



# 第39回 関数プログラミングの集い 2011 in Tokyo

2011年9月17日  
(株)インターネットイニシアティブ 大会議室  
(東京都千代田区)

山本和彦 (株) IJ イノベーションインスティテュート

## 開催経緯

関数型言語 OCaml に特化したイベント「OCaml Meeting」が2009年と2010年に開催され好評を博した。その主催者である小笠原さんと古瀬さんから「せっかく ICFP (International Conference on Functional Programming) が日本にやってくるのだから、今年は関数型言語の集まりを東京で開催したい」というお話をいただいた。

そこで、関数プログラミングの普及を願い、「関数プログラミングの集い」を開催することにした。会場は(株)インターネットイニシアティブの大会議室をお借りし、日時はICFPと併設される各種イベントと重ならないように配慮してICFPの直前とした。

事前に各関数型言語のキーパーソンと打合せと称して一緒に飲み、プログラムの概略を決めた。午前中にチュートリアルを設け、午後は関数型言語をビジネスで活用している事例、各言語の最近の動向、そして Lightning Talk で構成することにした。最終的なプログラムは、表-1の通り。集いは、学会ほどフォーマルではなく気軽に参加できるが、内容はしっかりとしたものにしたつもりである。飲んだ勢いで、タイトルの関数だけ古い漢字を使うことになった。

参加を partake.in で募ったところ、180名の枠に対し250名の登録があり、キャンセルが72名出て、最終的には178名の方が参加された。Twitterのタグを #fpm2011 に指定したところ、当日は数千

10:00 ~ 10:10	挨拶と諸注意
10:10 ~ 11:00	チュートリアル:「関数プログラミングの道しるべ」山本和彦
11:00 ~ 12:00	チュートリアル:「モナドについて」田中英行
13:30 ~ 14:00	「ITプランニングにおける関数プログラミング」今井直洋
14:00 ~ 14:30	「COBOL meets Haskell」岡田譲二
14:50 ~ 15:35	言語アップデート 1 Scala 水島宏太 Clojure 登尾徳誠 Erlang Uenishi Kota
15:35 ~ 16:35	言語アップデート 2 Haskell 青江光敏 F# いげ太 SML 上野雄大 OCaml 小笠原啓
16:55 ~ 17:40	Lightning Talk 「プログラミング Coq」池淵未来 「マルチパラダイムで行こう in F#」ながと 「Paraiso 計画 -- Haskell によるスパコン向けコード生成」村主崇行

表-1 プログラム

のツイートが観測されるほど盛り上がった(図-1)。Togetterにまとまっているので、興味がある方はご覧いただきたい(速読版もある)。ツイートを拝見すると、関数プログラミングの初心者から実際に業務で使っている人まで、幅広い方々が集まったようだ。また、どの関数型言語を使っているかのアンケートも実施され、図-2のような結果となった。

関数型言語の強みは、高品質のソフトウェアを短期間に作成できることである。発表には業務での利用例がたくさんあり、実体験に基づいた話は大変興味深かった。以下、主に発表内容について報告する。



図-1 Twitterの様子

なお、発表資料は文献1)にまとまっている。

## 関数プログラミングの道しるべ

筆者が関数プログラミングについて説明した。関数プログラミングのパラダイムは、命令プログラミングのそれとはまったく異なる。そのため命令プログラマが関数プログラミングを学ぶと、それまでの知識が足枷となって本質を理解できず、入り口で引き返してしまうことが多い。コンビネータライブラリのような本殿を見ないのは、実にもったいないことなので、本殿まで駆け足でお連れしようとして試みた。

関数プログラミングという言葉には共通理解がないことを説明し、筆者の定義は「永続データプログラミング」であると述べた。永続データとは、破壊できないデータのことであり、永続データプログラミングとは再代入を使わないプログラミングである。

再代入を使わないプログラミングの初歩としてリスト操作を説明し、与えられた文字列の中から最長重複文字列を探す問題の解き方を示した。また、永続データとして使える木構造に触れた。関数型言語では、次のように木を簡潔に、しかも危険な null を使わずに定義できる。

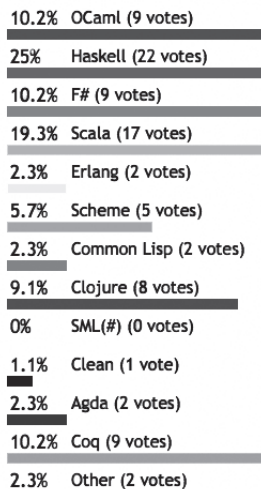


図-2 利用している関数型言語のアンケート

```
-- Haskell で二分木を記述した例
data Tree a = Leaf | Node (Tree a) a (Tree a)
```

この構造を発展させて探索木を作成すれば、集合やハッシュテーブル（辞書）として利用できる。また、木構造はヒープやキューに応用することも可能だ。最後にコンビネータライブラリが、パーサ定義やデリバティブ記述に利用されていることを述べた。たとえば、パーサコンビネータライブラリを使えば、パーサをBNF (Backus-Naur Form) 通りに記述でき、しかも静的型付き言語であれば型検査のおかげで、たくさんエラーをコンパイル時に発見できる。

## モナドについて

2番目のチュートリアルとして、田中さんによるモナドの説明が続いた。モナドという言葉仕様に組み込まれている言語 Haskell が有名だが、ほかのプログラミング言語でも利用できる。

まず、モナドがいろいろなメタファで語られていることが紹介され、Haskell の最大の難関と言われるだけあって、多くの人が抽象的で分かりにくいと考えている様子うかがえた。そして、モナドとは

結局計算コンテキストの抽象化であり、構造化定理に必要な要素を自然かつ簡潔に記述できると説明された。

次にモナドの具体例として、リスト、直和型、入出力などが紹介された。また、モナド変換子を用いれば、複数のコンテキストを合成できると説明された。最後に具体的なモナドの作り方が示されたが、ここはかなり Haskell に依存した話となっていた。Haskeller には有益だったが、Haskeller 以外には分かりにくかったかもしれない。

## IT プランニングにおける関数プログラミング

昼休みを挟んだ午後の始めの発表 2 つは、集いの基調講演に相当する。一番バッターは、(有) IT プランニングの今井さんであった。同社は、関数プログラミングを中心に据えた企業であり、多数の実績を誇る。技術力が高く、関数型言語も OCaml, F#, Scala などを必要に応じて使い分ける。

まず、FX (外貨証拠金取引) のチャートを表示する案件が紹介された。24 時間 365 日稼働しなければならないシステムで安定性が求められていた。OCaml を用いたおかげで、コンパイル時にたくさん間違いを見つけることができ、拡張に強く、見通しがよく、パフォーマンスもよいシステムが構築できた。

次に名古屋大学のネットワーク管理システムの案件が説明された。さまざまな管理ツールを統合するシステムであり、JVM を要求されていたことから、Scala で実装された。強力な型システムのおかげで、複数人による開発が容易であり、しかもパーサなどのコードの見通しがよくなった。

次に紹介されたのはスマートデスクトップの案件であった。サーバ側を GAE と Scala, クライアント側を haXe, OCaml, および Coq で実装した。特筆すべきは、コードの一部を Coq で証明していることである。

最後に、まだ普及したとは言えない関数型言語を使うには、お客さんの信頼を得ることが一番大切だ

と述べられた。信頼を得られれば、言語を選択でき、よい製品ができるので、さらに信頼を得ることができると経験が語られた。

## COBOL meets Haskell

2 番バッターとして、(株) NTT データの岡田さんが、Haskell による COBOL のソース解析事例を紹介した。昨今の不況で、社内システムに対する大規模な投資はほとんどなく、既存のシステムを拡張する案件が多い。まだ現場で動き続けている COBOL を相手にすることになるが、仕様はほとんど残っておらず、ソースを解析するのが一番だとのこと。

パーサは関数型言語の最も得意とする分野の 1 つであり、言語には Haskell を選んだ。Haskell で作られたオープンなライブラリを利用することにより、大規模な解析システムを実装している。このシステムでは、COBOL のソースを解析し、最終的には案件のシステムの仕様を作成できる。Haskell を使いこなせる人の確保に苦労しており、最近はインドの方も採用されているそうだ。

## 言語アップデート

続いて、言語アップデートのセッションを 2 つ設けた。主催者の意図としては、各関数型言語のホットピックをなるべく若手の方に発表していただくことであった。

## Scala

Java VM でオブジェクトと指向と関数型を融合する言語 Scala の近況については、水島さんが説明された。なんといっても、Twitter, LinkedIn, Amazon.com, VMWare などが採用し、急速に普及している感がある。Scala 2.9 からは、永続の並列コレクションを用いると、par メソッドを挟むだけで、自動的にマルチコア化される。次に、並列コ

レクションから和を計算するコードの例を示す。

```
def psum[T:Numeric] (t: Traversable[T]) :T =
  t.par.sum
```

これは、関数型言語の中でも一番進んでいる並列化の機能と言えるだろう。また、プロセスの入出力をパイプのように繋げる新しいプロセス処理ライブラリの紹介もあった。Scala の商用サポートをする Typesafe 社が設立されたことも報告された。Scala IDE for Eclipse の開発を移管され、これから優れた開発を提供していくことになる。また、Scala コミュニティでは、フレームワークではなく、RESTful HTTP サービスのライブラリが流行っているそうである。

## Clojure

Java VM 上で動く革新的な Lisp である Clojure については、登尾さんが話された。まず、近日中にリリースされる Clojure 1.3 の機能が紹介された。1.2 と比べて、型アノテーションを付けることでパフォーマンスが改善されること、defrecode が機能強化されたことなどが挙げられた。プロジェクト管理ツール Leiningen も、最後に失敗したテストの実行やリポジトリからの検索などの機能が追加され、使いやすくなった。

また、JavaScript に変換される ClojureScript が紹介された。ブラウザのみならず、スマートフォンやサーバサイド (node.js) でも活用できるとのこと。さらに Heroku クラウドでも Clojure を利用できるようになったそうだ。

## Erlang

Uenishi さんが Erlang の概要と近況を話された。まず Erlang は言語ではなく、OTP (Open Telecom Platform) とともにサーバプログラミングフレームワークとして捉えるべきであることが強調された。Erlang 特有の軽量プロセスやホットスワップにつ

いても触れられた。Erlang というと動的型付き言語であるという印象があるが、最近では型アノテーションが付けられるそうだ。Tidier というリファクタリングツールも紹介された。

## Haskell

純粋関数型言語 Haskell の近況を語ったのは青江さんである。以前人気のあったインタープリタの Hugs はすでに保守されてない。一方で、GHC (Glasgow Haskell Compiler) が最新の機能を取り込みながら、年 1 回のペースでバージョンアップされている。ライブラリの開発が盛んで、929 名のプログラマが合計 3,368 個のパッケージを公開している。中でも、Web フレームワークの開発が活発であると説明された。

## F#

いげ太さんが、OCaml から派生した .Net フレームワークの関数型言語 F# の近況について発表された。F# 3.0 が Visual Studio 11 の一部として、レビューリリースされた。F# 3.0 はデータとサービスを統合することを目指しており、それを実現する機能として、型プロバイダとクエリ式が提供される。これらを使えば、外部データに対して自動的な型付けが容易になり、サービスとの連携が強化される。

## SML

上野さんが自ら開発されている SML の一実装 SML# について説明された。SML# は、関数型の理想の世界を追求するのではなく、既存の基盤技術である C や SQL との親和性を高めることに注力している。SML# の関数を C に渡したり、C の関数をファーストクラスとして扱うなど、C とシームレスに連携している。次は、C の関数 qsort に、SML# の配列と SML# で定義した比較関数 compareReal を渡し、ソートするコードの例である。



```

fun qsort (a,f) = _ffiapply "qsort"
  (a: 'a array,
   Array.length a : int,
   _sizeof ('a) ,
   f : ('a ptr, 'a ptr) -> int) : unit
fun compareReal (p1,p2) =
  case Real.compare (!!p1, !!p2) of ...
qsort (realArray, compareReal) ;

```

なお、ICFP では SML# が基調講演に抜擢されたこともあり、SML# の存在が広く認知される結果となった。

## OCaml

主催者の1人である小笠原さん（写真1）が、OCaml について発表された。新しい機能としては、第一級のモジュール、シグニチャの破壊的代入、GADTs (Generalized Algebraic DataTypes) がある。これらにより、飛躍的に強力な抽象化と型検査が可能になった。名前が正式に Objective Caml から OCaml に変更となった。また、OCaml を商用サポートする OCamlPro という会社が設立され、Windows 7 用のバイナリ配布や、コンパイラの改善に取り組んでいる。

## Lightning Talk

稲妻のように短時間で話す Lightning Talk では、3つの発表があった。まず、筆者のもとで「プログラミング Coq」というチュートリアルを作成し公開している池淵さんが「Coq で証明するのは楽しいよ」という話をされた。次に、ながとさんが F# を使うと、スクリプティングやオブジェクト指向プログラミング、そして副作用の利用が簡単にできることを GUI を例にとり説明された。最後に村主さんが、Haskell を使って並列化マシン用のコードを生成する話をされた。コードの見通しがよく、アノテーションを付けるだけで最適化の戦略を変えられるそうだ。



写真1 OCamlについて説明する小笠原さん

## 懇親会

集いに引き続き、同じ会場でケイタリングサービスを使い懇親会を開催したところ、約60名の方が参加された。同じテーマに興味がある方々の懇親会の例にもれず盛り上がった。発表者のほとんどが参加されていたので、質問攻めにあつたのでないだろうか。なお、(株)インプレスジャパンより発売前の「Scala スケーラブルプログラミング第2版」を2冊景品としていただいたので、ジャンケンに勝った参加者2名に贈呈された。

## 次回

今年も何らかの形で、続編のイベントを企画することになると思う。興味のある方は、ぜひ参加していただきたい。

### 参考文献

- 1) 資料一覧, <https://sites.google.com/site/fpm2011papars/>  
(2011年12月27日受付)

山本和彦 | kazu@ij.ad.jp

IJ イノベーションインスティテュート主幹研究員。最近では、もっぱら Haskell でソフトウェアを開発している。翻訳書「プログラミング Haskell」。主なフリーソフトウェア作品に Mew, Firemacs, Mighty がある。