



2008 年度下期未踏 IT 人材発掘・育成事業 採択案件評価書

1. 担当PM

竹内 郁雄 PM (東京大学大学院 情報理工学系研究科 創造情報学専攻 教授)

2. 採択者氏名

チーフクリエイター: 松本 智行 (フリー)

コクリエイター: なし

3. プロジェクト管理組織

株式会社メルコホールディングス

4. 委託金支払額

3,000,000 円

5. テーマ名

世界一短いコードで web アプリ作成ができるフレームワーク開発

6. 関連Webサイト

<http://web4r.org/>

<http://github.com/tomoyuki28jp/web4r>

7. テーマ概要

世界一短いプログラムの記述量で web アプリケーションを作成をすることができる Common Lisp の web アプリケーションフレームワークを開発します。ここで言う”フレームワーク”とは、HTML 生成、入力値のバリデーション、オブジェクトデータベース、認

証管理など、web アプリケーション開発に一般的に必要とされる全ての機能を All in one で提供をするフルスタックのフレームワークを意味します。上記の機能に加え、HTTP サーバーも実装します。

8. 採択理由

提案概要も恐ろしく短かったが、世界一短いとはキャッチーな言葉である。ダイナミックな Web ページの生成に Lisp がとても便利であることは、Allegro Common Lisp を販売している Franz 社の Kunze 社長から数年前に聞いたことがある。Lisp を使って Web ページを作っていたある顧客が「Lisp を使って我々が成功していることを公開しないでほしい。なにしろ、たった 2 人でほかの 10 人規模の会社と同じ売上を上げているのだから…」そのため、この事例は 1 年間だけ非公開にしたとか。少々出来すぎた話だが、もともと Lisp 屋である竹内には信じられない話ではない。

Lisp には不思議な魔力があるようで、フリーランスの松本君が Lisp を使い始めたのはまだほんの 1 年前なのに、すっかり取り憑かれたようだ。オーディションでもそれが如実に現れた。もちろん、ゼロからの提案ではなく、すでにあれこれの部品が積み上がってきている。Lisp らしいやり方だ。素でコード量が 1 桁小さくなるというのだが、具体的な話はどれを聞いても首肯ける。HTML などのマークアップ言語は Lisp の S 式のちよいと太った子孫と言っていいからだ。

竹内の期待を、松本君はぜひ成果物として裏付けてほしい。Lisp に取り憑かれた人には竹内を含めちょっと怪しい人が多いのだが、松本君はいたって元気で威勢がいい。ダイナミックな Web コンテンツをつくるには LAMP (Linux、Apache、MySQL、Perl + PHP + Python) がいいという話があるが、最初の L に Lisp を参加させるべく、ぜひ頑張ってもらいたい。

9. 開発目標

シンクライアントが進む昨今、web アプリケーション作成の需要は益々増大するばかりである。web サービスの開発・運用をする為に、必要となるコストがコンピューター機器等への設備投資と、人件費である。ムーアの法則に伴ったコンピューター機器の計算処理能力向上と低価格化により、必要な設備投資額は年々下がり続けており、人件費の占める割合がどんどん大きくなっているのが現状である。

本プロジェクトは、既存のものよりも飛躍的に短いプログラムでwebアプリケーション作成が可能となるフレームワークを開発することにより、プログラマーの生産性向上、開発・保守費用等の人件費削減、納期の短縮等の効果をもたらすものである。

10. 進捗概要

Common Lisp という言語で web4r という名前の web アプリケーションフレームワークを開発した。ここで言うフレームワークとは、HTML 生成、入力値のバリデーション、データベース操作、認証管理など、web アプリケーション開発に一般的に必要とされる全ての機能を All in one で提供する、フルスタックのフレームワークを意味する。加えて、実用化をする為に、単体、結合、メモリ消費量、実行速度等のテストも行った。

11. 成果

本プロジェクトの主な目標は次の 3 点である。なお、プロジェクト開始時点で web4r のプロトタイプは既に開発済であった。

- (1) 世界一短いプログラムで web アプリケーション作成が出来るようにする
- (2) 実用化をする
- (3) web 上で公開する

目標の達成結果は次の通りである。(1)の世界最短は計測が難しい問題ではあるが、特定の要件を満たす web アプリケーションの実装コードを比較した結果、web4r の解は公開されている汎用プログラムの中で最短であった(詳細は後述)。(2)の実用化と(3)の web 上での公開は、当初予定を前倒して達成することができた。

具体的な開発と作業の内容は以下の通りである。

- ・ web4r のプロトタイプに対して、ユーザー認証管理や、マークアップ言語生成 (XML、HTML、XHTML が生成可能)、クライアントサイドバリデーション、web アプリケーションのテストツールなど、web4r の実用化に必要であった他多数の機能を実装した。
- ・ 単体テストと結合テスト合計で 615 個のテストケースを作成し、それらすべてのテストが後述のサポートプラットフォームで合格することを確認した。
- ・ オープンソフトの Common Lisp 処理系である SBCL を利用して、html の p タグで数値を出力するだけの継続手続きを登録した場合、ひとつの継続手続きのメモリ消費量は約 29,345 バイトであった。(メモリ消費量は Common Lisp の room 関数でだまかに計測。)これは 1GB のメモリスペースに登録できる継続手続きは約 36,590 個ということを意味する。ひとつの継続手続きあたりのメモリ消費量を減らすことは難しいが、実運用上問題のない形で不要な継続手続きを破棄する方法は存在するので、今後の課題として取り組むものとする。

- ・ 実行速度を測定する目的で、HTTP サーバーとフレームワークのベンチマークを行なった。その結果を表 2.3.1 に示す。ベンチマークの実行環境と実行結果は次の通りである。

ハードウェア

CPU: 1.20GHz Intel Core 2 Duo

Memory: 4GB

Storage: 64GB SSD

ソフトウェア

Ubuntu 9.04 32bit

SBCL 1.0.18

Apache 2.2.11

Hunchentoot 1.0.0

PHP 5.2.6

リクエスト数と同時接続数

リクエスト数: 100

同時接続数: 10

表 2.3.1 ベンチマーク実行結果 (単位は秒)

	2 Bytes	477 KBytes
Hunchentoot 単体	3.001	4.561
Apache + mod_proxy + Hunchentoot	0.108	5.694
Hunchentoot + web4r	3.000	12.376
Apache + mod_proxy + Hunchentoot + web4r	0.175	13.875
Apache + mod_php + php (比較対照として計測)	0.099	5.102

- ・ ダウンロード方法や、変更履歴 (CHANGE LOG)、インストール方法、チュートリアル、API マニュアルなどのドキュメントを日本語と英語で作成した。
- ・ バージョン管理ツール git のレポジトリや、作成したドキュメント、及びデモを web 上のプロジェクトページで公開した (図 2.3.1)。



図 2.3.1 <http://web4r.org/> トップページ (日本語版)

- web4r の依存パッケージがすべて動作する Common Lisp の処理系が SBCL と Allegro Common Lisp のみであったため、開発環境であった Linux x86 32bit 上の SBCL 1.0.18 に加え、Allegro Common Lisp 8.1 でも動作するように修正した。

完成した web4r 最大の特徴は、既存のものよりも短いプログラムで web アプリケーション作成ができることである。ここではトークンの数をプログラムの長さとして、文法として扱われるカッコやセミコロン等は、トークンとしてカウントしないものとする。

web4r を利用した場合、どれだけプログラムが短くなるか、実例を挙げて説明をする。

Paul Graham 氏が出題をした、Arc Challenge と呼ばれる、web アプリケーションの実装コードを比較するためのシンプルな問題が存在する。Arc Challenge の要件は次の

通りである。まず 1 ページ目でテキスト入力欄と送信ボタンを表示する。送信ボタン押下時、2 ページ目で 3 ページ目へのリンクを表示する。リンク押下時、3 ページ目でテキストエリアの入力値を表示する。但し、入力値を URL で渡してはいけないものとする。以上が Arc Challenge の要件である。この問題に対して、Arc Forum には世界中から 70 以上の回答が寄せられたが、web4r の解は 2009 年 7 月 24 日時点で公開されている汎用プログラムの中では最短であった。なお、フレームワーク自体のコードが公開されていなかったり、汎用性のない実装方法の場合、有効な回答と見なさないものとする。

より実用的な web アプリケーションで比較をするために、Blog アプリケーションを例として用いた。要件は次の通りである。Blog にはタイトルと本文が存在する。この web アプリケーションは Blog の新規作成、編集、削除、詳細表示を行なうことができる。タイトル、本文ともに未入力は許可せず、指定文字数以内でなければいけない。

近年流行している Ruby on Rails のバージョン 2.3 の場合、DB スキーマとモデル、コントローラの主要コードのみでも 253 個のトークンが必要になるが、web4r の場合、genpages という汎用マクロを利用すれば 10 個のトークンで実装可能であり、genpages マクロを利用しなくても 30 トークンで実装可能である。

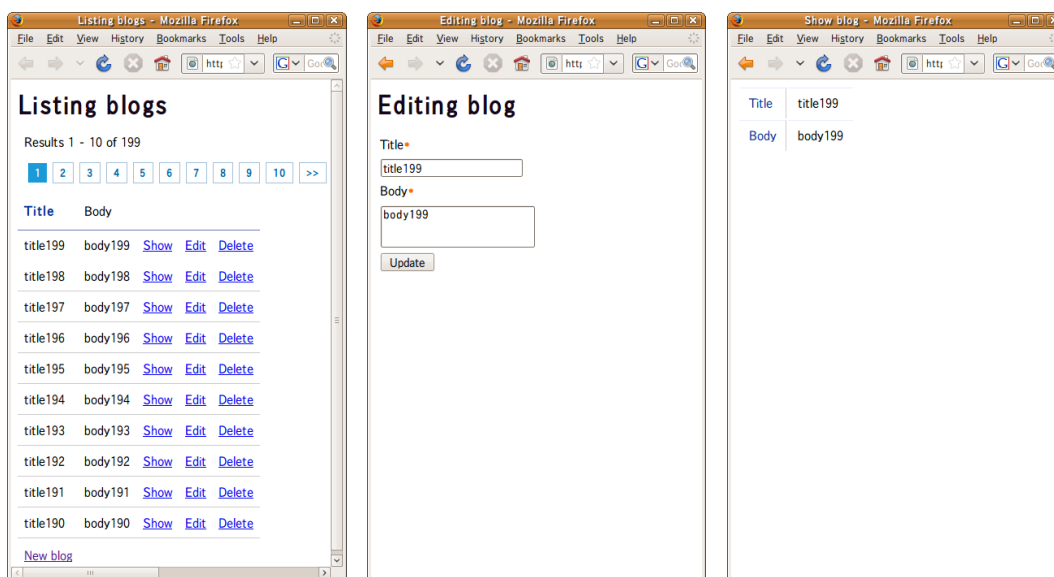


図 2.3.2 web4r で実装した Blog アプリケーション スナップショット
(左から順に一覧, 編集, 詳細ページ)

12. プロジェクト評価

こう言ったら失礼かもしれないが、松本君はまさに猿飛佐助を連想させる、実に活きのいい青年である。提案文書も短かったし、成果報告書も（当初は）激しく短かったのだが、レビューのときの発表や、未踏ユースでのいろいろな発表ではこれでもかこれでもかのプログラムリストのオンパレードなのである。これはプログラミングそのものが大好きな人の典型的なパターンと言えよう。

もっとも、報告書類が短いのは、web4rに関するWeb ページが充実しているのも、それで十分という合理的な発想に基づいているのだろう。20 歳で米国に留学、インターンなどを経て日本で起業、いろいろあって現在フリーランスという、半端ではない波乱(?)の人生の松本君ならでは、と言えないことはなさそう。実際、web4r に関する Web ページはプロジェクト当初からかなり充実していた。まず、英語のページを先につくり、それからそれを翻訳して日本語のページをつくるのだそう。こういう人は竹内の担当したクリエイター（開発者）の中では初めてである。実際、web4r のページアクセスやダウンロードは、最初のうち英語版が日本語版よりも多かった。最初から国際的である。

少なくとも未踏期間中はフリーランスだったこともあり、全精力をweb4rに費すことができ、かなり早く当初の計画が達成できてしまった。そして、約束通り、Arc Challenge の問題では、汎用フレームワークとしては世界最短、またより実用的な Blog ページは、Ruby on Rails に比べても圧倒的に短いトークン数（行数よりは正確な測り方）で記述できてしまった。ちょっとサンプル数が少ないと言えないことはないが、「フォーミュラ・ワンの競技会」としての最短競争では間違いなく世界トップの成績を達成した。

Lisp を学び始めて 1 年以内にここまでを成し遂げたというのは半端な才能ではないが、逆に Lisp だったからこそこれだけ短期間にこれだけのことを成し遂げることができたのかもしれない。実際、書かれた Lisp プログラムの行数は、機能上の見掛けに比して、わずか 2~3000 行なので、決して多くない。つくづく Lisp はいい言語だと竹内はつい思う。むしろ、プログラムよりもドキュメント作成のほうの作業が多かったのではなかろうか。

上にも述べたように、web4r は主として記録競争のために開発されているが、その本質は、継続渡しのプログラミングスタイルなど、非常に強固な構造設計に基づいている。Web アプリの開発者には、ぜひその原理と美しさ、(現時点ではまだ潜在的な) 効率の良さを知っていただきたいと思う。「短く書ける」ということの背後には、よい設計原理が潜んでいるのである。Arc Challenge に対する一部の挑戦者の中には、そのような設計原理のない、単なるアドホックな短縮だけを目指した人達もいた。

ただし、web4r が実用的な Web 開発フレームワークと言えるようになるためには、Web ページデザインに関わる要素が必要である。現状では骨格の機能としては十分だが、デザイン的にはずいぶん素っ気ないページしか作成できない。このフレームワ

ークをベースにした、多くの Web アプリ開発者の心をくすぐるようなシステムの開発は別途の課題である。とはいえ、とても独りでできる規模の課題ではなさそうだ。

それでも松本君には、よく多くの人が web4r の魅力にトラップされるような（広い意味での）プレゼンテーションをするようにと、常日頃から進言していたのだが、最後まで「通」の心をくすぐるレベルに留まってしまった印象がある。これは竹内としては残念だった。

13. 今後の課題

文章や動画、講演などを通じての web4r の普及活動、web サービスのリリース。