

# Seasar Conference 2007 Autumn



## Ymir

～Wiki感覚でWebアプリケーションを作ろう～

Ymirプロジェクト プロジェクトリーダー

(株)アークシステム

横田 健彦 (a.k.a. Skirnir)



- 横田 健彦 (よこた たけひこ)
  - ネット上ではSkirnir(すきーるにる)と名乗っています
  - (株)アークシステム所属
    - Javaや.NETのWeb系技術者募集中！
  - プラグイン機構がウリのCMS「Kvasir/Sora」を開発
    - 2006年度上期IPA未踏ソフトウェア創造事業に採択
  - 北欧神話好き
    - Skirnir、Kvasir、Ymirも北欧神話のキャラクターの名前です
  - 阪神タイガース好き
    - 今年は何も言うことはありません...(涙)



- なぜYmirを作ったのか
- Ymirとは
- しくみ
- 何がうれしいか
- 他にできること
- 開発デモ



- 既存の (Java製) フレームワークを使ったWebアプリ開発は面倒
  1. たった1画面足したいだけでもやらないといけないことがたくさんある
  2. URIとページテンプレートパスの対応が直感的でない
  3. URIを自由に決められない



# Strutsの場合

# 1. たった1画面足したいだけでもやらないといけないことがたくさんある

- struts-config.xmlに
  - <form-bean>追加
  - <action>追加
- ActionFormクラス追加
- Actionクラス追加
- validator.xmlにもエントリ追加
- ...



## 2. URIとページテンプレートパスの対応が直感的でない

### – URL

/view-form.do

### – テンプレートパス

/WEB-INF/jsp/view-form.jsp

### 3. URIを自由に決められない

- カテゴリ「コンピュータ」に属する記事で、2007年2月28日に投稿されたもののうち最初のもものを表示
- こんな感じで
  - /category/computer/20070228/0001.html
- 現実には
  - /view.do?category=computer&date=20070228&seq=0001
  - /execute/view?category=computer&date=20070228&seq=0001





やってられない



Webアプリ作るのって、  
もっと簡単にできるんじゃないの？



というわけで

- とにかく簡単にWebアプリを作れるようにする
  - 画面を増やす場合の手間を少なく
  - 変更に対するフィードバックが即座に得られるように (STEP by STEP開発、by ひがさん)
  - 必要な道具 (テンプレートエンジン、O/Rマッパ) は一緒に提供
- 制約はなるべく設けないようにする
  - URLと処理クラス・呼び出しメソッドの対応を柔軟に指定できるように
  - テンプレートエンジン、O/Rマッパ等との自由な組み合わせを可能に



- CMS「Kvasir/Sora」の動的プラグイン(ログイン機能、日記機能)を実現するためのフレームワークを作る必要性
  - 一応あったけれどかなりてきとーに作ったので作り直したい
- WebUDA「Tuigwaa」のチームでプラグインAPIの検討が進められていた
  - どうせならどちらでも使えるものとして作るといいのでは
    - 基本構想をTuigwaaチームと一緒に考えた
    - その後はほぼ独自に開発を進めてしまいました^^;

こうしてできたのが  
Ymir(ゆみる)です



- フィルタ指向のシンプルなWebアプリケーションフレームワーク
- 読み方は「ゆみる」
  - 北欧神話に出てくる巨人。血肉が世界の元となった。
- 公式サイト: <http://ymir.sandbox.seasar.org/>
  - このサイトはKvasir/Soraで作られています



- J2SE5以降 + ServletAPI2.4で動作
- Seasar2.4ベース
  - HOT deploy対応
- MVCのCの部分を担当
  - お勧め構成はYmir+FreyjaのZPT実装+S2Dao
    - Freyja...XMLテンプレートエンジン構築F/W。ZPT実装を持っている。
      - <http://www.skirnir.net/product/freyja/>



- 開発モードとリリースモード
  - アプリケーションの動作モード
  - 開発中は「開発モード」
    - HOT deploy有効、自動生成有効
  - 運用中は「リリースモード」
    - HOT deploy無効、自動生成無効



- Zope Page Template
- Python (Zope) の世界で生まれたテンプレート言語
- 独自属性 (TAL、METAL) でHTMLの書き換え指示を記述





- **tal:attributes...**属性を置き換え
  - `<img tal:attributes="alt string:aaa" />`  
→ `<img alt="aaa" />`
- **tal:content...**タグの中身を置き換え
  - `<span tal:content="string:aaa">題名</span>`  
→ `<span>aaa</span>`
- **tal:replace...**タグ全体を置き換え
  - `<span tal:replace="string:aaa">題名</span>`  
→ `aaa`
- **tal:omit-tag...**値が真ならタグだけ除去
  - `<strong tal:omit-tag="not:isStrong">内容</strong>`  
→ `内容`



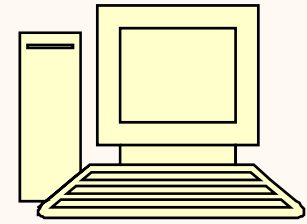
- **tal:repeat...** 繰り返し
  - `<p tal:repeat="name names" tal:content="name">中身</p>`  
→ `<p>名前1</p><p>名前2</p><p>名前3</p>`
- **tal:condition...** 値が真なら描画
  - `<p tal:condition="isError" tal:content="errorMessage">エラーメッセージ</p>`
- **tal:define...** 変数を定義
  - `<div tal:define="c a/b/c"><span tal:content="c"></span></div>`



```
<html>
<body>
  <h1 tal:content="self/title">題名</h1>
  <ul tal:condition="self/messages">
    <li tal:repeat="message self/messages"
        tal:content="message">繰り返しメッセージ</li>
  </ul>
</body>
</html>
```



- サーブレットフィルタとして動作
- URIとPageコンポーネント名のマッピングを持つ
- リクエストURIにマッチするマッピングがあれば、対応する名前のPageコンポーネント(POJO)をS2から取り出し、リクエストパラメータをsetしてアクションメソッドを実行
- アクションメソッドの実行後は基本的にスルー



Webブラウザ

/list.html

### サーブレットコンテナ

#### マッピング

①URIがマッチするかチェック

```
URI: ^/(.*)¥.html$  
コンポーネント名: ${1}Page  
メソッド名: _${method}
```

②マッチした！

コンポーネント名: listPage  
メソッド名: \_get()

#### Seasar2

③listPage  
コンポーネント  
を取り出し、

```
listPage →  
com.example.ListPage
```

④ListPage#\_get()  
を呼び出す

レスポンス

ビューテンプレート  
エンジン  
(ex. Freyja)

⑤サーブレット(ビュー  
テンプレートエンジン)  
がレスポンスを返す



これだけ



- URIのマッピングルールを自由に指定可能
  - Pageクラス名、アクションメソッド名
  - マッピングルールをF/W側から押しつけられずにすむ
- アクションメソッドをHTTP methodによって切り替え可能
  - RESTと親和性が高い
- URIとテンプレートパスが一致
  - 「直接アクセスされたくないから/WEB-INF/の下にJSP置いて」...とかしなくていい
  - テンプレートの位置が分かりやすい
  - テンプレート中で相対パス指定だってできちゃう
  - RESTっぽい
    - 「URIがリソース(テンプレート)に結びついている」感ありあり



- 例外ハンドリング
- ファイルアップロード
- バリデータ
  - form項目単位、submit処理単位で指定可能
- アクセス制御
  - ユーザ毎に操作の許可・不許可を制御可能(実装含まず)
- オブジェクトスコープ
  - sessionスコープ、conversationスコープ、redirectionスコープ...
- リクエスト駆動のJavaクラス自動生成機能(ymir-extension)





- リクエスト駆動
  - アプリケーションを実際に動作させる中でJavaコードを生成
  - 生成に関してWebブラウザ上で指示
- 自動生成できない部分はEclipse上で開発
  - HOT deploy機能によって自動生成フェーズと手動開発フェーズをシームレスに結合
- Beantable
  - DaoのEntityからテーブルを自動生成
  - Entity定義が変更になった場合テーブル定義も自動更新



- ymir-skeletonを利用
- Maven2
- Eclipse3.2
  - Maven Additional Pluginがあると便利
    - <http://www.skirnir.net/eclipse/updates/3.2/>
- Tomcat5.5



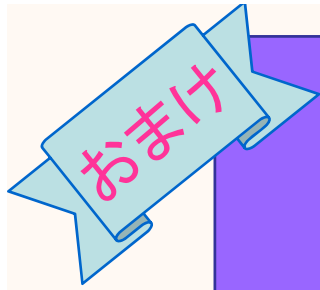
1. ymir-skeletonを展開
2. Eclipseにプロジェクトとしてインポート
3. Tomcatにデプロイ
4. Tomcatを起動
- 5.あとはインタラクティブに開発！



- 簡易掲示板



- 「とにかく簡単にWebアプリを作れるようにする」を目指したWebフレームワークです
- 「机上のフレームワーク」ではありません
  - 多数のアプリプラグイン(Kvasir/Sora)開発事例あり
  - 普通のWebアプリフレームワークとしても実案件に適用中
    - B to C系予約システム(50画面程度×4アプリ、国際化対応)
- まずは触ってみてください



# J2EEシステム開発における標準化 のアプローチ 事例紹介セミナー

フレームワーク導入なども含めた標準化のアプローチを通じ、品質/生産性/保守性向上、スキル育成、レガシースキル技術転用など、開発プロジェクトを成功に導く秘訣を紹介！

## 【事例発表1】日立建機ロジテック様

J2EEオブジェクト指向開発による  
物流基幹系システム開発推進の勘どころ

## 【事例発表2】ソニー・ミュージックアクシス様

開発環境やフレームワーク活用による  
標準化の効果とめざすべき姿

- 11/28(水)15:00～18:00
- JJK会館(東京都中央区)にて
- 参加費無料

- 主催 株式会社アークシステム
- <http://www.arksystems.co.jp/>