

純粋関数型言語 Haskell を用いた システム開発方法論に関する一考察

System Development Methodology using Pure Functional Language Haskell

青木 唯一†
Yuiitsu Aoki

須栗 裕樹‡
Hiroki Suguri

1. はじめに

本研究では、純粋関数型言語を採用したプロジェクトに適したシステム開発の方法論を設計、実践、及び評価する。具体的には言語として Haskell を用いる。

これにより、実際のシステム開発プロジェクトにおける純粋関数型言語の普及を促進し、もって高品質の情報システムを容易に構築できるようにすることが本研究の目的である。

2. 背景

現在、より品質の高い情報システムの開発をより迅速に行うことは喫緊の課題である。そこで、高品質のシステム開発を行う手段の一つとして純粋関数型言語を用いる事が考えられる。本研究では Haskell を用いる。Haskell は次に示すように様々な優れた特性を持つ。

- ・副作用を持たない関数型
- ・遅延評価
- ・強力な型システム
- ・柔軟なパターンマッチング

したがって Haskell は、一般的な手続き型言語や不純な関数型言語に比べ、「開発を迅速に行える」「バグを抑えやすい」「拡張性が高い」といった利点があると考えられる。

しかし、純粋関数型言語を用いて成功した現実のシステム開発プロジェクトは少ない。その理由として、我々は純粋関数型言語には次の3つの欠点があると考ええる。

1. ウォーターフォール、アジャイルなどの、従来のプロジェクトマネジメントフレームワークとの相性が悪い。
2. 統合開発環境、大規模ライブラリー、アプリケーションフレームワークなどの周辺技術が、高い品質で提供されていない。
3. プログラムの構成、デバッグ、ドキュメンテーション等のコーディング手法が、一般的な手続き型言語とは異なるため、従来の経験の蓄積が役に立たない。

† 宮城大学大学院事業構想学研究科

‡ 宮城大学事業構想学部

従って、優れた性質を多く持つ純粋関数型言語を、実際のシステム開発において上手く活用するための方法論を研究することが必要である。

3. 研究方法

3.1 概要

本研究では、次の四つの手順をふむ。

- (1) 純粋関数型言語と従来型方法論との不整合部分を特定する。
- (2) 特定した不整合部分の改良方法を提案する。
- (3) 提案した改良方法を実装する。
- (4) 実装した改良方法を評価する。

3.2 予備研究

本研究を進める為の予備研究として、PHP によるシステム開発と Haskell によるシステム開発とを比較した。具体的には簡単な Web アプリケーションの機能設計（実装する言語とは独立したユーザーインターフェースの外部仕様）を作成し、それを二つの言語で実装した。

これによって比較したのは以下の3点である。

- (1) 処理速度
- (2) コード量
- (3) 開発の難易度

3.2.1 予備研究の動作環境

動作環境は図1の通りである。

OS	Windows7 上に VirtualBox4.3.6 を用いて構築した、 Ubuntu12.04 の仮想環境
ウェブサーバー	Apache 2.2.22
PHP	PHP 5.3.10
Haskell	GHC 7.4.1
ブラウザ	Firefox 26.0

図1 動作環境

Haskell で Web アプリケーションを作成する際には、GHC (Glasgow Haskell Compiler) [1] がコンパイルした実行ファイルを CGI として用いた。

PHP インタープリターは Apache のモジュールとして動作する。動作結果を図 2 に示す。左のウィンドウが Haskell によるプログラムである。右のウィンドウが PHP によるプログラムである。

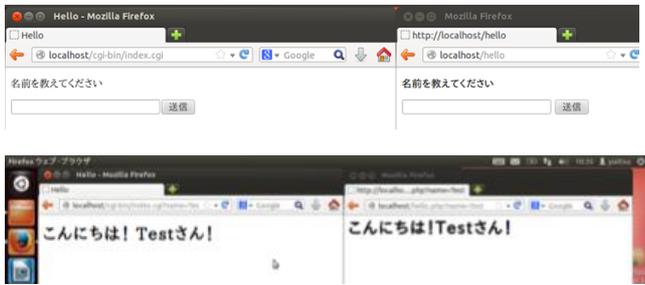


図 2 動作結果画面

3.2.2 予備研究の成果

(1) 処理速度の比較

動作させた Haskell と PHP のコードの速度を端末上で比較した。time コマンドで時間を測り、20 回の平均をとった。その結果、Haskell を用いたプログラムの平均処理時間の方が短い事が確認できた。

PHP : 処理時間 平均 0.042sec

Haskell : 処理時間 平均 0.020sec

(2) コード量の比較

Haskell に比べて、PHP は短いコードでの記述が可能だった。Haskell では計算処理部分は短く記述が可能であるが、HTML 出力部分の記述内容が多かった。

PHP : 文字数 145

Haskell : 文字数 520

(3) 開発の難易度の比較

このプログラムを作成するのにかかった時間によって、開発の難易度を比較した。

PHP : 1 時間

Haskell : 48 時間

3.2.3 予備研究の考察

計測時間の比較は Haskell の処理速度が PHP の処理速度を上回っている結果が出た。しかし、端末上で Apache のモジュールとして動作する PHP インタープリターと CGI として書き出した Haskell のバイナリープログラムを比較しても、正しい処理速度の比較はできない。よって、この結果から正しい考察は行えない。

コード量の比較については、Haskell のほうがより多くの記述が必要であるという結果が出た。これは、副作用を伴う処理である Web アプリケーションの入出力処理を行うための XHTML パッケージと CGI パッケージ [2][3] を用いたためである。したがって、これも適切な評価では

ない。

開発の難易度の比較については、Haskell による開発に必要なライブラリー等の情報が少なかったため、作業が難航することが多かった。そのために時間がかかってしまった。

また、Haskell での開発を行う際、書き換えが行われる変数の使用を前提とした構造設計を Haskell で実装することができなかった。純粋関数型言語によるプログラム開発においては、関数一つ一つについての処理の流れを指定する構造仕様の記述が必要であるとわかった。さらに、実装する言語に応じた機能仕様を作成することも必要であると考えられる。従来の方法論では、実装とは切り離して機能仕様を考えるのが一般的である。しかし、その場合、純粋関数型言語を用いる場合にはきわめて開発効率の悪い機能仕様ができてしまう。

4. 今後の展望

今回、単純な Web アプリケーションを実装した際に、Haskell の利点をほとんど実感する事ができなかった。副作用を伴う処理である Web アプリケーション開発に対して Haskell をどのような形で用いる事が最適であるかが今後の課題となる。

予備研究においては、処理速度、コード量、開発の難易度のすべてについて、適切な検証ができなかった。まずこれらを改めて適切な方法で行うことが第一の課題である。

システム開発の全体を効率化するには、Haskell による実装を、システムにおいて GUI の入出力を伴わない特定のモジュールのみに適応させる方法が考えられる。これにより、Haskell の利点を最大限に発揮できる。従来のシステム開発方法論のノウハウを生かしながら純粋関数型言語によるシステム品質の向上が見込める。

3.1 (2) で述べた、不整合部分の改良方法については、次の二つの方針が考えられる。

- (1) 言語の改良
- (2) 方法論の改良

(1) については、従来型の機能設計や構造設計に適した言語となるように、Haskell の文法を改造し、コンパイラを実装する。(2) については、Haskell の言語仕様を前提とした、機能設計や構造設計の設計方法論を検討する。

5. 結論

純粋関数型言語を採用したプロジェクトに適したシステム開発の方法論を設計、実践、及び評価する研究について述べた。予備研究の結果を紹介し、考察を述べた。

今後は予備研究の成果をもとに、3.1 及び 4. で述べたように研究を進める予定である。

References

- [1] The Glasgow Haskell Compiler. <http://www.haskell.org/ghc/>
- [2] The xhtml package. <http://hackage.haskell.org/package/xhtml1>
- [3] The cgi package. <http://hackage.haskell.org/package/cgi-3001.1.5.2>