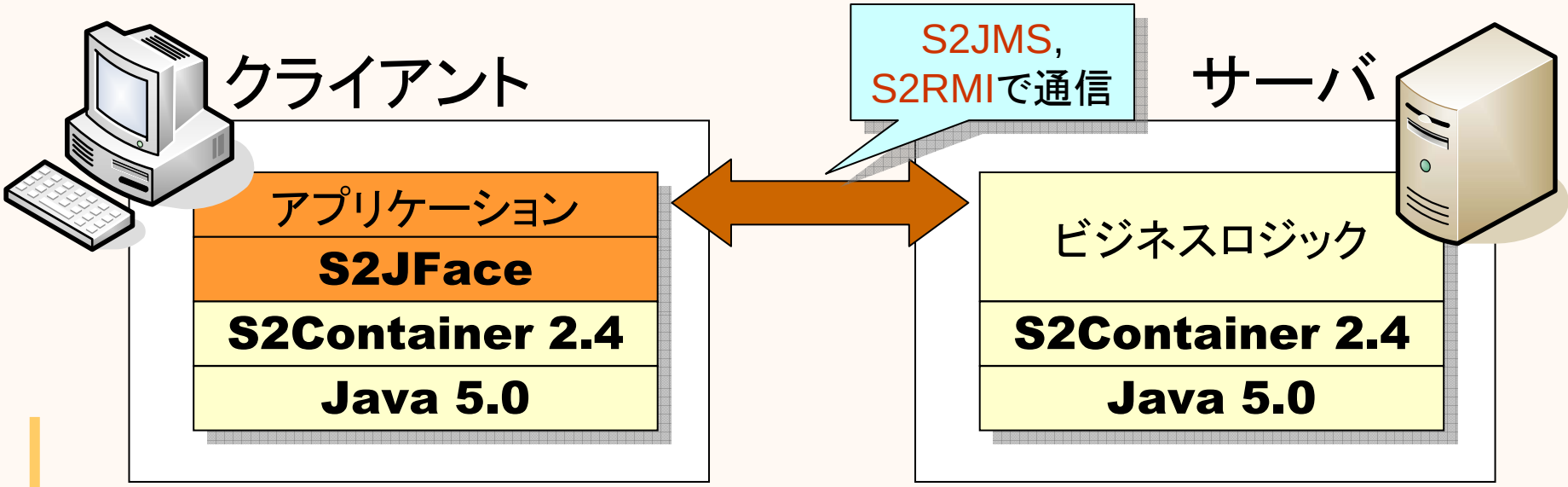
S2JFaceの利用シーン(1)

- 単体アプリケーションのフレームワークとして
  - –最もシンプルな使い方
  - –個人レベルで作成するアプリケーションやツールの開発も可能
  - –JavaWebStartを利用してサーバからの一括配布も可能



S2JFaceの利用シーン(2)

- クライアントアプリケーションとして
  - 業務アプリケーションで最も多い利用が予想される
  - クライアント側はプレゼンテーション層までを実装
  - サーバ側にロジック層以下を実装
  - サーバ側にはEJBを利用することも可能？



S2JFaceのまとめ

- XMLとGUIエディタの2Way開発
  - 敷居の低いGUIエディタ
  - 大量生産に適したXML編集
- GUIにおける画面とロジックの完全分離
- 強力なデータバインディング
- Webアプリケーションシステムで作成した
  バックエンドがそのまま利用可能
- JavaでGUI開発を望む人にとっては福音(^^;

今後の予定

- **機能面**
  - バリューバインディング機能の向上
  - バリデーション機能の提供
  - 画面定義XMLの簡素化(継承、コンポーネント機能、etc…)
  - GUI画面エディタの提供
- **非機能面**
  - 品質向上(テスト網羅性の向上)
  - サンプルアプリケーションの充実
  - ドキュメント＆サイト整備

S2JFaceのリソース

- **Webサイト**
  - http://s2jface.sandbox.seasar.org/ja/
- **コミッタのブログ**
  - こもりん日記
    - http://d.hatena.ne.jp/y-komori/
  - 大樹の色々日記
    - http://d.hatena.ne.jp/dkameya/
  - 衛星黒猫迷想記
    - http://d.hatena.ne.jp/bskuroneko/

最後に

S2JFaceは現在
Sandboxプロジェクトにて
鋭意開発中です

みなさまからのご意見、
ご要望をお待ちしております

Seasar Conference 2006 Autumn

GUIプレゼンテーションフレームワーク

S2JFace

~なぜ、あなたはWebアプリケーションを作り続けるのか~

2006.11.12

エスエムジー株式会社　小森　裕介（komori@smg.co.jp）

株式会社グルージェント　亀谷　大樹（kameya@gluegent.com）