

Seasar Conference 2009 Spring

2009年06月



Seasar

Seasar カンファレンス 2009 春

SAStrutsおさらい
と

SAStrutsで**Struts**設定ソースを
動かしてみる



◇ 自己紹介

◇ SAStrutsのおさらい

◇ なぜSAStruts上でStruts設定？

◇ Demo

◇ Seasarプロダクト群の紹介

◇ まとめ



- 名前: 新田 智啓(しんでん ともひろ)
- ブログ: <http://d.haneta.ne.jp/newta/>
- 所属: 自由と束縛の狭間 → 無職!
(株式会社フルネス入社予定)
- 活動
 - SAStrutsコミッタ(一応)
 - S2Csvコミッタ
 - JavaSTHフレームワーク

ちなみにこれ私が描いた落書き。
名前はシーサーキングくん





SAStrutsの

おさらい



と、その前に



Strutsの おさらい



• Strutsとは

- JavaでデファクトスタンダードなWebフレームワーク
- 現場開発者よりもマネージャ的に使いやすい
- シンプルでフレームワークとして出過ぎない適度なつくり



- マネージャ的に使いやすい
 - ✓ Strutsを使ったことがある人が多い
 - **人を集めやすい**
 - ✓ デファクトである
 - **無難、実績がある**
 - ✓ 情報が多い
 - **日本語情報、書籍**



「現場開発者 よりも
マネージャ的に使いやすい」



現場開発者 **よりも**



「よりも」

→ **現場開発者の不満**



→現場開発者の不満

- 設定ファイル地獄
(struts-config.xml、validation.xml)
- 設定ファイルとクラスの関連の再定義
- ソースコード修正時の再起動
- テストがしにくい
... など



そこで！

SAStruts登場



現場開発者の不満の**解決**！

- 設定ファイル地獄
 - 設定ファイル**未編集**
- 設定ファイルとクラスの関連の再定義
 - クラスファイルからの**自動組立**
- ソースコード修正時の再起動
 - **HotDeploy**による再起動なしの修正の反映
- テストがしにくい
 - **POJO**なActionクラス
 - ... など



ではマネージャ的に使いやすい。

- ✓人を集めやすい
- ✓無難、実績がある
- ✓日本語情報、書籍

と言われていた

Strutsのメリットは

SAStrutsでどうなったか？



✓人を集めやすい

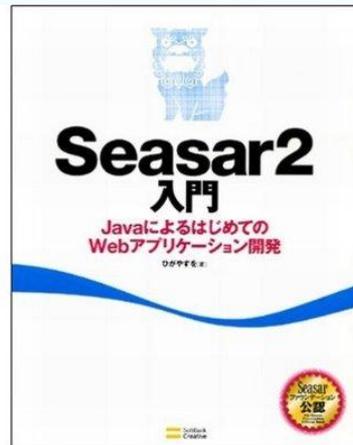
→ Strutsの考え方が同じ(必要なスキルが同じ)
(ActionやFormの考え方やロジックの流れ)

✓無難、実績がある

→ 別セッション

✓日本語情報、書籍

→ Seasarプロダクトは日本発
SAStruts関連の書籍も
ファウンデーション公認の本が発売済み





SAStrutsは

Strutsの**メリット**はそのままに
デメリットを消し去っている
プロダクト



SAStrutsの使い方 ちよつとだけおさらい



SAStrutsにおける基本的機能

- URLとActionクラスの関連付け
- 簡単なValidationの場合
- 色々なチェックを行うValidationの場合



• URLとActionクラスの関連付け

http://localhost:8080/sa-struts-tutorial/helloWorld/hello

アプリケーション名

```
package tutorial.action;
```

```
public class HelloWorldAction {
```

```
@ActionForm
```

```
public HelloWorldForm helloWorldForm;
```

```
@Execute (validator = false)
```

```
public String hello() {
```

```
helloWorldForm.message = "Hello World!";
```

```
return "hello.jsp";
```

```
}
```

```
}
```

アクションクラス名

メソッド名

遷移先パス



- **Validationの指定1 (簡単なもの)**
 - アノテーションで指定

```
package tutorial.form;  
  
public class HelloWorldForm {  
    @Required  
    public String arg1;  
    @IntegerType  
    public String arg2;  
    public String message;  
}
```



Validationの指定2(いろいろするとき)

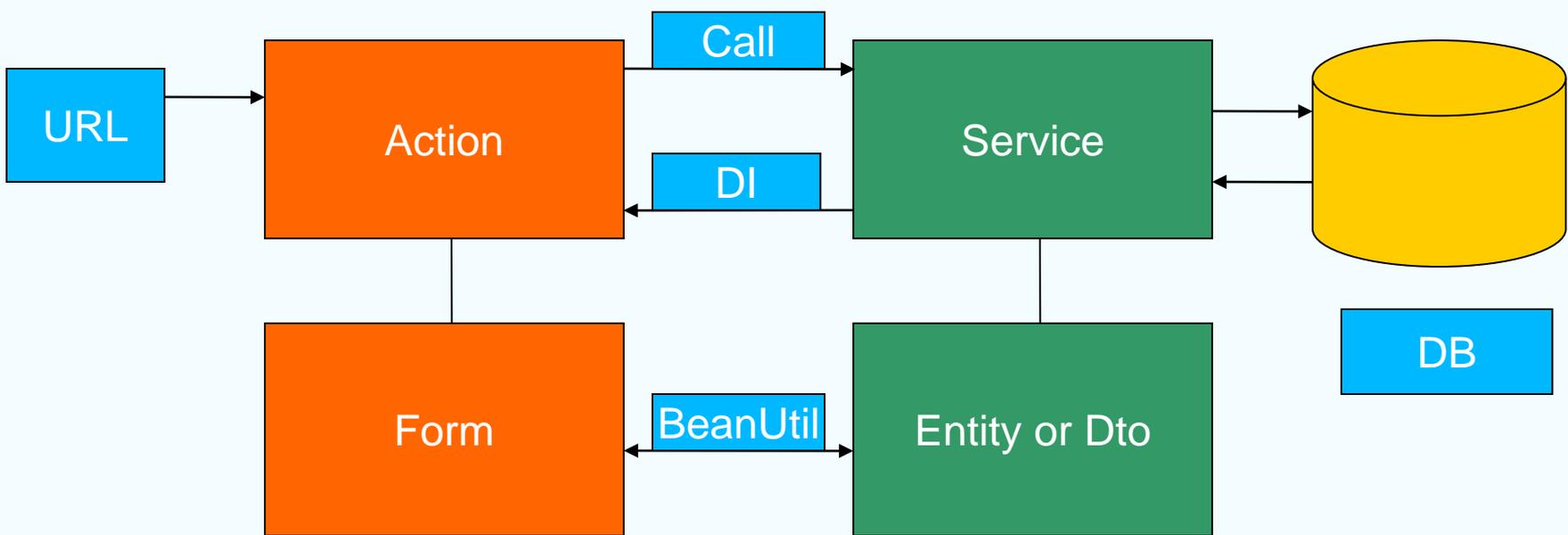
- メソッドでチェック

```
package tutorial.action;
```

```
public class HelloWorldAction {  
    @Execute(validate = "checkMethod")  
    public String submit() {  
        return "checkOK.jsp";  
    }  
    public ActionMessages checkMethod() {  
        ActionMessages errors = new ActionMessages();  
        ...  
        return errors ;  
    }  
}
```



基本図





- FormクラスはServiceクラスに渡さない
 - 画面データ情報である
 - Formクラスのメンバは全てStringにしておく
 - (バリデーションされていないデータも保存)
 - ActionでDtoに詰め替える
 - Form、Dtoの詰め替えはBeanUtilを利用



- 入力系、出力系のメソッドは分ける
 - 入力系はRedirectすると良い
 - (ブラウザのF5[更新]対策)
 - 出力系はDBからデータを取ってきて表示のみ
 - 出力系はバリデーションしない
 - メソッド名で分かるようにしておくとう便利
(入力系はdoXxxxとするなど。)



やっときた、
SAStrutsで

Struts設定を動かす



まず、SAStruts上でStrutsの設定で
動かすためのAddonを作りました
これを追加することでStrutsの設定ファイル
を読み込ませて
動作させることができます



とりあえず
Strutsの設定で
何とかかんとかと言う説明がめんどいので
このStruts設定で動くAddonに
名前を付けました



SAStrutsAOS

Additional Original Style

と名づけてみた



• どんな動き？

- **SA**Strutsでは処理しないようになっている
Strutsの設定ファイルの**struts-config.xml**に
ある**action-mapping**や**form-bean**の定義も読
み込みます
- **struts-config.xml**で読み込んだ設定は
SAStrutsの処理ではなく、
Strutsの処理で動作します

これがポイント！



まとめると

`struts-config.xml`で読み込まれた設定は
Strutsで動作し、
SAStrutsの配置ルールで設定された
Actionクラスは**SAStruts**で動く



- うれしいこと

- Strutsの資産はそのまま動かして新しく作る部分は**SAStruts**で作れる
- Strutsがそのまま動くため新規開発でなくても改修時にSAStrutsでの開発を**提案**できる
- SAStrutsを利用して作る部分は各種Seasarプロダクトと簡単に**連携**できる
- SessionやRequestなどが**共有**できる
- Struts技術スキルだけでなくStrutsのソースそのものを**再利用**できる(かも)



SAStrutsAOSの利点

- SAStrutsで作れる → 開発者にやさしい
- 提案できる → コストと現場理想の間の現実路線
- 連携できる → 便利なS2プロダクト付
- 共有できる → 小細工なしで共有可能
- 再利用できる → 今ある資産を無駄にしない



- 使用上の**注意**

- AddonのJarを追加することにより動きます

- SAStrutsの**正式のものではない**

- (要望、フィードバックによっては正式にSAStrutsのリポジトリに入れることも考えるかも)

- SAStrutsとStruts部分が入るためプログラムとして**きれいな形ではない**

- **(コストと理想の現実路線を拾ったプロダクト)**

- (Struts動作の互換を優先しているためSeasarコンテナにコンポーネント管理していない、HotDeployも出来ない)

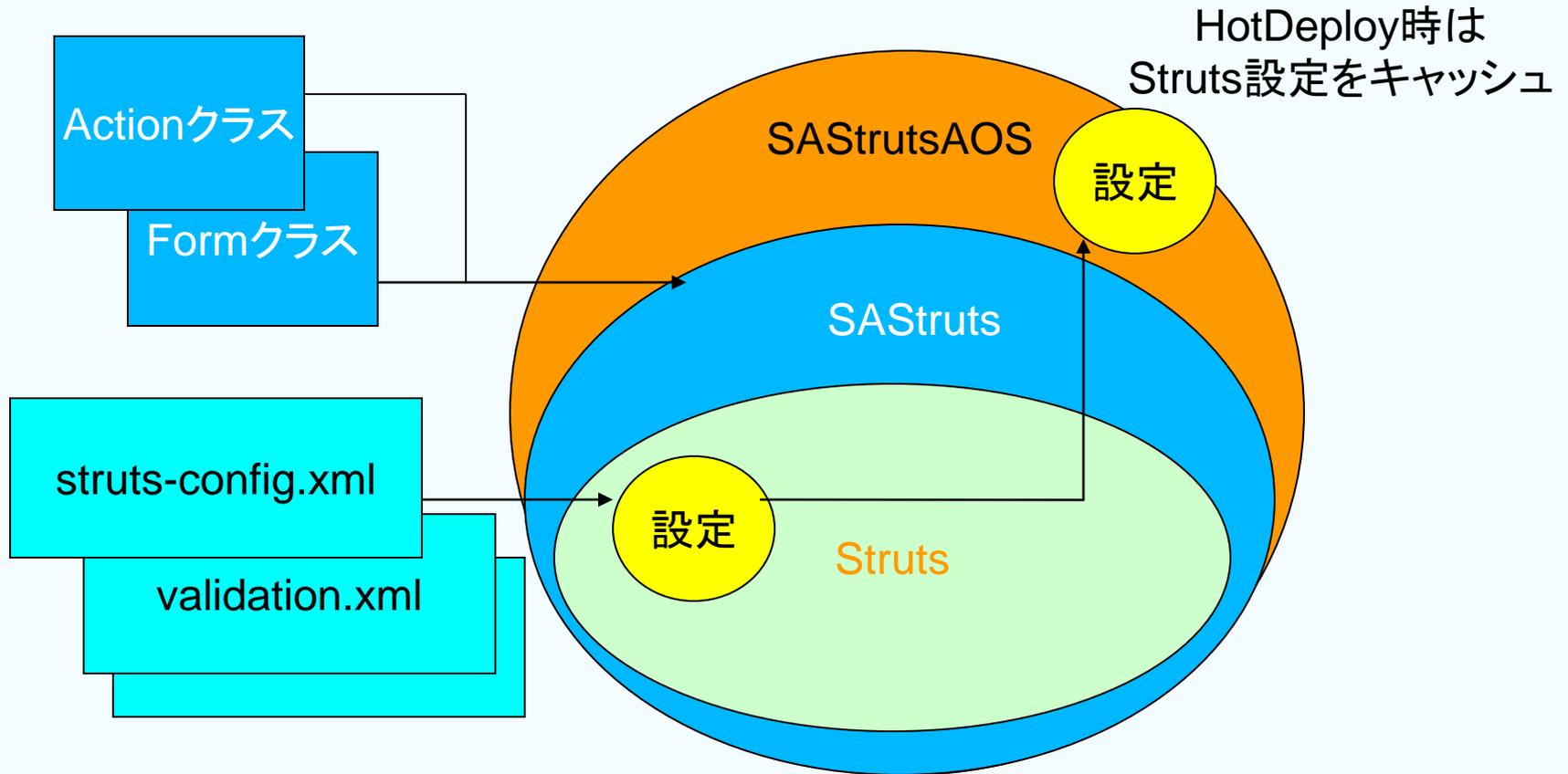
- 動作StrutsのバージョンはSAStrutsの利用している**Struts1.2.9**



SAStrutsAOSの 動作イメージ図



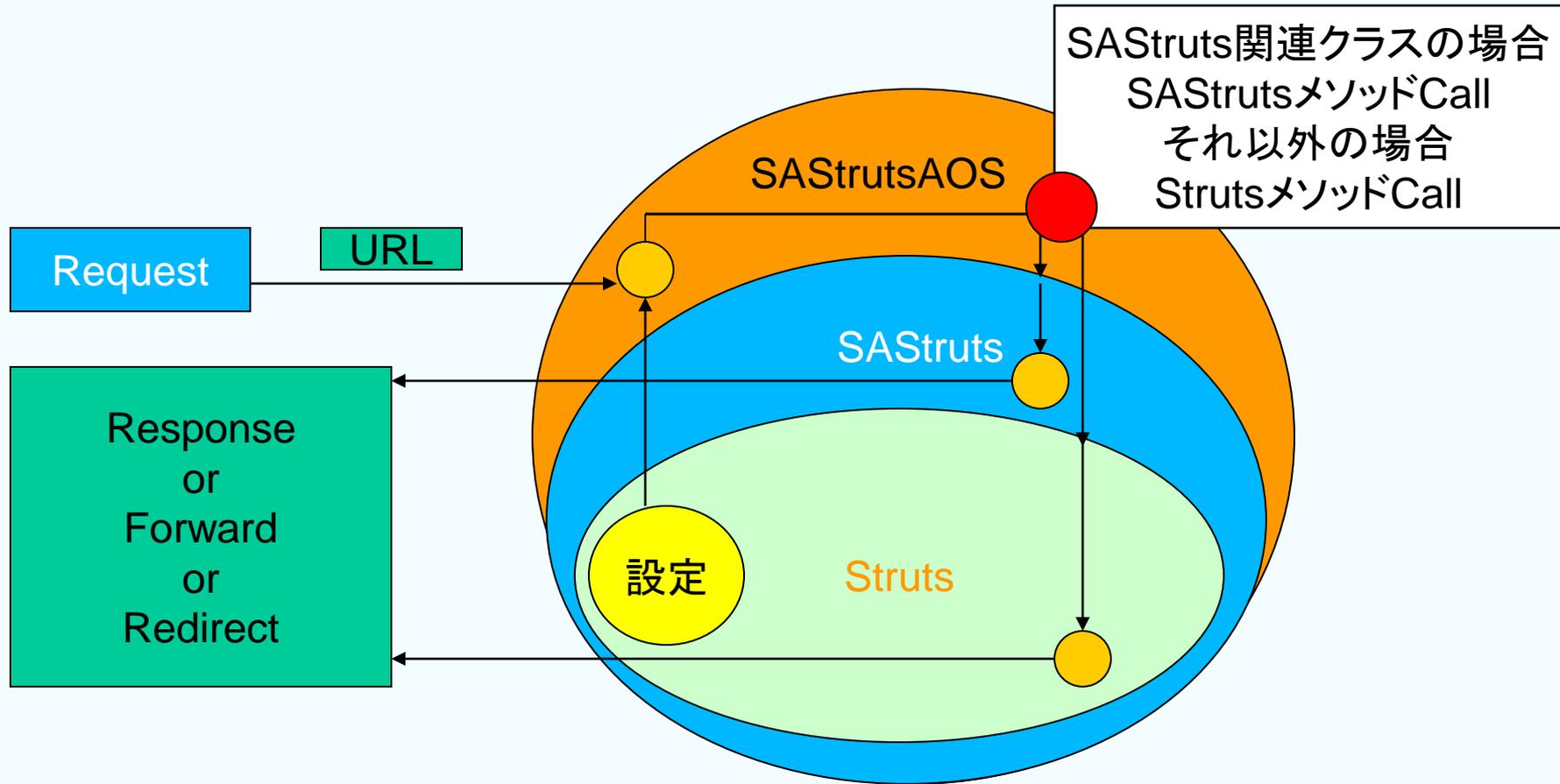
SAStrutsAOSの動き設定ファイル読込時



※注 これは簡略化したイメージ図です
実際にはSAStrutsとStrutsはもっと相互に連携して実行されます



SAStrutsAOSのAction実行時



※注 これは簡略化したイメージ図です
実際にはSAStrutsとStrutsはもっと相互に連携して実行されます



使い方



- SAStrutsのプロジェクトをベースにします
 - (Doltengでプロジェクト作成をしてください)
- StrutsAOSのJarをダウンロード
 - <http://sourceforge.jp/projects/javasth/docs/sa-struts-aos.jar/ja/3/sa-struts-aos.jar>
- クラスパスにjarを追加
- **web.xml**と**struts-config.xml**に設定してある
 - S2ModuleConfigFactory
 - S2RequestProcessor
 - S2ValidatorPlugIn
 - ↓
 - S2AOSModuleConfigFactory
 - S2AOSRequestProcessor
 - S2AOSValidatorPlugIn

Packageも同じ位置にしたので
修正簡単

に変わる。



- 旧Struts資産と新規に追加するSAStruts資産の**パッケージ**は分けておく
- 旧Struts資産と新規**SAStruts**のServiceクラス利用時などにトランザクション管理やコンポーネント管理に注意する。
 - (あまり**混ぜて使わない**)
- 共有するものは**Session**や**Request**、**DB**。
 - (ほかクラス群についてはUtil系クラス、計算のみのロジック系クラスに抑えたほうがはまらないかも)



■ Demo

Struts



StrutsAOS

(Struts & SAStruts)



Seasar(**SA**Struts)
を導入することで
連携できる
Seasarプロダクト



- O/Rマッピング

- S2Dao
- S2JDBC
- ツール(pluginなど)
 - Dolteng
 - DBFlute
 - S2JDBC-Gen

- Viewエンジン

- Mayaa

- Sandbox

- バッチフレームワーク
 - S2Chronos
- Mail送信
 - S2Mai
- Excel出力
 - S2Fisshplate



まとめ



SAStruts**AOS**で手軽に
既存**Struts**システムに
SAStrutsを追加してみよう



Struts から SAStruts で
Seasarプロダクトの第一歩を
!



ご清聴ありがとうございました