



Seasarとともに歩んできたプロダクト「Backlog」の開発事例 ～どっこい生きてるSeasar2.2～

Seasar Conference 2007 Spring
2007年5月27日
株式会社ヌーラボ 縣 俊貴

自己紹介

- 縣 俊貴(あがた としたか)
- 株式会社ヌーラボ
 - Backlog開発者
- コミッタ
 - S2Dao - S2Pager
 - ぶり(escaflow) - Webフローエディタ
 - Cubby – 100% Pure JSP Web Application Framework
- 執筆
 - WEB+DB PRESS,Javaセンスアッププログラミング,etc

私のSeasar2との関わり

- 歴史
 - 2003年 DIコンテナ記事の執筆@WEB+DB PRESS
 - WebWork2
 - PicoContainer
 - 2004年7月 Seasarのからさわぎ@福岡
 - 弊社(ヌーラボ)協賛
 - ここで知り合った方が入社し、Backlogのメイン開発者になったり
 - 2004年11月 S2Pager開発
- 社内のJavaプロジェクトはほとんどSeasar2ベース
 - Seasarに恩返しできれば
 - 少しでもノウハウの共有になれば
 - 開発に対しての安心感とリスクのポイント
- OSSと企業の関係に関心があります

アジェンダ

- Seasar2とBacklogの歴史
- Backlog開発のポイント
- Backlogの原材料(ライブラリ)
- 企業とOSSとの付き合い方
- Q&A

Seasar2とBacklogの歴史

Seasar2とBacklogの歴史

Backlog開発のポイント

Backlogの原材料(ライブラリ)

企業とOSSとの付き合い方

Q&A

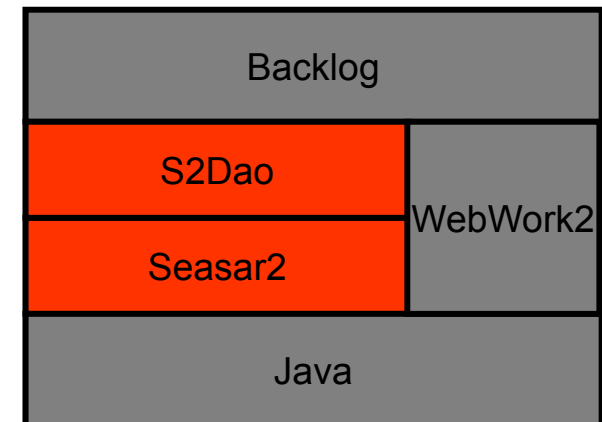
Backlogとは？

Web上で簡単に使えることを目指した、ASP型のプロジェクト管理ツールです。



- 複数プロジェクトの管理
- タスク管理
- Wiki
- 今後の予定
SVN/ML/ファイル管理
- フリー版、商用版、パッケージ版

Seasar2を基盤として採用！



Backlog開発の理由

- ・「易しさと楽しさ」があるプロジェクト管理ツールがなかったの
で・・・作っちゃいました。
- ・自分が困っていることはみんなも困っているはずだ、というこ
とで、プロダクトにしました。

①どこでも、いますぐ

東京・福岡での開発、デザイナーさんとの協業が多い。さくっと、プロジェクトごとに、タスク管理ソフトをセットアップして、セキュリティに気をつけて・・・めんどくさい。。



②簡単・シンプル！

簡単に使えるものがない。やりたいことは、タスクの「未着手」「着手中」「処理済み」「完了」で十分。機能を削ることに集中しました。

状態：未対応

状態を「処理中」に変更

状態を「処理済み」に変更

状態を「完了」に変更

完了理由

担当者

③楽しく使える

楽しいほうが持続するし、敷居も下がる。ゲーム感覚でタスクをクリアしていく。

4月26日(木)			
BLG-165		山本@ヌーラボ (17:53)	>担当者を...
BLG-232		山本@ヌーラボ (17:50)	おお！ありが
BLG-232		ゲスト (17:49)	直りました
BLG-232		ゲスト (17:46)	おそらく入

Backlog ASPサービスの規模

2005年6月オープン～

2007年5月現在(フリー版、商用版の合計)

- スペース数:1400
- プロジェクト数:1800
- ユーザ数:4400

Seasar2とは？

オープンソースのDIxAOPコンテナ。S2Daoをはじめ、Seasar2をベースにした周辺ライブラリ・フレームワークが充実している。



Seasar2はフレームワークのためのフレームワーク！

なぜSeasar2を採用したの？

- 誕生のタイミングがぴったりだった。
- ORM MapperとしてS2Daoが素晴らしかった。



2003年	<ul style="list-style-type: none">• ロッド・ジョンソンJ2EE実践• Spring Framework、PicoContainer• WebWork2	<ul style="list-style-type: none">• プロトタイプをWebWork2、Hibernate2で開発・・・Hibernate2の敷居の高さに挫折・・・
2004年	<ul style="list-style-type: none">• 1月 Seasar2.0の開発開始• 3月 Seasar2.0リリース• 7月 S2Dao1.0リリース• 11月 Seasar2.1リリース	<ul style="list-style-type: none">• 5月 Backlog開発開始• 10月 S2Pagerの実装・Blogでのリリース• 11月 社内プロジェクトで使用開始
2005年	<ul style="list-style-type: none">• 2月 Seasar2.2リリース• 11月 Seasar2.3リリース	<ul style="list-style-type: none">• 6月 ベータ版リリース
2006年	<ul style="list-style-type: none">• 2月 S2PagerのS2Daoへの組み込み• 11月 Seasar2.4リリース	<ul style="list-style-type: none">• 7月 商用版リリース
2007年		<ul style="list-style-type: none">• 6月 パッケージ版提供開始

Backlog開発のポイント

Seasar2とBacklogの歴史

Backlog開発のポイント

Backlogの原材料(ライブラリ)

企業とOSSとの付き合い方

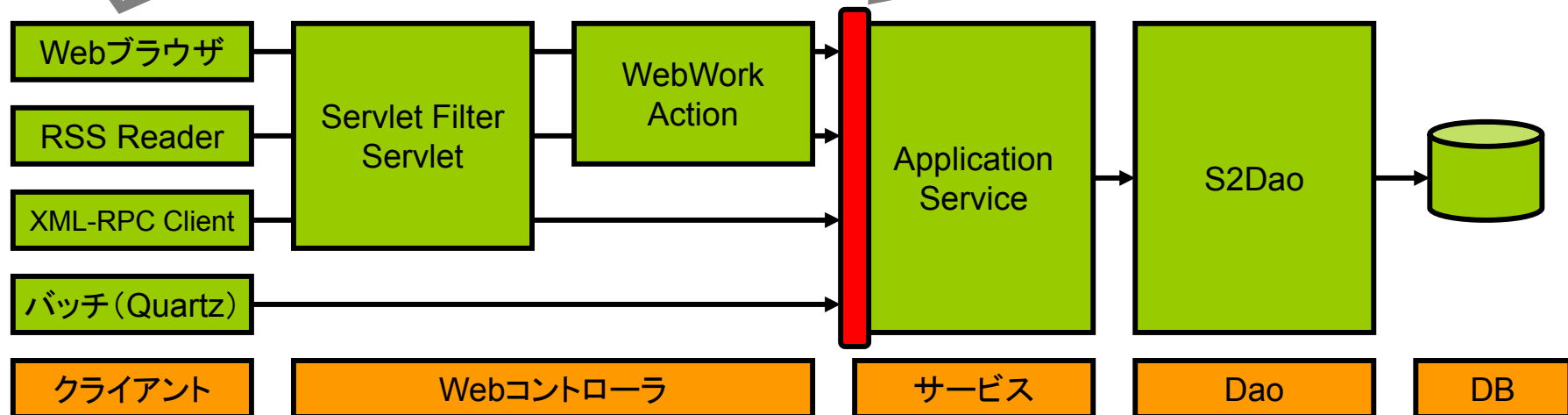
Q&A

アーキテクチャ: 全体のデザイン

レイヤーアーキテクチャです。クライアントに依存しないアプリケーション機能をサービスレイヤーとして用意して使用しています。

開発当初はWebブラウザ以外のクライアントは想定していませんでした。

クライアントに依存しないサービスレイヤーのおかげで、機能の重複が減りました。サービスレイヤーはPoEAAやGoya、Spring Frameworkのサンプルアプリケーションを参考にしました。



アーキテクチャ:DI

Seasarでライブラリやサービス同士ののりづけと設定をDIで行っています。

設定ファイル

```
<!-- メール送信クラス -->
<component class="com.ozacc.mail.mock.MockSendMail" instance="prototype">
  <property name="host">backlog.smtpHost</property>
  <property name="debug">true</property>
  <aspect pointcut="send">async</aspect>
</component>
...
<component class="...SummaryMailServiceImpl" />
```

ソースコード

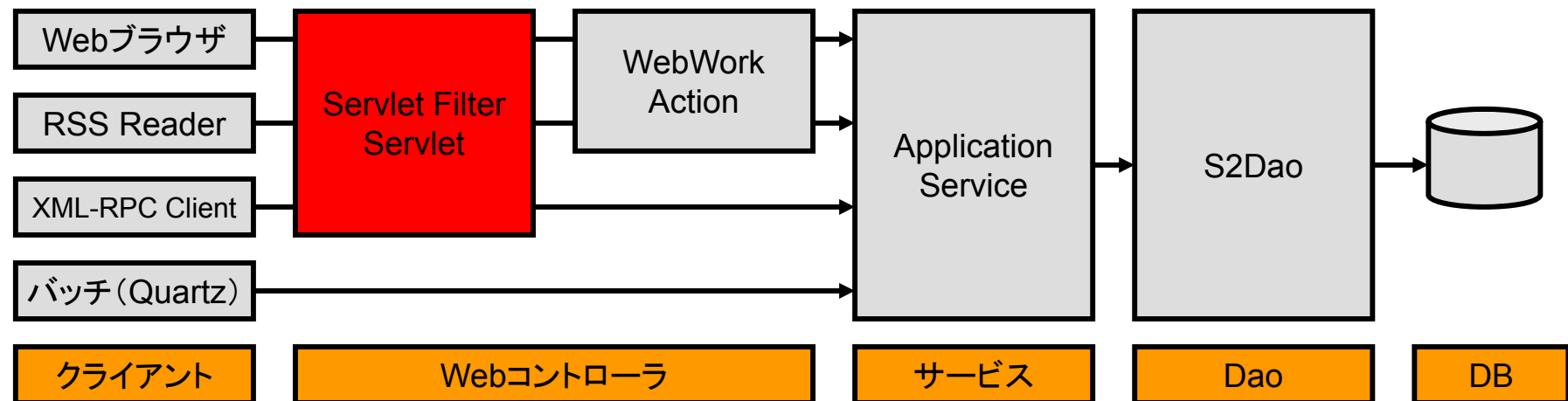
```
public class SummaryMailServiceImpl implements SummaryMailService {
  /** メール送信クラス */
  private SendMail sendMail;
  public void setSendMail(SendMail sendMail) {
    this.sendMail = sendMail;
  }
}
```

アーキテクチャ:AOP

- 3つのAOP機能
 - ServletFilter
 - Seasar2
 - WebWork2

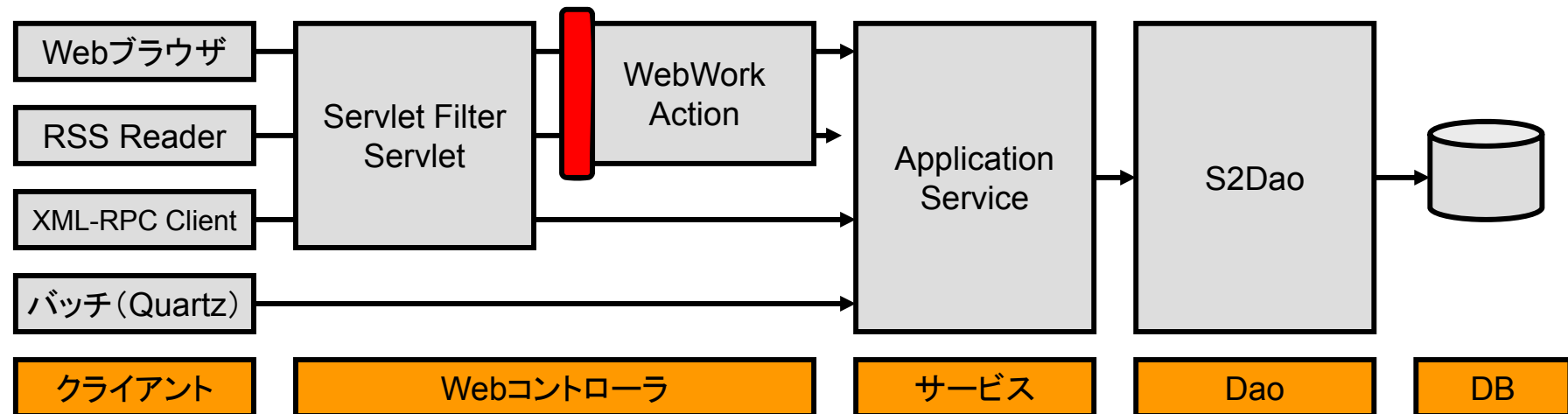
AOP:Servlet Filter

- 全てのリクエストに適用できます。
 - EncodeFilter
 - URL解析・不正アクセスチェック
 - リクエストスコープに対してのAOP (DevFilter, S2.4のHotDeployFilter)
- WebWork2アクション決定・実行前の処理



AOP:WebWork (Xork) Action Interceptor

- 個々の画面処理(Action)単位で適用
 - セキュリティ、アクションごとに不正アクセスチェック
 - デモユーザの制限



AOP:WebWork (Xork) Action Interceptor

- WebWork(XWork)のAction専用なので、引数や戻り値が型に安全でキャストの必要がない

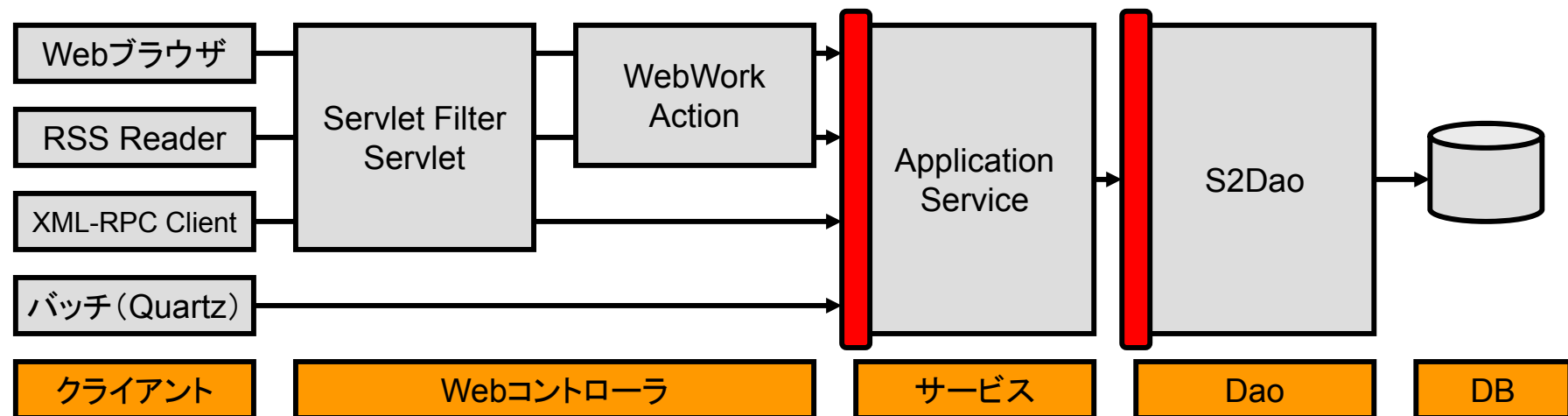
ソースコード

```
import com.opensymphony.xwork.interceptor.Interceptor;

public class HogeInterceptor implements Interceptor {
    public String intercept(ActionInvocation invocation)
        throws Exception {
        Action action = invocation.getAction();
        if (チェック) {
            return "error.notfound";
        }
        return invocation.invoke();
    }
}
```

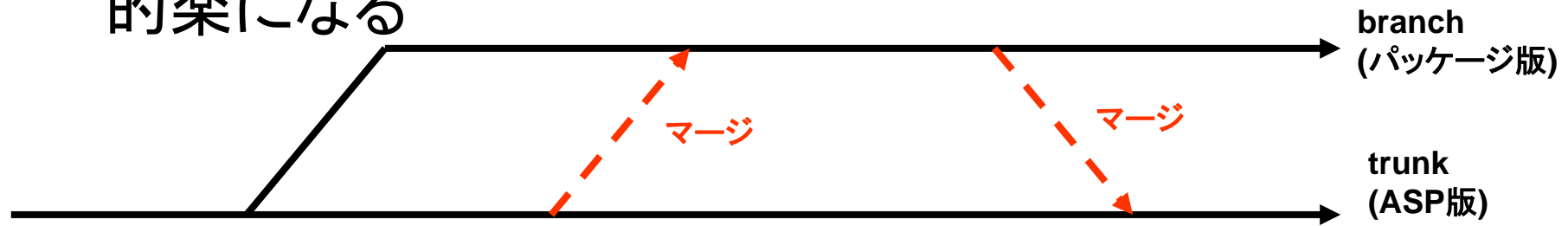
AOP: Seasar2

- 主にサービスレイヤーに対して、共通的な処理を編み込み
 - キャッシュ、ロギング、トランザクション、S2Pager
 - 横断しない関心ごと(後述)



AOP:Backlogでの使いどころ

- 一般的には「横断する関心ごと＝共通なコード」に適用される
- Backlogでは「横断しない関心ごと＝特殊な状況」に適用することが多い。(パッチ的な適用)
 - 複数のプロダクトラインの違い
 - 期間限定機能
 - 特定ユーザ向け
- ブランチ後のマージがファイル単位になるので比較的楽になる



AOP: 横断しない関心ごとの例

- CampaignInterceptor
 - フリー版からの期間限定乗り換えキャンペーン
 - 初期費用を無料に
- UnlimitInterceptor
 - ヌーラボスペース用
 - ファイルサイズなどの制限なし
- DemoAuthInterceptor
 - デモサイトの制限
 - プロフィール変更ページなどを使用不可に

AOP:コード例(AOPを使用しない場合)

特殊な状況、一時的なコードで本来のコードが汚染されてしまう。

```
public class LisenceServiceImpl implements LisenceService {  
    /** 初期費用計算 */  
    public int getInitialCost() {  
        if(キャンペーン期間だったら) {  
            return 0;  
        }  
        ... 本来の初期費用計算処理  
        ....  
    }  
}
```

AOP:コード例(AOPを使用)

特殊な状況、一時的なコードをAOPで編み込み。本来のコードを汚染しない。 →マージが楽になる

```
public class CampaignInterceptor extends BaseInterceptor {
    public Object invoke(MethodInvocation invocation) throws Throwable {
        if(キャンペーン期間だったら) {
            return 0;
        } else {
            return invocation.proceed();
        }
    }
}
```

```
public class LisenceServiceImpl implements LisenceService {
    /** 初期費用計算 */
    public int getInitialCost() {
        ... 本来の初期費用計算処理
        ...
    }
}
```

S2採用のメリット:S2Dao最高

- キラーフレームワーク
- SQLファイル最高
 - SQLを書かなくていいけど書ける
 - 別ファイルに書ける
 - SQLの保守性向上
 - 普通のSQLで書けた

S2採用のメリット:社内で標準化

- Backlog集めたノウハウをためて、他のプロジェクトに適用できた。標準化。
 - 社内のJavaプロジェクトの9割がSeasar2
 - 全てのメンバーが、レイヤーアーキテクチャ、S2Dao、S2TestCaseを使用できる。
- 社内プロジェクトでのSeasarとの組み合わせ例
 - S2Struts + Mayaa
 - Java Web Start + SWT
 - Cubby
 - DB移行変換バッチもSeasarベースで

S2採用のメリット:オープンソース

- トラブル時にコードを見たりパッチをあてたりできる
- 古いバージョンでも自分で修正ができる。
- コミュニティが活発、イベントドリブン、意見を提案しやすい
- コミッターに直接コミュニケーションできる
- ドキュメントが日本語

Seasarで困ったこと

- Seasar本体、S2Daoなどfinalが多くて、拡張が難しいところがあった
 - 必要であれば、徐々に開いていくポリシー
 - S2Pagerのように透過的な拡張方法もある
- エラーメッセージがわかりにくかった
 - 最近のバージョンではずいぶん改善されてきた
 - MLなどで困っていることを報告してあげることが大事
- ライブラリ関係のバージョンの組み合わせが困った
 - リリースノートにだいたい書いてある
 - 無理してバージョンを上げないのもあり
 - 安定的な組み合わせを知りたい人は多いかも

トラブルの解決方法

- ソースコードを読めること、ソースコードを改変できることが武器。
 - 現実問題として、OSS全般、これらの力がないと難しい部分がある。
 - 想像しているよりもソフトウェアの寿命は長い。
- コミュニティを活用しましょう
 - MLでの質問
 - パッチの提供
 - コミッターになる
 - OSSプロジェクトをはじめる

ライブラリをバージョンアップしたい症候群

- バージョンアップのポリシー
 - S2.2、WW2.1系のまま
 - 必要十分なので、ベータリリースからあまりあげていません。
 - あげて問題ないときは基本的にあげる。うまくあげれない場合があるし、あげなくてもいい場合もある。
 - 必要に応じて、パッチやバックポート
 - WW 2.2→2.1(Fileuploadの機能)
 - S2Dao 1.0.最新版 →1.0.28(IDアノテーションによるシーケンスの切り替え)
- バージョンアップ時の注意
 - チェンジログをきちんと見るべし
 - 検証やテストをながす。

作り直したい症候群

- バックログ自体を作り直したくなる
 - 他の言語で作り直したい
 - ビューのフレームワークを変更したい
- バージョン2の法則
 - だいたいうまくいかない
 - 作り直しても開発者満足で終わる場合がある
 - うまくいっているものを伸ばす

Backlogの原材料(ライブラリなど)

Seasar2とBacklogの歴史

Backlog開発のポイント

Backlogの原材料(ライブラリ)

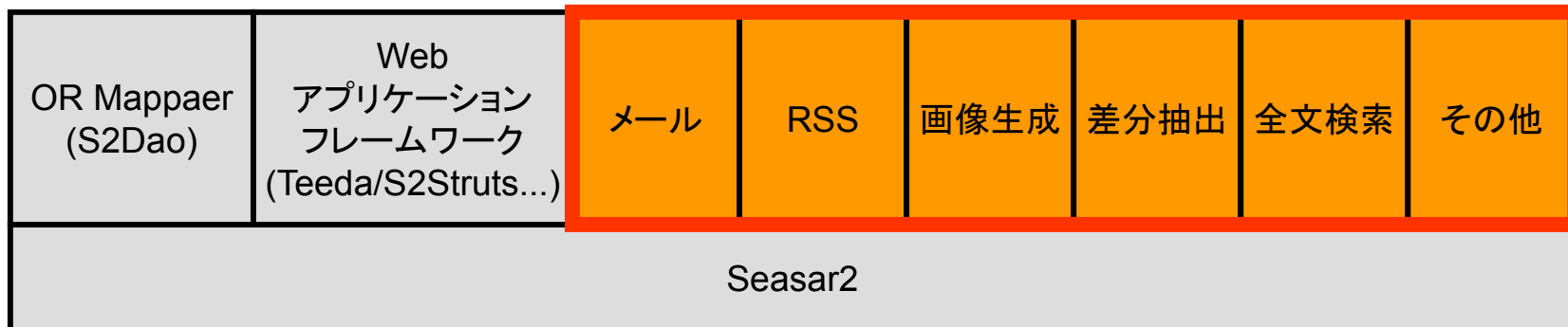
企業とOSSとの付き合い方

Q&A

周辺ライブラリ

Seasarファミリーの守備範囲以外の部分をサポートする、ライブラリセットで、いわゆる「Webアプリケーション」でよく見かける機能のフルスタックが揃う。

Seasarファミリー以外のライブラリセットで振るスタックを補完する。
そのうち、Seasarでサポート(S2xxxのように)されると、安心して使えるなあ。



原材料その① – seasar.org

- S2Container 2.2
- S2Dao & S2Pager
- S2Quartz
- S2WebWork2(ヌーラボオリジナル)
- Trickle – workflow engine(未リリース)
- S2CSV(未リリース)

原材料その② – seasar.org以外

- WebWork 2 – WebApplication Framework
- Jakarta Commons Lang/Collections ...more
- Ozacc Mail Library – Mail送信
- Lucene – 全文検索
- Jakarta Commons JRCS – Diff
- jcaptcha – captcha(ゆらゆら文字)
- jsonrpc - Ajax
- rome - Atom/RSS Java utilities
- FreeHEP – GIF出力
- loom - wiki engine(社内ライブラリ)
- sventon(予定) - SVN

WebWork2



- Strutsの対抗馬的なWebアプリケーションフレームワーク
 - 2003年～あまり流行ませんでした・・・
 - DI(IoC)/AOP/POJO Action/コンポーネント志向
 - 先進的だけど、過渡期っぽい
 - そしてStruts2へ



ozacc-mail library

- Mail送受信を簡単に行えるライブラリ
 - メール送信はほとんどのWebアプリで必要
 - DI時代のライブラリ、組み込まれることを想定しているので、さくっと使えます。
 - Springのサンプルが付属、ほとんどそのままSeasar2に組み込めます。
- ほとんどの場面でOMLで対応可能
 - Velocityによるテンプレート機能
 - HTMLメール/添付ファイル対応
 - モックによるメール送受信のテストができる
 - 一括送信機能(大量メール)
- WEB+DB PRESS Vol.30に記事あり

Jcaptcha



- Captcha
 - Completely Automated Public Test to tell Computers and Humans Apart
 - コンピュータか人間かを判定するためのもの、社内ではゆらゆら文字って呼んでいます。
- Jcaptchaはかんたんにゆらゆら文字を生成するライブラリ
 - 文字の揺らぎなど細かくカスタマイズ可能

画像に表示された確認用ゆらゆら文字を入力してください。

s m i n g

確認用ゆらゆら文字

次へ

Apache Lucene

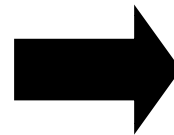
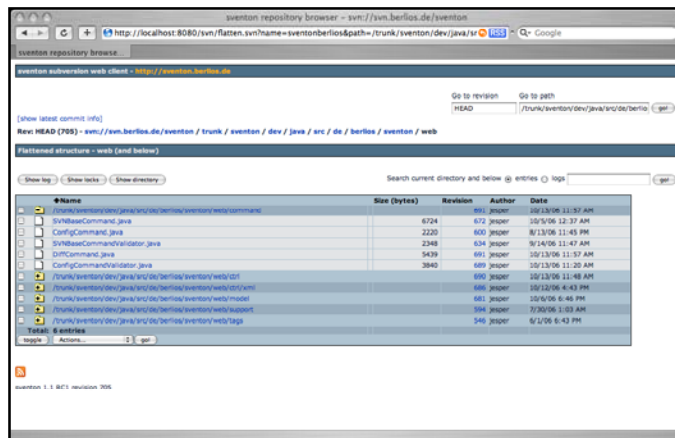


- 課題の全文検索に使用
- 登録されたスペースごとにインデックスを管理
- やりたいこと
 - 通常の検索条件と組み合わせて、キーワード検索したい。
例:「**カテゴリー=開発 AND キーワード=インポート処理**」
- 検索の流れ
 - ①課題の登録・更新時に、Luceneでインデックス作成
 - ②キーワード検索時に、Luceneへ問い合わせ、対象の課題IDリストを取得
 - ③上記課題IDリストをSQLのinで検索条件に加える→プロジェクトで絞り込まれているのでinでも十分早い
- DBの全文検索機能を使用していないので、DBが変更されても対応可能

Sventon

sventon

- WebベースのSVNリポジトリブラウザ
 - Luceneによるコミットログの全文検索
 - ファイルのdiff閲覧
- レイヤーアーキテクチャ
 - Springベース/Spring Web MVC Framework
 - サービス層をSeasarに組み込んで使用可能
 - Webコントローラ/ビューを書けばある程度使えてしまう



Coming Soon.

今後、実装予定です。お楽しみに……

企業とOSSの付き合い方

Seasar2とBacklogの歴史

Backlog開発のポイント

Backlogの原材料(ライブラリ)

企業とOSSとの付き合い方

Q & A

Backlogの開発で学んだこと

- 基本、コミットメントしたほうが双方にとって利益になる
 - MLへの質問・要望・バグ報告
 - パッチの提供
 - プロダクトやライブラリのOSS化
- 反省点
 - いくつか提供していないパッチもあります。
 - 問題点を発見して、社内のみで解決・納得したこともあります。
 - 結果的に、問題点は残留したままなので、そのうち自分たちも困ります。

S2Pagerの例

- S2Daoでページングを実現するライブラリ
 - Backlog + 別プロジェクト用に社内用に作成
 - S2Dao1.0.3xより同梱

全 107 件中 1 件 ~ 20 件を表示			[1]	2	3	4	5	6	次へ>
種別	キー	件名							

- OSS化のメリット
 - コントリビュートしたおかげで、社内ライブラリがデファクトスタンダードに
 - 新しい別の実装を覚える必要がない
 - メンテナンス可能性を手に入れることができる
 - バグフィックスや便利な機能追加もありました。

まとめ

- Seasar2は2.2でほとんど完成していましたが、枯れていて、実績も十分にありません
- 周辺のライブラリセットを確立させると、パターン化できて開発が楽になります
- コミットメントするともっと使いやすいSeasarになります
 - まずはMLから
- Backlogを見かけたら、そこにSeasar2が生きていると思ってください。

ご清聴ありがとうございました。

Q & A。

なにかありましたら。