

Ruby
と
Meeting2000

松尾尚典 @ ForUs

目次

- 1 Meeting2000について
- 2 Rubyを選択した理由
- 3 開発してみても
- 4 今後の予定
 - グループウェアサーバへの道
- 5 まとめ

1 Meeting2000について

- 参加希望者の都合を調整し、参加申込を受け付けるWebアプリケーション (便利!)
 - 幹事さんの苦労を軽減
 - 参加者の手間も削減
- グループウェア
 - 情報の共有、加工と提示
- Rubyの普及にある程度貢献できた感触

Meeting2000について(2)

- 構成
 - 自作HTTPサーバ機能
 - 自作Webアプリフレームワーク
(リクエスト、ページ、セッション)
- 規模
 - 3000行、0.5人月 (1人*0.5月)
 - 日曜プログラム程度、中規模というより小規模
 - Webアプリとしては結構複雑
(商用、社内システムを除けば^^)

2 Rubyを選択した理由

- 私
 - 言語屋さんでも言語マニアでもない
 - コーディングが趣味でもない(趣味はゴルフだ)
 - やりたい事がすばやく出来れば良い
 - 以前は Perl4
- Ruby
 - 書きたい事が書け、書きたくない事は書かなくて済む度合が、自分にとって丁度気持ち良い

Rubyのよいところ 概観

- vs システムプログラミング言語
 - 開発効率がよい(C、C++)
 - 型等による制約が少ない(Java、C、C++)
 - 多少遅いが気にならない
- vs 他のスクリプティング言語
 - OOである。しかも分かりやすいOO
 - 日本製で、日本語のコミュニティが本家

キーとなった機能

- 素直なOO
- gc、thread、socket、正規表現、ハッシュ、例外...
- 変数がオブジェクトのハンドル
- Ruby Home Page を見て(それが本当なら)
良いと確信

選択肢

- スクリプティング言語
 - Perl[45]
 - Tcl
 - 以前5000行ぐらい書いたけど、超苦い記憶だから早く忘れたい、ていうか忘れた^^
- システムプログラム言語
 - Java
 - C++
 - 出来るだけかかわりを持たないようにしてます

選択肢 (1) : vs Perl

- 好きだった
- スクリプティング言語の良さを教えてもらった
- どうにでも書ける、というのはちょっと嫌
 - Perl4
 - Nifty4U は沢山の人に使ってもらえた
 - 今更な感、OOでない
 - Perl5
 - 素直なOOでない。Rubyの決定的な勝ち

選択肢 (2) : vs Java

- コンポーネントはC++よりある感じ
- お手軽度が決定的に足りない
(ニーズが違うから仕方がない)
 - 例外のcatchを強制される
 - スレッドが面倒
 - 正規表現がない
 - 日本語が下手
 - new Hoge より Hoge.new かな
 - RMI等は、Rubyにも欲しい

3 開発してみても

- やっぱり極めて書きやすい
 - 生産性高い (Perlよりよいぞ。OOだから)
 - 仕事でもプロトタイプ作成に利用
 - 分析結果をすばやく検証できる
- どうでもよいと思ってたけど、
実は結構よかった^^機能
 - 1 定義が動的
 - 2 特異メソッド
 - 3 制御構造がオブジェクトを返す

3-1 定義が動的

- 定義も実行文も同じ
- 例えば、既に定義済みのクラスに、メソッドを自由に追加できる
- クラスの機能拡張が容易かつ当該スクリプト中に閉じた変更が可能
 - 標準ライブラリや、人様の作ったクラスに(失礼ながら)機能不足を感じても、気軽にメソッドを追加出来る

3-2 特異メソッド

- インスタンスに固有のメソッドを定義出来る
- ホワイトボックス型フレームワーク
 - サブクラスを定義して、インスタンス化
 - システム中に一つしか要らないインスタンスの為にサブクラスを定義しなければならない
- 今回使ったWebアプリ用のフレームワーク
 - Pageクラス、Requestクラスのインスタンスは画面、要求毎に一つだけ存在すればよい

特異メソッドを使わなければ

```
class PageX < Page # サブクラス定義
  def form(session)
    # 固有のメソッドを定義
  end
end # サブクラス定義終了
p = PageX.new # インスタンス化
...
```

特異メソッドを使えば

```
p = Page.new # インスタンス化
```

```
def p.form(session)
```

```
  # 固有のメソッドを定義
```

```
end
```

...

- サブクラスを定義しなくてよい
- 実際は大した差ではない
 - が、それが気持ちよい^^場合がある
 - が、それが開発にとっては重要

3-3 制御構造が オブジェクトを返す

- ページ生成で利用

... + if i.url then

```
"<A HREF='#{i.url}'>#{i.name}</A>"
```

```
else
```

```
  i.name
```

```
end + ...
```

4 今後の展開

- 現在 0.93 のベータ版をRAAにて提供中
- 現在の機能にほぼ満足しているが、機能拡張の要求は多い
 - 時刻も調整しよう、日と場所で表にする等
 - eRuby化、国際化、見栄えをよくしたい
 - しかし、今のMeeting2000は保守性が低く、機能や画面の追加が難しい
- グループウェアサーバ構想
 - その上で次期Meeting2000を構築する予定

グループウェアサーバ

- グループウェアに対するニーズは高い
- でもちまたのCGIスクリプトは物足りない
 - ユーザ登録、掲示板や更新情報通知システム、ランキング物。
 - もっともっと便利なアプリケーションを!
 - ユーザ、グループによる情報の選別、共有
- グループウェア用のアプリケーションサーバが欲しい
 - アプリケーションサーバって???

アプリケーションサーバ

- アプリケーションを開発するためのフレームワークであり、且つアプリケーション群を管理するサーバ
- クライアントがWebブラウザであるアプリケーションが対象 (Webアプリケーション)
- HTTPサーバ、データアクセス、トランザクション、セッション、サーバ分散、セキュリティ等の様々な機能を有する
- OAS(オラクル)、WebSphere(IBM)、NAS(Netscape)、INTERSTAGE(富士通)等

欲しいもの

- 製品版は
 - 高い。買えないって。
 - 機能が多すぎる。そんなにいらない。
- 作りたいものは
 - Webベースのグループウェアが対象のアプリケーションサーバ
 - 日曜プログラマ(僕)が手軽に使える
 - 機能は最小限で、簡単、わかりやすい

現在

- GroupWareServer (仮称) 0.0.2 作成中
 - ユーザ、グループ管理機能
 - ユーザクラス、ログイン、設定変更画面等
 - 排他制御、データのロード、セーブ
 - プレゼンテーション層とビジネスロジック層の分離
 - リクエスト処理オブジェクトとページレイアウトオブジェクトの分離
 - まだ人様に使ってもらえる段階ではない
- 御意見など頂ければ歓迎

5 まとめ

- Ruby は一見して便利、使ってやはり便利
 - 試してみたい
- Meeting2000も便利、でも保守性が悪い
 - グループウェアサーバを作成中
- Rubyについて思うこと
 - 本を! (まつもとさんへ)
 - アプリケーションを! (Rubyユーザへ)