



Ruby on Rails 評価 抜粋

A. Okubo

はじめに

- 目的: Ruby on Rails 評価報告

- Ruby on Rails とは:

注目のWeb開発フレームワーク(環境)

Web開発者へ大きなインパクト

世界中で大ブレイク 適用例が続出

テーマ

- Ruby on Rails (RoR) の特徴
- Struts (Java) と比べる
- RoRは仕事で使えるか？



Ruby on Rails の 特徴

実例 Railsデモ

- Railsの自動生成機能を使って、
10分で書籍一覧のアプリを作ってみる

Ruby on Railsの特徴 1

■ 徹底した省人力指向

- ・「実用的なデフォルトルール」
決まりきった部分は、割り切ってルール化
- ・ルールに従えば、ほとんど定義不要
(ベストプラクティスの押し付け)
→ 定型作業から脳を開放する

Ruby on Railsの縛り

- テーブルの主キーは“id”固定で連番
外部キーは“参照テーブル名_id”
- テーブル名は複数形、クラス名はその単数形
ex) テーブル:USERS → クラス:User
- カラム名がそのままモデルのプロパティ名に

Ruby on Railsの縛り

- モデル・コントローラ・ビュー・CSS・JavaScript...すべてデフォルトの格納場所が決まっている
- ページ名から、コントローラ・ビューの名前も決まる

ex) `http://localhost:3000/admin` というページなら

コントローラ: `app/controllers/admin_controller.rb`

ビュー: `app/views/admin/アクション名.rhtml`

Ruby on Railsの特徴 2

■ 徹底したアジャイル指向 開発リズム重視

- ・スピード感のある仕様反映（作成・テストの自動化）
効果が**すぐ**見えて、仕様が**すぐ**議論できる

→ 開発者もユーザもリズムに乗る 楽しい 達成感

■ オープンソース

- ・コード自身が、「Rails流」のサンプル
- ・コードのパクリがしやすい → 仕組みが見える
→ 手作り感覚 → うさんくさを覚える人もいる？



Struts (Java) と 比べる

実例 ST管理表

同じ画面を作ってみた

加工費S ST管理表画面

プロダクトコード	<input type="text"/>	ST表修正拠点	APN	
イベント名	TRANS	生産拠点	AOIW <input type="button" value="v"/>	
取込日付	2006/04/17 00:00	BOM情報	M-BOM	

第1版	<input type="button" value="Approval"/>	<input type="button" value="Check"/>	<input type="button" value="Create"/>
2006/04/17 00:00			

	CP	自採	リフロー	マウント	サブ	メイン	PKG	トータル集計
ST(分)	0.00	0.00	2.96	0.00	0.00	0.00	0.07	0.00
加工費 (APN) JPY	0.0000	52.2000	856.5001	514.6000	30.2250	258.8500	237.9250	1950.3001
加工費 (AOIW) JPY	0.0000	52.2000	856.5000	514.6000	30.2250	258.8500	237.9250	1950.3000

	NE	コンテンツコード	コンテンツ名	加工区分	ST(分)	加工費	更新日	ISS	ステータス
			96-02632Z94	E-ASSY,MODULE					対象外
			66-02846Z95	SMT MODULE					対象外
			66-02846Z98	AI(CMC-C-B)MODULE	11042-1 Ref low(2RF)	0.00	0.0000	2006/01/17 00:00	0 M 確定
			66-02846Z99	AI(CMC-B)MODULE	11042-1 Ref low(2RF)	7.31	115.2800	2006/01/17 00:00	0 M 確定
			66-02907Z01	AI(CMC-C-A)MODULE	11042-1 Ref low(2RF)	0.00	0.0000	2006/01/17 00:00	0 M 確定
			66-02907Z02	AI(CMC-A)MODULE	11042-1 Ref low(2RF)	0.07	85.3600	2006/01/17 00:00	0 M 確定
			66-02846Z97	ASSY,MODULE PANEL	11031-1 Mount	2.09	83.3900	2005/06/02 00:00	0 M 確定
			66-05403Z99	A-ASSY,MODULE					対象外
			66-05403Z70	M-ASSY,MODULE					対象外
			96-02632Z95	E-ASSY,MAIN					対象外
			66-02907Z06	SMT MAIN					対象外
			66-02907Z11	AI(CMC-C-A)MAIN	11041-1 Ref low(RF)	0.00	0.0000	2006/01/16 00:00	0 M 確定
			testdgxx	dgxx	11051-1 Radial	0.00	20.8800	2006/10/24 00:00	1 E 確定(変更あり)
			66-02907Z12	AI(CMC-A)MAIN	11041-1 Ref low(RF)	1.00	141.9600	2006/01/17 00:00	0 M 作業中
			53-07407Z92	INL,V0.994_REAL501A					対象外
			66-02907Z08	ASSY,MAIN PANEL	11031-1 Mount	0.05	122.4500	2006/01/17 00:00	0 M 作業中

Struts (Java) と比べる ～トピック

- やる作業は同じ でも記述量が少ない
 - ・型宣言が不要 Ruby
 - ・シンプルなラッピング式のO/Rマッパー ActiveRecord
 - ・ダブった記述は悪・同じ操作の繰り返しは悪
Don't Repeat Yourself
 - ・使いやすいデフォルトルール
Convention over Configuration

Struts (Java) と比べる ～トピック

■ ST管理表のコード量比較

	Struts (Java)	RoR
Model	1500	330
Controller	700	200
View	350	260
定義	140	10

Struts (Java) と比べる ～トピック

■ すぐ試せる

・そのまま動く雛形の生成

モデル・アクション・ビュー…テストケースまでも雛形生成

テスト用Webサーバの組込み

・変更、即ブラウザへ反映

Tomcatより徹底した変更反映

※コードを内部生成、実行時に解釈

Struts (Java) と比べる ～ 良し悪し

■ メリット

- ・インストール簡単で、
（特に初期の）手作業が劇的に少ない
- ・実働する雛形に機能を追加
- ・自動テストを強力にサポート
 フィクスチャ、モックアップの雛形を自動生成
- ・仕様変更の影響範囲が小さめ
- ・簡潔に書ける
 → 効果的で得した感覚

Struts (Java) と比べる ～ 良し悪し

■ デメリット

- ・ 実行速度が遅い
- ・ ライブラリ資産・情報が少ない
- ・ 複合キーが使えない (既存DB適用に難)
- ・ Decimalサポートは2007年1月リリース
- ・ 無償のソースデバッガがまだ無い (もう少し)
- ・ 継承的なクラス設計がフィットしにくい
(Include Moduleによる機能追加を多用)
→ Javaのinterfaceベースのクラス設計に近い
- ・ Web全体の仕組が見えていないと、理解しにくい

Decimal-String変換
を書いて、
モデルに組み込んで
使ってみた。
割と簡単

Struts (Java) と比べる ～結論

- 仕様変更対応が早い
- 大人数開発より、少人数のチーム開発(加工費Sなど)の方が効果大
- ビジネスロジックが面倒 かつ Webパーツの使い回し多いほど効果大
- ビギナー or Web開発を通して経験した方向き
- リプレース時の移植性は一長一短でトントン(多分)。



Ruby on Rails は 仕事で使えるか？

実例 ツール作成の報告

- 加工費S向けサンプルツール
 - ・ログビューア
 - ・DB比較
 - ・レガシーデータ検索

RoRは仕事で使えるか？

- 統合パッケージでインストールしやすい
- 素早く開発・変更に強い
- 実行速度遅い(が、スケールアウトは容易)
- もう少しエンタープライズ向けノウハウの蓄積待ち
- ユーザ多く、5～10年は使えるだろう



今はまだ

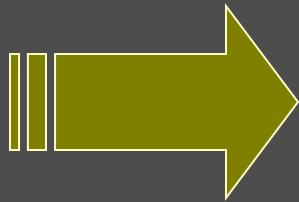
- ファイアウォール内
- 中小チーム開発(加工費クラスならOK)
- 初期プレ・ツール・ユーザフロントエンド向き



最後 に

最後に ～個人的感想

- Web-SEとして 眼から鱗を体験
- Rubyのパワー データ加工・ロジックプロト



- 今まで使っていた環境を
超えた視点
→ 現システムへのF/B
- 自分のブレイクスルー

もっと知りたい人に RailsとRuby

- Ruby on Rails入門—優しいRailsの育て方
～1冊だけ買うなら
- RailsによるアジャイルWebアプリケーション開発
～バイブル(AWDwR)。とっつき悪いが中身凄い
- はじめよう Ruby on Rails
～バイブルのサブ向き。開発実況風
- Rails 本家 <http://www.rubyonrails.org/>
- プログラミングRuby 第2版 言語編・ライブラリ編
- Ruby ドキュメント <http://www.ruby-lang.org/ja/documentation/>

もっと知りたい人に Railsの根っこ

- IBM Java technology 境界を越える
http://www-06.ibm.com/jp/developerworks/java/lib_series.shtml#cb
- Rubyist Magazine 本当は面白い Ruby on Rails
<http://jp.rubyist.net/magazine/?0015-ForeWord>
- Ruby on Rails の作者 DHHへのインタビュー
<http://capsctrl.que.jp/kdmsnr/wiki/transl/?AnInterviewWithDHH>
- でこぼこ翻訳ハジメ Railsへの反論は根拠のないものではないことについて <http://funk.exblog.jp/2335721/>
- Martin Fowler's Bliki エンタープライズRails <http://capsctrl.que.jp/kdmsnr/wiki/bliki/?cmd=view&p=EnterpriseRails>
- Rubyist Magazine 諸々の連載 <http://jp.rubyist.net/magazine/>

Any Question? -> Fin.

ありがとうございました