◆座談会◆

11 Rubyの20年，Rubyのこれから

一般

松本行弘（Rubyアソシエーション） 加藤和彦（筑波大学）
千葉 滋（東京大学） 増原英彦（東京工業大学）
進行・構成：小野寺民也（日本IBM東京基礎研究所） 笹田耕一（Heroku, Inc.） 高橋征義（達人出版会）

自己紹介

小野寺：本日は，「Rubyはなぜ流行ったのか」「次のRubyをどうすればよいか」の2つのテーマについて議論していただければと思います．まずは，自己紹介からお願いします．

松本：Rubyを作りました．今回はよろしくお願いします．

加藤：筑波大の加藤です．言語の専門家ではないのですが，言語オタクであることは結構自信を持っています．松本さんは確か言語オタクというところから入ったとのことなので，共通の趣味の仲間としてお話できるのではないかと期待しています．

千葉：東京大学の千葉です．専門はプログラミング言語の実装とか，デザインだと思うのですけれども，趣味は言語オタクだと言えればよかったのですが，僕は典型的に趣味が仕事になってしまいました．でもコーディングは楽しい．

増原：東工大の増原です．私も分野的には千葉さんと非常に近くてプログラミング言語のデザインから実装，開発環境とかまで手を出しています．一応，Rubyに関する研究も少しやっております．

Rubyはなぜ流行ったのか

◆OSSとコミュニティ

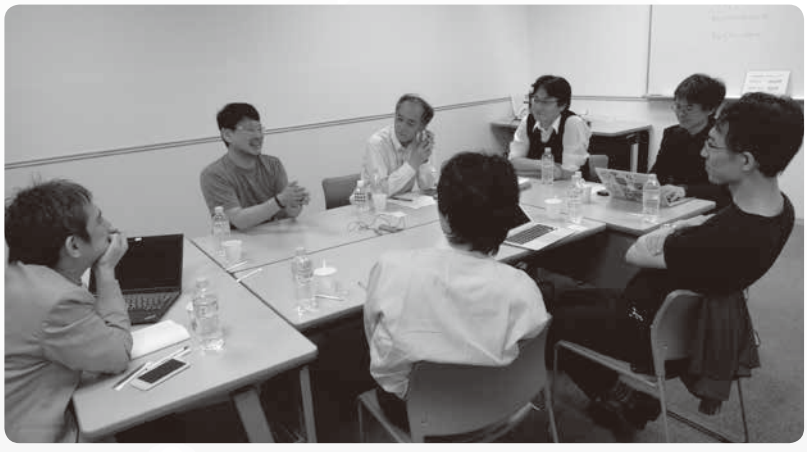
加藤：プログラミング言語を作る人は多く，日本にもたくさんいるのですが，オープンソース的なアプローチをした人というのは意外と少ない．一番多いのは博士論文とか，修士の論文とかで，言語をデザインしましたみたいな人が多い．でも，それは研究の一環としての言語なので，Ph.D.言語で終わってしまう．Rubyは，まずはそこがうまかったのではないかと思います．

小野寺：Rubyは1993年から開発が開始され，1995年に公開とすると，オープンソースというのがまだ普及していないですね．

松本：オープンソースという言葉そのものがないですからね．その頃は，ソースは見ることはできるかもしれないけれども，オープンソース的な，コミュニティで，というアプローチをした言語というのがたくさんはなくて，それがRubyが流行った一番最初の理由だったのではないかなと思います．

千葉：あのタイミングで，Perlはすでにあって，オブジェクト指向みたいなものが欲しいねといったときに，良いデザインをしたというのがRubyの勝因だと僕は思っています．時代があるから，あの前でもいけなかったし，後でもいけなかったと思う．

加藤：Perlはあったのだけれども，Pythonが Rubyよりちょっと先ですよね．あと，1995年



というのは Java の年なのですよね．Windows 95 も 1995 年でした．TCP スタックが Windows 95 に入ったおかげでこの年はインターネット元年と言われています．この特別な年に Ruby が世に出たのは，ラッキーだったと思います．特別な年の出来事として，一緒に語られますから．そして，日本のコミュニティに受け入れられた．

松本：たくさんの方が参加してくださいましたね．それから，1999 年に ASCII から最初の 1 冊が出版されました．英語では，2000 年に Dave Thomas, Andy Hunt による『Programming Ruby』が出版されました．

加藤：世界的にも，あの本がきっかけで Ruby が流行った．もしかしたらあの本が出なかったら世界の Ruby にならなかったわけだ．Ruby on Rails を開発した David Heinemeier Hansson さんもあの本で勉強した．あの本で Ruby を知って，Rails みたいなのを作ろうと思ったときに Ruby を採用してくれた．

松本：あれは，もう二度と Ruby の本は出さないかもしれないから頑張ったというのもあったそうです．Ruby が流行るとは限らなかったので．

加藤：Web 時代にちょうどタイミングがよかったのですね．そして，Rails のために Ruby が出回り始めた．日本で見出されて，世界でも見出されてという，幸運と言っていいか，必然と言っていいか分からないのですけれども，その両方が折り重なって，それで流行って今日に至っているので素晴らしい．

千葉：でも，いろいろ偶然はあったと思うのですけれども，これが好きな人がたくさん世界にいたということは理解できます．別に作者を前にしているからではなくて（笑）．

◆ Ruby のデザイン

加藤：Ruby が素晴らしいのは，Ruby 以降の言語が Ruby の影響を受けているということです．Ruby が画期的なのは，Smalltalk のオブジェクト指向的な，ある意味で本当のオブジェクト指向的なものが現場で役に立つということを人々に気づかせた点だと思います．なので，Ruby のあとの言語はみんな Ruby の影響を受けている．Python は Ruby の前にできてしまったから，Ruby の影響を受けていないのですよね．

松本行弘 ▸（Ruby アソシエーション）

松本：頑なに Ruby の影響がないのですかね（笑），面白いですね．

加藤：プログラミング言語を日本から出すというのは日本の言語研究者の夢でした．1960 年代から．面白いのは，日本の，プログラミング言語の好きな人が，好きなように作った言語が，日本人に受けるのはあり得るとして，それが世界にもちゃんと評価されたことです．

千葉：ある種，新基軸をねらったというのはやはりすごく大事で．93 年当時で，実用アプリを書こうと思うときに，プリミティブタイプをオブジェクトにしてしまったら，きっと速度が出ないから，本当にスクリプティング専用になってしまうという，そういうふうに思ってしまうのではないかと思うのですが，でもいつの間にかコンピュータが速くなって，Ruby で Web サーバまで実装されるようになった．

松本：当時はスクリプト言語ですので，大体 100 行くらい，実行もたぶん 2 分くらいで終わりとかというふうに思っていたものが，たとえば，サーバや Web アプリケーションで何万行です，みたいな話になるというのは，正直あまり予想してなかったです．そうなると，簡易言語ではなく，オブジェクト指向を備えた，ちゃんとした言語でよかったなというのはあります．最初は，スクリプ

◂加藤和彦（筑波大学）

ト言語にオブジェクト指向なんかいるわけない，とか怒られたりもしました．

加藤：時代の流れを捉えていたんですね．

松本：それはあるでしょうね．だから，たとえば，Lispでガベージコレクションが発明されたのはもう60年代でしたが，メジャーな言語でガベージコレクションが採用されたのはJavaが最初みたいな話に．

千葉：Railsの存在というのは大きいと思うのですけれども，Railsが書かれるためには当然そのネットワークまわりのライブラリがないと書けないわけですよね．その辺が入ったのはオープンソースのおかげですか，それとも松本さん自身が．

松本：最初に公開した時点で基本的なTCP/IPの通信機能を持っていました．自分が必要だったので．それで，オープンソースにした直後に，itojun（萩野純一郎）さんが，IPv6を入れてくれました．なので，おそらくRubyは世界で最初のIPv6対応のスクリプト言語でした．

千葉：あとスレッドが最初から入っていました．簡単にそういうのが書けるというところは最初から持っている強みだったのかなと．

松本：Webフレームワークは当然あるのだけれども，ああいう，Railsに相当するようなものは，Hansonみたいな人がRubyにインスピレーションが湧いて，「こうすればActiveRecordみたいのができるんじゃないか」と発想して，要するに，あるルールでデータベースの接続コード，そういうものがダイナミックにコード生成されて，それが非常に道筋よくできるという世界を，Hansonがこういうのがあったらいいなと発想してくれたのをうまくリアライズするきっかけになってくれたのではないかと．そういうものがほかにあったかというと，当時はなかった．

◆ Rubyの楽しさ

笹田：Rubyが流行った理由が「楽しさ」と言われることが多いのですが，その「楽しさ」とは何なのでしょうか．

千葉：たとえば，笹田さんご自身がなぜRubyにしたのですか，Pythonではなくて．

笹田：まわりで流行っているからでした．

千葉：それって，若い人にとっては結構重要なファクタなのではないかと思うのです．当時，もちろん大学の人はLispやSmalltalkなどを知っていたのですが，普通のエンジニアの人がRubyを学ぶと，そのエッセンスを学べる．SmalltalkやLispはやはり時間が経ち過ぎていて，いまさらなんか勉強するのはかっこよくないわけですよ．Rubyは新しいものだから，若い人はやはり新しいものをやりたいじゃないですか．たぶん学ぶ楽しさがあった．ある種，Rubyをやることによって，LispとSmalltalkを同時に学べていた．何言語分かの遺産を全部吸収しているので，やはりそれが楽しかったのではないかと思います．

Rubyがナンバーワンになるには

小野寺：IEEE Spectrumの2014年のプログラム言語ランキングでは，Rubyは8位ですが，ナンバーワンになるには何が必要でしょうか？

◆ Rubyと教育

千葉：座談会だから面白い発言をするという意味でいうと，Rubyがナンバーワンになったら嫌な感じがしますね．授業で教えなければいけない（笑）．今はうちの大学でも教えていますけれども，初心者向けということで，結構機能を封印して使っています．もしメインストリーム言語になって

千葉　滋▸
（東京大学）

しまったら，隅から隅まで，メタプログラミングを含めて教えるわけではないですか．かなり嫌な感じがするのですけれども．

増原：Rubyは文法がフレキシブルなので，1年生の人がちょっと間違えたことを書いてもなんとなく解釈されてプログラムが動いてしまうのですね．TAが呼ばれて，動かないのですけど，どこが動かないのだろう，ああ，ここにピリオドが1個ないからだよとか，なんかそういう世界のところが．

千葉：僕もRubyを東大で教えるというとき，いいことをやるじゃないかと思ったのですけれども，実際，現場に入ってみると，Rubyというのはまだちょっと．Javaとかのほうが今は教えやすいです．なぜかというと，IDEでとりあえず何か打つと，ここが違うと赤線が出ます．なので，とっつきは実はIDEを使うJavaのほうが良くて，Rubyはそういう意味では玄人っぽい．もし今やるのだったら，JavaScriptのほうが，ブラウザにIDEがついているのでとっつきやすい．

松本：一応Rubyにも，そういうツールもないわけではないのですが，まあ，出来がね．さらに言うと，次のメジャーバージョンである，Ruby3.0の重要なテーマとして，静的な解析を強くしたいと思っています．なので，IDEを作れるために必要な道具立てを言語処理系のツールとして用意して，IDEはたとえば，プログラムを書いて，「このメソッドを呼びます」とか，スペルミスをしたら，「あなた，こっちのほうの意味じゃないですか」ぐらい，そういうことができるといいなという話はしています．

千葉：初心者教育は本当にIDEの出来が大きいです．昔，東工大のときの話ですが，Javaを教えていまして，ある年までEmacsだったのを，Eclipseに変えました．そうしたら，いわゆるシンタックスエラーで引っかかる学生が激減して，みんなすいすいいくようになりました．それ以外，授業の内容のカリキュラム，全部同じでしたから，すごく印象的でしたね．

加藤：でも，Rubyを教育に使って，みんなで教えるようになると，プログラミングに対して難しいと思っている，バリアを感じている人にとって，トライできるチャンスが増えるかもしれませんね．

増原：加藤さんや千葉さんはきっとたくさんの言語を自由に使えると思うのですけれども，多くの人は言語の習得というのは時間がかかるので，たくさんの言語を勉強しないのではないかという気がします．最初の言語というのは重要ですよね．そうするともうちょっと大学よりも手前の教育言語にRubyが採用されるのがいいんでしょうか．

松本：たとえば，Scratchっぽい感じのSmalrubyというツールを提供して，小学校にRubyを教えようとしたのですけれども，タイピングできないのですよね，アルファベットがどこにあるか分からない．それで，メニューから選んで積み上げていくというScratchというやり方がいいだろうという話になって，それでそういうツールを作った．面白いのが，Rubyのプログラムに双方向で変換する機能を備えていることですね．積み上げたタイルに対応するRubyプログラムを書き換えると，タイルにも反映される．もちろんある程度パター

◂増原英彦
（東京工業大学）

ンマッチできるものだけなのですけれども．それで小学生でも一応，プログラムみたいなのはできる．今年は松江市の中学校の何校かで試験をしていて，来年度から松江市立中学に行く全部の中学生が勉強する予定です．

◆ 足りない機能

小野寺：Ruby が 1 位を取るためにはこの言語機能が，という視点からだといかがですか．

松本：機能ということはもうないと思うのですよ．ただ，たとえば，千葉先生が指摘されたような，IDE をはじめとした開発環境みたいなところであるとか，エラーを発見するためのものであるとかがあると思います．それから，玄人向けにはどうしても性能が大事．どんなに速くても必ずだれか文句を言う．Web の場合だと，性能とメモリは必ずトレードオフの関係になって，どっちが支配的になるかは，やはり時代によってころころ変わるのですよね．特に Web の領域では，ここ数年はやはりメモリが支配的になることが多かったので，同じことを同じスピードでやっても消費するメモリを減らすと効果がある．また，もうちょっとするとメモリが安く，潤沢になってくるので，今度はメモリを消費してもいいからパフォーマンスを上げようと．あるいはそれを切り替えられるようにしようみたいなことは要求されるようになってくるのではないかと思っています．

加藤：Ruby のライバルというか，Ruby が抜く相手というのは，ダイレクトには Python かなと．

松本：土俵が一番近いのは Python ですね．

加藤：そのランキングで見ても，上にいるのはあと Java，C，C++，PHP とかでしたっけ．それらの言語は分野が違うというか，ドメインが違うから，ちょっと競争相手と違うという理解ですかね．それともその辺にも，Java を押しのけて，となるのですかね．

千葉：というか，ナンバーワンになるということは，結局，業務案件の数では．でもオールラウンダーになっても多分 Java は抜けないと思うのですよ．だから今の状況が続く限りはこの順位はあまり変わらなくて，なんかコンピュータの使い方ががらっと変わる，エポックメイキングみたいなことが起こって，開発者が，わっと流れ込むような，要するに 2000 年頃の Web 業界みたいに，わっと流れ込んでくるときに，そこのマーケットを押さえた言語が次の世代のナンバーワンになると思うのです．

加藤：言語というのはやはりどうしても古びたりするところもあるでしょう．常にリニューアルしているのですけれども，その点，Ruby の展望はどうですかね．

松本：僕は，すべてのものに寿命があると思っています．結局，バックワードコンパチビリティを捨てなければいけないので，それをどこでできるかという．

千葉:たとえば，松本さんの中では，拡張限界である，だからもう次のやつ，ゆっくり，じっくり待つみたいな感じなのですか,それともまだまだ広がる？

松本：できることはいっぱいあるとは思っているのだけれども，本質的なところでは手は入れられないなというふうに思っています．手を入れたほうが良くなるかもしれないのだけれども，手を入れると過去の資産を破壊してしまうので，それはできないなというところがある．そういう意味で言うと,ある種の進歩はできなくなっているとは思う.

千葉：なんか全然違う分野ができて，付け足しみたいな機能がどんどんちょっとずつ拡張できるといいと思うのですけれどもね．新しい開発者を迎え入れて，リフレッシュするためには，新しい血を入れなければいけないわけではないですか．そのためにも進歩を止めてしまうと，やはり人が集まらないところがあるので．

◆ 企業とアカデミア

加藤：最近ちょっと話題になっている言語というのは Go にしても，Swift にしても，割と企業からきているのですよね．それに対して，個人の集まりみたいな Ruby はトレンドに逆行しているか

もしれない，どう生き残っていくかというのは課題ではありますね．幸い，大変に良いコミュニティがあるので，頑張って生きていってほしいなと．しかし，なぜ日本のIT大手企業さんは，言語作りをやらずに，草の根的に発生した言語が世界で評価されて頑張っているのかなと．

千葉：それは，話が終わらなくなってしまうので恐縮ですけれども，やはり産業構造に起因していて（笑）．やはり請負業務が多いから，自分たちで生産性を上げても得しないという．Goをなぜ作ったかというと，あれを作ってメンテすることで自分たちの業務が簡単になるから，業務に直結するじゃないですか．自社開発をしている会社が少ないと，自社のサービスを自社で作っているというところがもっと増えてこないと．

加藤：でも日本の大手IT企業さんはやっているんじゃないの，経済規模をもってやっているでしょ．

松本：だけれども，生産性を上げることのインセンティブが少ないのですよ．グーグルというのは自分のところがサービスを提供しているわけなので，要するに，お客さまのためにソフトを作っている会社ではないので，そうすると，その自分のところの生産性が上がれば，自分のところの儲けが上がるという構造がある．でもたとえば何人／月という仕事をよくしてしまったりすると，早く終わったら，半分で終わったら，支払いは半分でいいなと言われてしまうので（笑），フルフルで働いてもらったほうがいい．

加藤：産業構造なり，評価システムを変えないといけないのかもしれないのだけれども，それを言うと必ず雇用慣行の話が出てきて，これまた話が長くなってしまいますが（笑）．ソフトウェア重要とか，言語重要とか思う経営者が少ないからということでしょ，きっと．

松本：あとは，オープンソースって，やはり学生は道楽の延長なのですよね．アカデミア的には研究にもならないことをやっているという扱いを受けることが多いのだけれども，うまくできるようになるといいなと思います．情報処理学会的にはそんな感じのまとめを．

加藤：道楽だから楽しくやっていられる．松本さんのスタイルも，Rubyを生み出した頃からずっと飄々としてやっていらっしゃる．今日見てもなんか飄々とした感じがあって（笑）．その飄々感をキープしていると，飄々としているRuby開発者もついてきて．

松本：ありがたいですね，本当に．

加藤：これがいい感じですよね．

松本：ですね．

小野寺：皆さん，今日はありがとうございました．

（2015年6月23日　化学会館にて）

（左から）笹田耕一，高橋征義，加藤和彦，松本行弘，小野寺民也，千葉滋，増原英彦

松本行弘　matz@ruby.or.jp

1990年筑波大学第三学群情報学類卒業．（財）Rubyアソシエーション理事長，米Heroku Chief Architect, Rubyなど兼務．プログラミング言語の設計と実装に興味を持つ．

加藤和彦（正会員）kato@cs.tsukuba.ac.jp

東京大学理学部情報科学科助手，筑波大学電子・情報工学系講師，助教授を経て2004年より筑波大学大学院システム情報工学研究科コンピュータサイエンス専攻教授．現在，同専攻長，日本ソフトウェア科学会理事長．

千葉　滋（正会員）chiba@chibas.net

1991年東京大学理学部情報科学科卒業，1996年同大理学系研究科情報科学専攻博士課程退学．博士（理学）．東京大学助手，筑波大学講師，東京工業大学教授などを経て，2012年より東京大学情報理工学系研究科教授．

増原英彦（正会員）masuhara@acm.org

東京工業大学情報理工学研究科教授．博士（理学）．1995年東京大学理学系研究科情報科学専攻中退後，同大学院総合文化研究科助手，講師，准教授を経て2013年より現職．