

# Ruby on Rails による 求人票閲覧システムの開発

岡山県立水島工業高等学校進路指導課

大島 修

国産言語である Ruby による Web アプリケーション開発用フレームワーク Ruby on Rails を使い、生徒の就職指導に使う求人票閲覧システムを開発した。このシステムでは、生徒は事業所名の一部の文字（ひらかなでよい）を入力して検索し、出力されたリストの中から希望する職種の表示をクリックすれば、求人票を PDF ファイルとして画面に表示し印刷することができる。校内の公開系 LAN に接続されたすべてのパソコンおよび行政系 LAN の一部の IP アドレスを持つパソコンから Web ブラウザーを通して自由にアクセスできる。使ったサーバ OS は Debian GNU/Linux で、データベースには MySQL を使い、Ruby on Rails も Web サーバもすべてオープンソースで構築した。Web アプリもデータベースのソフト開発も初めてという筆者が取り組んだ様子を報告する。

## 1. はじめに

H19 年 3 月末の異動赴任先の本校において、校長面接でいきなり進路指導課のシステム化を担当して欲しいと言われたのが取り組みの始まりである。

これまで水工では就職を希望する生徒は、求人概要を集約した「求人台帳」を渡され、その中から希望する求人票の受付番号を担当教員に告げてコピーしてもらっていた。大判のクリアファイルに入った求人票をコピーするこの作業は、求人票の枚数が多いと結構大変だった。パソコンで求人票を表示できれば、そのままプリンターから出力でき、随分省力化が図れることになる。

これまで私は、データベースソフトを使った Web アプリケーションを仕様を決めて発注する側であったが、開発した経験はない。しかしこのようなシステムの開発には興味があったので、二人分の給料をもらわないとやってられないなあとぼやきながらも苦労しながら開発に取り組んだ。

また、生徒に渡す「求人台帳」は、ある程度まで求人票の数がたまったところで、紙に印刷・製本し配布する。生徒は、配られた求人台帳を見て目的の求人の目星を付け、実際の求人票のコピーをとって自宅に持ち帰るといった仕組みである。この方法だと、求人件数がある程度まとまらないと公開できないため、求人票が届いてから生徒が知るまでにタイムラグが生じる。

この点、システムをオンライン化すると、求人 1 件ごとに入力作業が終わると、即公開されていくため、タイムラグが生じないというメリットがある。

これまでのエクセルによるシステムは、求人台帳作成以外にも、推薦会議資料作成や応募と結果、さらに年度末の統計処理まで、進路課に必要なすべての機能が備わっている。これらすべてを一度にオンライン化するのは困難であり、とりあえず求人票の受付と閲覧に関する以下の機能に絞って開発することにした。

## 2. 従来の進路課業務とシステム化

これまで進路課で使ってきたエクセルのマクロを利用したシステムは校内教員による手作りであるが、何年もかけて改良を加えてきたもので、必要な機能はすべて含まれている。しかし、これを扱う進路課員は、業務のこともソフトの操作方法もともに知らないと、手が出しにくい物である。これまででは習熟した進路助手一人が一手に引き受けていた。また、エクセル内のデータが毎年増えることで、処理が年々重く遅くなっていくのも、改善すべき点であった。

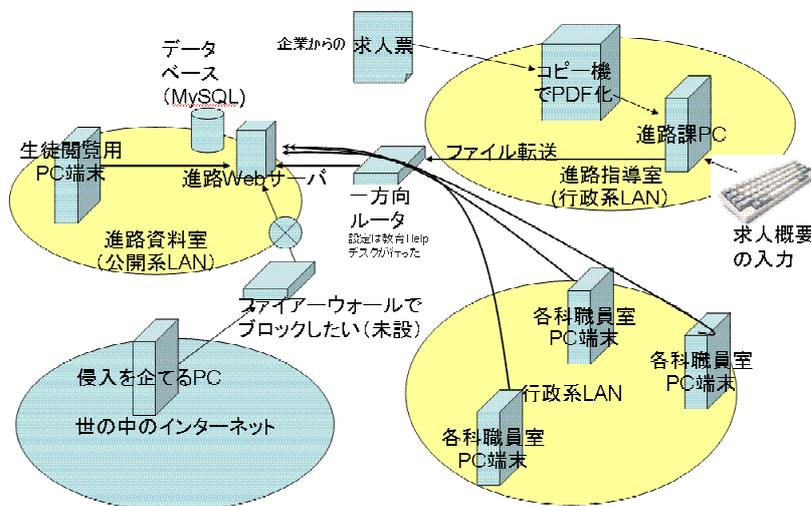


図 1. 求人票閲覧システムの概要

### 求人台帳機能

求人概要を一覧でき、希望に合った各種条件で検索できること。

### 求人票検索表示機能

企業名や希望の条件で検索し、求人票を表示できること。

### 企業台帳機能

求人内容は毎年内容が変化するが、企業自体に関する情報はほとんど変化しないので、求人案件とは別のデータとして保存できること。

### 分散入力機能

求人台帳に求人要件の概要をまとめるためには、多くの情報を人間が手で入力しなければならないが、一度に多数の PC から分業で入力できるように、Web 上で行えるシステムにした。

## 3. システムの構成要素

### (1) ハードウェア編

#### コピー機: Canon iRC2570F

求人票を PDF ファイルとして読み込み、PC へ転送する。重ねて置いた印刷物を連続的に PDF 化できる。

### Web サーバ

データベースと検索・閲覧・データの入力のサービスを Web として提供する。生徒がアクセスできるように公開系 LAN 上に置く。校外からのアクセスはファイアーウォールで遮断できるようにする。

### 校内 LAN の一部改変

県立学校では、安全上の配慮から、校内 LAN の構成は行政系と公開系とは直接には接続できない仕組みになっている。今回、上記の Web サーバに教員が直接アクセスできるようにするため、行政系から公開系への一方向のみのアクセスを許すルータを新設した。その設定については、教育ヘルプデスクの原田主事（総合教育センター）のご好意により、本校まで出張してきていただき、お手を煩わせた。ご協力に感謝します。

### クライアント用 PC

公開系 LAN に設置する端末は、ごく普通にインターネットにアクセスできる Web ブラウザー（IE やファイアーフォックスなど）が使えればよく、特に高性能なマシンは不要である。

### (2) ソフトウェア編

基本方針として、すべてオープンソースソフトを利用し、ソフト購入予算はゼロにした。

### サーバ OS Debian GNU/Linux 4.0r3 etch

採用した Linux は、ライセンス料が不要、かつ高信頼性高性能で、Windows 系 OS では心配の種であるウイルスなどもほとんど無縁である。Linux は、今では、みずほ銀行の勘定系基幹サーバや東京証券取引所など絶対的な信頼性が要求される業務にまで本格的に使用されるようになってきている。

### データベース MySQL 5.0.27

これもオープンソースソフトで、高速応答、軽快なデータベースとして定評がある。

### Web サーバソフト

開発時には Ruby on Rails (後述) に付属している Webllick を使用した。本番環境では Apache か Lighttpd を利用する予定であったが、Webllick のままで結構軽快な応答が実現できているのでそのまま使用している。

### Web アプリ開発

使用した言語は、Ruby 1.8.6 で、Web システム開発用にはフレームワークとして Ruby on Rails 1.2.6 を使用した。Ruby on Rails について（以下、Rails と略称）は、最近は一大ブームといってもよいくらいにソフトウェア業界では有名になっている。これまで Web システムを開発するための言語としては、JAVA や PHP が有名である。しかし、最近は使いやすさで世界的に有名になっている優れた国産言語 Ruby がある。これは松江市在住のまつもとゆきひろ氏が 1993 年に公開し、以来オープンソースとして世界中の人が協力して改良を加えている言語である。その Ruby で書かれた Web システム開発フレームワークが Rails である。Ruby の優れた特徴を生かし、記述量も開発時間も JAVA の 1/10 で開発可能という評判がある。使ってみて、Rails は確かに DRY 原則 (Don't Repeat Yourself = 同じ記述を繰り返さない) を大切にしたフレームワークで、JAVA や PHP で開発する場合よりもスマートに作られている先進的なシステムである。ただ先進的であるが故の問題点もあり、これについては後述する。

## 4. システム開発の実際

このシステムを構築するにあたり、最初は簡易データベース Access を使ったシステムにするか、本格的なデータベースを使った Web 系システムにするか、判断に迷い、両方の参考書を数冊ずつ購入し勉強した結果、後者を採用することにした。その 1 番の理由は、Web システムにすると、ユーザー（クライアント）側 PC には Web ブラウザーだけが動けばよいので、OS の種類もマシンの性能も選ばず、楽であること。2 番目に、メンテナンス性がよいことで、サーバ側のアプリさえ入れ替えれば、クライアントが何台あろうともまったく無関係にシステムをバージョンアップができる。Access のようにクライアントにソフトをインストールするタイプでは、全台のソフトを 1 台ずつ入れ替える必要があり、少しずつソフトを改良していくという開発スタイルはとりにくい。3 番目に予算がかからない。すべてサーバ側のアプリは、オープンソースソフトで構築できるので無料であるし、何年かごとにマイクロソフトの都合に合わせてバージョンアップのために OS や Office ソフトを買い換える必要もない。

### MVC モデルについて

Rails は、Web アプリの構造としていわゆる MVC (モデル・ビュー・コントローラ) を採用している。

## 画面の表示を動的に変えるビュー(V)

ホームページの表示を動的に変更する役割。普通の Web ページは、あらかじめ用意された画面を必要に応じて切り替えて表示しているだけであるが、Web アプリの場合は、プログラムにより必要な表示を柔軟に書き換えて表示する。それを行うには一般に JavaScript という言語がよく使われるが、Rails では Ruby が JavaScript を自動で生成するようになっている。画面のデザインを一貫したものに統一する仕組みは、CSS というやり方を使う。このあたりも私にとっては初めてで、1 から勉強を行った。

## データベースとデータをやり取りするモデル(M)

一般的には、データベースとデータをやり取りするには、SQL という言語を使う。Rails では、データベースのテーブルをオブジェクトとして扱い、その出し入れや加工に Ruby 言語を使う。つまり、SQL を知らなくてもデータベースが操作できるというたい文句である（実際には、データベースと SQL を知らないで Web アプリを構築できるほど甘くはないことが、後でわかってくるのだが、...）。

## 全体を統括するコントローラ(C)

利用者の指示に応じて、画面を切り替え（作り変え）、データを出し入れし加工する働きは、コントローラが担当する。これにも Ruby が使われる。Ruby は文字列処理や時間表示、配列やハッシュといった処理が得意な言語であり、しかも少ない記述量で、プログラマが楽しく行えるという特徴があるので、このようなシステムには最適な言語である。ただインタープリター言語なので、比較的動作が遅いのが欠点であるが、今回のシステムでは、一度にアクセスするユーザーは最大で数十人程度であることを考えれば、十分な処理速度を持っている。それよりも、コードの記述がわかりやすく少ない量で済むなど、開発の容易さの点で優れている。

## 困難を極めた開発作業

何度も書くが、私は Web アプリを開発したことがない。データベースを使った開発どころか、データベースを SQL で直接操作したことさえない。Excel からデータを移行する場合、どこをどう変えればいいのか、そこでテーブルの「正規化」という作業を行う必要があることも初めて知った。

伏兵はそれ以外にも潜んでいた。選んだ Ruby on Rails は、流行り始めたばかりの「旬」なツールであり、どんどん改良されバージョンアップが頻繁に繰り返えされている。この文を書いている現在では、すべてを買い揃えられないくらいのたくさんの参考書が出版されているが、開発を開始した時点で入手

できた書籍は少なく、しかも執筆に時間がかかると

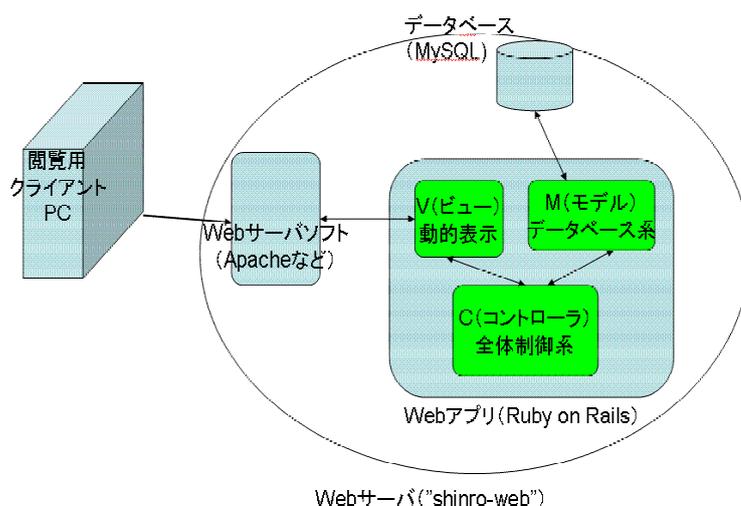


図2. Web アプリを構成する MVC モデル

Web アプリといっても、実際にはこのように役割分担を行い、それぞれの機能が協調して一体のアプリケーションとして動くように作られている。

というのが本の宿命で、その間に怒涛のように Rails の改良が進み、本が出版された時には、すでに新しいメジャーアップデートが完了しているという事態が起こっていた。

どう考えてもエラーの原因がわからず、さらに文化祭の時期には部顧問の仕事や、学年一のやんちゃなクラスの生徒たちが次々に問題を起こしてくれたりと数ヶ月間開発がストップしたままになったこともあった。50歳も半ばにかかる頭では、プログラム開発は、いったん作業を中断したら、再開するには同じ所から始めるのも難しく、最初から理解し直すくらいでちょうどよい。エラーが出る。通常の文法エラーなら、根気良くエラーメッセージに従って、プログラムのおかしい辺りを調べれば何とか原因を突き止め修正することができるのだが、もっと原始的な段階のエラーが起こり困惑したこともあった。本体とライブラリーのバージョンが合っておらず、必要なライブラリのソースを入手し、自分でコンパイルを行ったこともあった。情報が古く、新しいバージョンでは別な記述法にしないとイケないとか、参考にした本のプログラムリストにミスプリントがあったりなど、5重6重の困難にくじけそうだった（これらの問題は、現在ではだいぶ落ち着いてきているようである）。この間、進路課の会議で「今月末までには、何とか皆さんにお披露目できるものにします」という約束を何度延長しただろう。

## 5. システム開発の現状と今後の方向

現在、当初予定した目標派はほぼ実現でき、生徒が進路資料室の PC から求人票を検索したり、印刷し

たりと実用的に利用している。

(1) 本番環境の整備

現在は、開発できた機能から公開して使われているが、その「サーバ」には開発に使ったPC（パーツから組んだ自作PC）をそのまま使っている。本格稼動するには、以下のようなハードウェアの整備が必要となろう。

校外からの侵入に備えて、アクセスしてきたものが校外か校内かを識別し、校内のものだけを接続する専用ファイアーウォールを置く（現在は、サーバ機内のソフト機能で対応）。多重アクセスに対して、ロードバランサーを置き複数のサーバプロセスに分散し、過負荷を避ける。ハードディスクは、クラッシュに備えてミラーリング構成とする。また、停電対策として、無停電電源を置く、など。

(2) 機能の進化

次年度以降は、これまでの Excel により行っていた残りの機能、つまり、生徒が求人票を見て就職試験に望むまでの手続きや校内就職推薦会議に向けた資料の作成と出力、統計処理など担任や進路指導課の教職員の業務をシステム化することが求められるであろう。

また、求人案件の概要をまとめるためには他の多くの情報を人間が手で入力しなければならない。今後の希望として、職業安定所には、求人票を電子データとして学校に配信していただければありがたい。

参考文献

- (1)「RailsによるアジャイルWebアプリケーション開発」第2版、D. トーマス、D. ハンネマイヤー・ハンソン、オーム社
- (2)「基礎 Ruby on Rails」黒田 努 佐藤 和人、インプレス
- (3)「Rails レシピブック 183の技」高橋 征義・諸橋 恭介、ソフトバンク
- (4)「Ruby on Rails 逆引きクイックリファレンス Rails 2.0 対応」大場 寧子他著、毎日コミュニケーションズ
- (5)「Ajax on Rails」、Scott Raymond (著)、牧野 聡 (翻訳) オライリー・ジャパン
- (6)「Ruby レシピブック 第2版 268の技」青木 峰郎、後藤 裕蔵、高橋 征義、まつもとゆきひろ、ソフトバンク
- (7)「たのしいRuby 第2版」高橋 征義 後藤 裕蔵、ソフトバンク
- (8)「標準MySQL 改訂第3版」田中 ナルミ (著)、阿部 忠光 (著)、ソフトバンク など

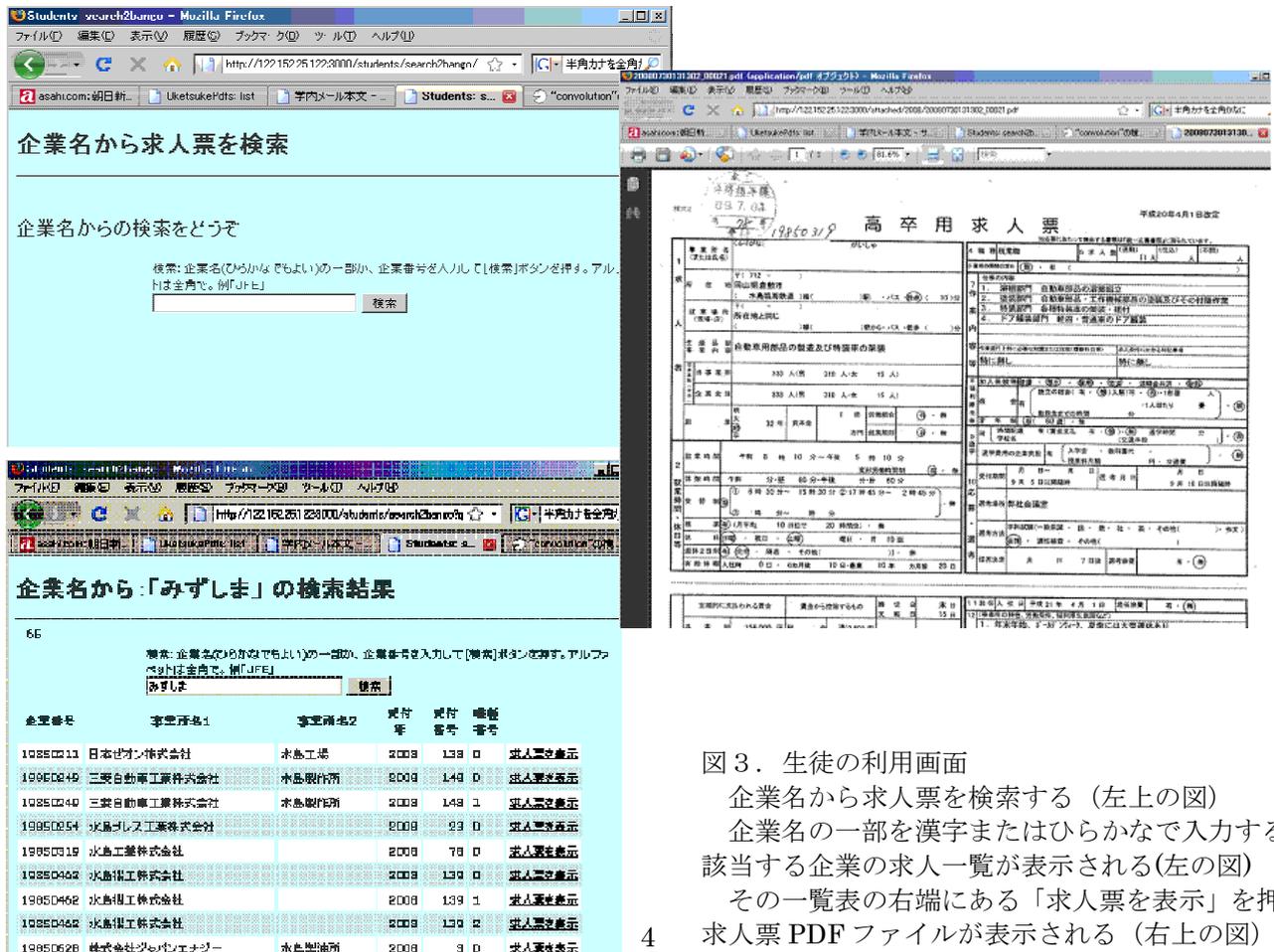


図3. 生徒の利用画面

企業名から求人票を検索する（左上の図）  
 企業名の一部を漢字またはひらがなで入力すると、該当する企業の求人一覧が表示される（左の図）  
 その一覧表の右端にある「求人票を表示」を押すと求人票 PDF ファイルが表示される（右上の図）