OSS COLUMN

オープンソース事情

6 Ruby 成功の秘訣

まつもと ゆきひろ
matz@ruby-lang.org
（株）ネットワーク応用通信研究所

Ruby のはじまり

私は 1993 年から Ruby という名前のプログラミング言語（と，その処理系）を設計開発しており，それをオープンソースソフトウェアとして公開しています．

当時，私が勤めていたソフトウェア開発会社はあまり景気の良い状態ではなく，私の部署はあまり仕事がない状況が続いていました．Ruby の開発は，そのようなときの余暇のプロジェクトとしてはじまりました．高校生くらいのころから自分の言語をデザインすることにあこがれていた私は，大学でも言語処理系の研究室に所属していましたが，きちんと動くレベルの言語処理系を最後まで完成させたことはありませんでした．しかし，この頃，Perl に代表されるようなスクリプト言語に触発され，「自分言語」を作りたいという情熱が蘇りました．こうなると仕事があまりないというのはかえって好都合です．それからしばらく理想の言語の仕様を考えてはそれを実装するという作業が，ごく身近な知人以外には誰にも知られず続けられていきました．このときの開発動機は「自分の趣味にあった言語を作りたい」というものだけで，他人に使ってもらうことはほとんど念頭にありませんでした．

最初の目標は「Hello World」を出力することです．そのためには字句解析器，構文解析器に加え，文字列クラスと File 入出力クラスが必要です．このように必要なものを芋づる式に実装していきました．その年の夏にはごく簡単なプログラムは動くようになりました．

しかし，これだけではただのオモチャです．他の人に使ってもらうつもりはさほどなかったものの，自分のツールとして使う気持ちは満々でしたから，これだけで満足するわけにはいきません．その後，Perl の機能をクラスライブラリとして再構成し，実装し直す日々が続きました．また，実行環境として必要な機能である例外処理や，ゴミ集めなどの機能も実装しました．とりあえず自分が満足できるレベルまで実装し，NetNews で公開したのは 1995 年 12 月のことでした（**図 -1** 参照）．

コミュニティの形成

Ruby を NetNews に公開したのと同じ日にメーリングリストも用意しました．メーリングリストのアーカイブを眺めると最初の 1 通がバグレポート，続く 3 通がパッチという非常に「アクティブ」な開発状況が見受けられます．このメーリングリストの参加者は数カ月のうちに 200 人を超え，Ruby の開発に必要なバグレポート，提案，要望などを供給できるようになりました．

この後しばらくはあまりコミュニティの成長について意識することはなかったのですが，その間も数多くの人たちがメーリングリストに参加してくださり，1999 年 9 月には Ruby だけを取り扱うイベントとしては最初のものとなる JUS Ruby ワークショップが開催されます．その直後，Ruby を扱った世界最初の書籍である『オブジェクト指向スクリプト言語 Ruby』が ASCII 出版局から出版されます．当時，超マイナーな言語をライターとしては素人の私が書いた本でしたが，意外なことにコンピュータ書籍としては異例の売れ行きを見せ，その月の IT 関係ベストセラーに選ばれたりもしました．このあたりが Ruby がマニア以外の層に知られるようになってきたはじまりではないかと思います．

海外進出

もうずっと前のことになりますが，インターネットからダウンロードしてきた言語処理系（たしか Scheme だったと思います）のソースコードを眺めていて，技術的には非常に面白そうなのに，ドキュメントやコメントがロシア語で書かれていて，ソースコードが十分に理解できず，悔しい思いをしたことがあります．そのことが強く印象に残っていたので，海外の人が Ruby について同じ思いをしてはいけない，という強い思い込みから，ドキュメントを英語化することにしました．元々さほど英語が得意とはいえない私でしたから，ドキュメントの英訳にはとても苦労しました．日本語でもドキュメントを書くのは面倒なのに，英語となればなおさらです．それでも，上記の思い込みによる不可解な情熱だけを頼りに，下手くそながらもとりあえずの言語リファレンスを英語化しました．

すると，おかしなもので，海外からの問合せがぽつぽ

1993-02-24	Ruby 誕生
1995-12-21	Ruby 0.95，メーリングリスト開始
1996-12-25	Ruby 1.0
1997-07-18	開発者メーリングリスト分離
1998-08-13	Ruby 1.4.0
1998-12-02	英語メーリングリスト開始
1998-12-25	Ruby 1.2
1999-09-04	JUS Ruby ワークショップ
1999-10-27	最初の書籍『オブジェクト指向スクリプト言語 Ruby』出版
2000-04-28	comp.lang.ruby 設立
2000-09-19	Ruby 1.6.0
2000-10-31	最初の英語書籍『Programming Ruby』出版
2001-10-12	第 1 回 Ruby Conference（Florida 州 Tampa）
2002-05-10	英語開発者メーリングリスト開始
2003-08-04	Ruby 1.8.0
2005-12-24	Ruby 1.8.4

図 -1　Ruby の歴史

つと入るようになりました．最初に Ruby に関心を寄せてくれたのはドイツ人でした．彼の強い勧めで 1998 年に Ruby について英語で話し合うメーリングリストを開催しました．最初のうちは参加者もあまりなく，静かなものでしたが，1999 年から 2000 年にかけて流量が増大しました．また，2000 年には NetNews にも Ruby について話し合う comp.lang.ruby ニュースグループが設立されました．2000 年 10 月に Ruby に関する英語圏で最初の書籍『Programming Ruby』が出版されてからというもの，コミュニティへの参加者はうなぎのぼりで，2002 年 2 月にはとうとう日本語版メーリングリストのメール総数を英語版メーリングリストが追い越すという事態にまで至りました．今では，Ruby の利用者は海外の方がずっと多いように思われます．

定着および発展

このような経緯を経て，Ruby は Perl，Python，PHP に並ぶスクリプト言語として一定の存在感を獲得しつつありました．しかし，広く知られるようになったものの，「ライバル」である「P」で始まる言語と比較すると，特に海外では，今一歩人気で及ばないというのが実情でした．

しかし，ここ数年で状況が変化してきました．いまや Ruby を本気でビジネスに応用しようという企業が増加しつつあり，また，実力はともかく，話題性や人気において，Ruby はようやく他の「P」言語を凌駕しつつあります．きっかけは「Ruby on Rails」と呼ばれる Web アプリケーションフレームワークでした．このフレームワークは，Ruby の動的で柔軟な性質を余すところなく利用したうえで，非常に高い生産性を実現し，世間の注目を浴びたのです．そうなるとキャッチアップが早いのが欧米人です．昨年 2005 年度の Ruby Conference には 200 名を超える Ruby 開発者が集まり，実にその 2/3 以上が「実際に Ruby を仕事に使っている」と答えたのです．

本家であるはずの日本では，趣味として Ruby プログラミングをたしなむ層はたくさんいましたが，なかなかビジネス界に浸透できていませんでしたが，この海外の動きと Ruby on Rails の話題性を背景に，ビジネスとしても Ruby を利用していこうという動きが現れてきました．Ruby が名実ともに他のスクリプト言語と同等以上の地位を築くのはそう遠い将来ではないでしょう．

13 年前，ひとりの技術者のオモチャとして誕生したプログラミング言語がいまや世界中に広がり，人々の役に立っているというのは，にわかには信じられないような驚きです．それもこれもみなオープンソースの力，コミュニティとソフトウェアの自由と相互協力による力だと思います．

成功の秘訣

Ruby は日本発のオープンソースソフトウェアのうち，もっとも成功したものと呼んでも構わないと思います．開発者としてはとても名誉なことだと思いますが，決して成功を狙って実現させたわけではありません．とはいえ，運良く成功を勝ち得た以上，後に続くもののために，「成功の理由」を分析しておくのは良いことだと思います．ふりかえると Ruby の成功の理由には以下のようなものがあると考えられます．

• 素材の良さ

私は若いときからプログラミング言語が大好きで，さまざまな言語を眺めてきました．いろいろな言語の良い点，悪い点について知っています．そのような知識を背景に設計された Ruby は，プログラマが快適にプログラミングできるような仕掛けがあちこちに埋め込まれています．Ruby にはひと目で分かる飛び抜けたすばらしさがあるというわけではありませんが，使い込むにつれ手になじむような感覚があります．多くのプログラマはその点を好んで Ruby を選んでくださっているのだと思います．

オープンソースソフトウェアとして成功するためには，ライバルとなるソフトウェアと同等以上に優れていなければならない，これは鉄則ではないでしょうか．もっとも一番優れたものが勝者となるとは限らないのも厳しい現実です．

• コミュニティ

オープンソースソフトウェア開発の秘訣の 1 つは，コミュニティ運営です．日本においても海外においても Ruby は大変コミュニティに恵まれました．

これには幸運もあったでしょうが，私をはじめとする主要メンバが友好的で建設的なコミュニティ運営を心がけたことも無関係ではないでしょう．私自身 Ruby メーリングリストにおいて，どのような意見に対しても決して乱暴に返答しない，相手を尊重し参考にできる意見は参考にする，いつもユーモアを忘れない，ことを心がけてきました．

そのおかげかどうかは分かりませんが，メーリングリストの雰囲気はおおむね友好的なものですし，意見の対立があっても人格攻撃にまで発展するようなことはほとんどありませんでした．また，カンファレンスなどで出会うコミュニティメンバは初めて出会ったとしても，まるで長年の友人であるかのように接することができました．

• ユーモア

すでに述べましたが，ユーモアは人間関係を円滑にする潤滑油の働きをします．オープンソースコミュニティも例外ではありません．Perl 開発者である Larry Wall はいつもウィットの効いた発言をすることで知られていますが，私も常々彼のようになりたいと思っています．そこでメーリングリストなどいろいろな機会をとらえてジョークを試みています．あいにくスベってしまうこともたびたびあるのですが，みな温かく見守ってくれているようです．

• 海外ユーザ

一説によるとアメリカの IT 関係者人口は日本の 10 倍以上だそうです．つまり，ソフトウェアの潜在ユーザ数も 10 倍以上ということです．教育その他の事情から日本人は決して英語が得意ではありませんが，だからといって日本人・日本語だけを相手にするのはもったいないことだと思います．私も英語が上手な方とは言えませんが，無理してドキュメントを英語に翻訳しました．最初の英語ドキュメントの品質はそれはそれはひどいものでしたが，それでも英語のドキュメントをつけただけで，かなりの反響を得ることができました．海外の人にとって，日本語ドキュメントは暗号と同じですが，英語であればたとえ品質が悪かろうととりあえず意味を想像することはできるからです．

Ruby を介して海外の開発者と交流するようになって 10 年近くなります．メーリングリストのやりとりや各種カンファレンスを通じた交流により，ずいぶん英語も上達しました．今では技術的な内容に限れば，とりあえず意思の疎通はできるようになりました．やはり習うより慣れろです．

「日本人はガイアツに弱い」と言われ続けてずいぶんになります．残念ながらまだそれは事実のようです．しかし，それならばそれを逆手にとって利用してやろうではありませんか．Ruby は海外進出を果たし，海外で広く使われるようになりました．その「圧力」が日本のビジネス界の態度を軟化させるのに役立ちました．「ガイアツ」を利用しない手はないと思います．

• スポンサー・パトロン

オープンソースソフトウェアはソフトウェアそのものは無償で配布されます．ですから，その開発では生活を維持することはできません．理想は高くても霞を食べて生きていくわけにはいかないのです．私も 1997 年に現在の所属に転職するまでは，「表の仕事」を持ちながら，私的に Ruby の開発を続けてきました．しかし，(株) ネットワーク応用通信研究所（以下，NaCl）は，Ruby の開発者として私を雇用し，私が積極的に Ruby の開発を行うことができるように援助してくれました．NaCl はオープンソースを基盤とした零細 SI 企業です．オープンソース界に対する「看板」として，私を「利用」してくれたわけですが，正直なところ 1997 年時点ではそのような価値は皆無に近かったと思います．それでもスポンサーとして Ruby の開発を支援してくれた NaCl には大変感謝しています．

あれから 10 年近くたった現在でも，オープンソース開発者の生活は安定したものとは言えません．しかし，少しずつではありますが，オープンソース企業も増えているようですし，以前よりはオープンソース開発で食べていける技術者が増えているようです．ガイアツその他により，オープンソースの存在感が向上することにより，開発者の境遇も改善されつつあるようです．

本当の秘訣

オープンソースソフトウェアは寿命の長いものが多いです．Ruby も開発開始から 13 年を数えますし，Linux, PostgreSQL などみな 10 年をはるかに超える開発期間を誇っています．これらは，「成功したから長続きした」というよりも，むしろ「長続きしたから成功した」というような傾向がある気がします．オープンソースソフトウェアが成功する最大の鍵は実は「継続的に開発する」ではないかと思います．それだけ長い期間，開発動機を維持すること．オープンソースソフトウェアの成功にあたって，もっとも必要で，もっとも難しいのが，この「動機の維持」ではないでしょうか．私自身を振り返ると，コミュニティに啓発されつつプログラミングする楽しさにはまっている間にあっという間に時間が過ぎ去ったように思います．こんなのも「動機の維持」になるんですね．

Ruby に続く，第二，第三のオープンソースソフトウェアが日本から羽ばたき出すことを心から願っています．

（平成 18 年 7 月 7 日受付）