

Ruby on Rails による ITSS 診断システムの実装

串松慎太郎[†] 前田陽平[†] 櫻井一欽[†] 大谷真[†]

湘南工科大学情報工学科[‡]

1. はじめに

Ruby on Rails (以下 Rails と記述) はオブジェクト指向言語 Ruby 向けの MVC モデルベースのフレームワークである。DRY (Don't Repeat Yourself) と CoC (Convention over Configuration) が特徴であり、高い生産性を持つと言われ、一方インタプリタ言語であるが故に中大規模システムでは性能上限界があるとも言われている。

現在様々な分野で使用されつつあるが具体的な評価は十分とはいえない。本研究では中規模 Web システムとして開発中の ITSS 診断システムを例にとり、同等機能のシステムを Rails を用いて開発し生産性と性能を具体的に評価した。

2. ITSS 診断システム

2.1 ITSS (IT スキル標準)

IT 技術者に必要なスキルの集まり、及び各スキルの習得レベルから該当者の技術レベル (7 段階) へのマッピングを定義したもので、2002 年に経済産業省が標準化した。ITSS ユーザ協会がこれを細分化し、ITSS_V2_2006 では階層で表現すると 7312 スキル項目が定義されている。本研究ではこのうち 5337 項目を使用した。

2.2 ITSS 診断システム

ITSS 診断システムは、利用者が各自の習得レベル回答を入力・蓄積し、技術レベルを判定する Web アプリケーションである。木構造で表現されたスキル項目だけでも約 5000 のレコードを持ち、利用者増加とともに回答データも大幅に増加する中規模のアプリケーションである。通常は Java や C# などを用いた Web アプリケーションとして開発するのが妥当であり、現在 Java と Struts を用いて実装がすすめられている [1]。本研究ではこれと同じデータベース (以下 DB と記述) [2] を用い、同等機能を実現するシステムを Rails で開発した。

3. Rails による ITSS 診断システム実装方針

本研究の主目的は Rails の生産性と性能の評

価である。このためアプリケーション個別の最適化を避け、Rails 標準の規約・機能を極力そのまま使用の方針とした。

(1) O/R マッピング

Rails の規約に従いテーブル名を付け、クラスを作成することにし、Rails による O/R マッピングをそのまま使用することにした。

(2) DB アクセス

DB へのアクセスは SQL を書くのではなく Rails に標準装備されているメソッドを利用し、SQL の発行は Rails に完全に任せることにした。

(3) テーブル間の関係

このシステムを構築するために 14 個のテーブルを使用する。テーブル間にはそれぞれ 1:多などの関係があり、その定義は ActiveRecord の標準機能を使用することにした。

(4) RDB による木構造の表現

スキル項目定義データは各スキル項目が自分の親ノードへのポインタ、自分に従属するノードを表す対の値を持っており、入れ子集合モデルと隣接リストモデルを足した構造になっている。この構造は直下の子供を探す時には隣接リストモデルの特徴を使用し、あるノード以下にある全てのノードを探す時には入れ子集合モデルの特徴を使用すると便利である [3]。そこで Rails の ActsAsNestedSet と BetterNestedSet を使用することにした。

(5) 画面表示

スキル選択画面はテーブルに格納されている木構造を整形し動的に表示させることにした。これによりスキル項目定義データが変更された場合でもプログラムの変更が楽になる。

判定結果表示画面も一部を除き動的に表示することにした。これは判定項目定義データが変更された場合対応するためである。

画面に表示させる入力フォームやリンクなどはヘルパーメソッドを使用した。

4. Rails による ITSS 診断システムの実装

O/R マッピングは表 1 の例のようにマッピングしたいテーブル名とモデル名を Rails の規約に従いモデル名を複数形にし、小文字化、アンダ

Implementation of ITSS Judgment System over Ruby on Rails
Shintaro Kushimatsu[†], Yohei Maeda[†], Kazuyoshi Sakurai[†] and
Makoto Oya[†]
Department of Information Science, Shonan Institute of
Technology[‡]

ースコア化して定義することにより実現した。

DB へのアクセスは Rails 及びプラグイン標準装備のメソッド使用し、SQL を直接かくことなくやりとりを行った。表 2 に一部を示す。

表 1. O/R マッピング例

モデル名	テーブル名
SkillLayer	skill_layers

表 2. メソッド使用例

メソッド名	使用例
find	SkillLayer.find :all
root	SkillLayer.root
children	@skill_layer.children

テーブル間の関係定義には belongs_to や has_many を使用することにより実現した。employees テーブルと rank_answers テーブルに 1 対多の関係を持たせるモデルへの定義を図 1 に示す。

木構造を扱うために図 2 のように acts_as_nested_set メソッドをモデルに定義した。

システムの画面遷は図 3 のようにした。ユーザは login 後メニューページから回答、判定、結果表示を選択し操作する。

```
class Employee < ActiveRecord::Base
  has_many :rank_answers
end

class RankAnswer < ActiveRecord::Base
  belongs_to :employee
end
```

図 1. テーブル間の関係定義例

```
class SkillLayer < ActiveRecord::Base
  acts_as_nested_set
end
```

図 2. 木構造の定義

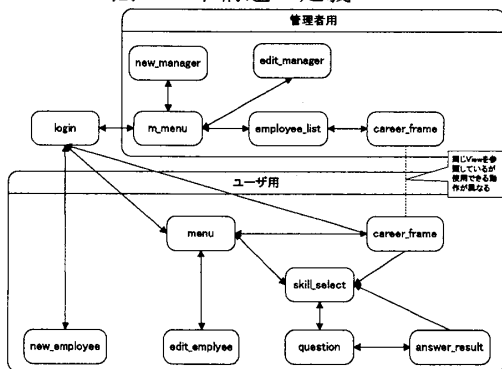


図 3. ITSS 診断システム画面遷移図

画面にテーブルなどを表示させる場合は直接 HTML を記述したが、入力フォームやリンクなどはヘルパーメソッドを使用した。

入力値の検証は標準で用意されている validate メソッドを使用した。カタカナの検証はなかったので自作し補った。

ユーザがこちらの意図しない順序で操作をした場合を考慮し判定処理を行うには回答が前提条件、判定結果を表示するには判定処理が前提条件とした。また login せずに URL を直接入力された場合に対しても before_filter を使用し使用できるアクションを制限した。

5. 評価

5.1 生産性

本研究で開発した Ruby/Rails 版と並行開発した Java/Struts 版の開発コード行数を表 3 に示す。約 1/5 で済んだことが分かる。また Ruby/Rails 版は非専任学生 4 人月程度で開発できたことも特記しておく。

表 3. LOC 比較

Ruby/Rails 版	669
Java/Struts 版	3106

5.2 性能

DB 量による処理増大が顕著なスキル選択表示時間の測定結果を図 4 に示す。なお測定環境は Ruby 1.8.6, Rails 1.2.6, JDK 1.5.0_14, Struts 2.0.11, PostgreSQL 8.2.5, Windows/XP, Pentium 4 (3.0GHz), MEMORY 2040MB である。Rails 版では登録人数が 40 人程度までの小規模組織が実用限界であり、一方 Struts 版は登録人数が増加しても実用上十分な性能となっている。参考としてインデックスを張った Rails 版も示しておく。

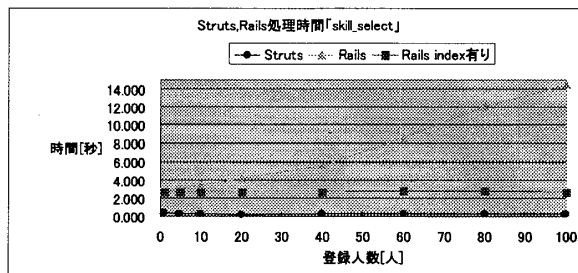


図 4. スキル選択表示時間

6. まとめ

本研究では通常 Java/Struts などで開発する中規模 Web アプリケーションを Ruby/Rails で開発した。この結果 Rails がこの程度の Web アプリケーション開発に十分な機能を持つこと、及び高い生産性を持つことが判明した。一方中大規模 DB アクセスを伴うアプリケーションでは、現時点、及び現構成では Rails は十分な性能を満足できない。今後の Ruby と Rails の高速化に期待したい。

謝辞

本研究に関して ITSS ユーザ協会からスキルコンテンツの提供を受けました。

参考文献

1. 前田陽平、ITSS 診断システムの開発、IPSJ 第 70 回全国大会、2008
2. 櫻井一欽、ITSS 診断システムの DB 設計、IPSJ 第 70 回全国大会、2008
3. Joe Celko、プログラマのための SQL 第 2 版、ピアソンエデュケーション、2001
4. D. Thomas et al、Rails によるアジャイル Web アプリケーション開発、オーム社、2006