

ノート

関数型プログラミング言語を用いた対話的なソフトウェア開発
— 簡易な全身運動解析への適用事例 —後濱 龍太*¹⁾Interactive software development using Functional Programming Language
- example of simple whole body motion analysis -Ryuta Atohama *¹⁾

キーワード：関数型プログラミング言語, Clojure, 対話的开发, 全身運動解析, モーションキャプチャ

Keywords：Functional Programming Language, Clojure, Interactive programming, Whole body motion analysis, Motion capture

1. はじめに

近年、運動の習慣化による生活習慣病予防の重要性が認識されている⁽¹⁾。筆者は、自己決定理論を援用した「運動の習慣化支援システム」の開発を目的として研究を推進している⁽²⁾。このシステムのコンセプトは、1人で使用でき、参加者の運動状況をリアルタイムに解析しながら、運動の適切さに応じて視覚効果をリアルタイムに生成する事により、参加者を適切な運動へいざなうものである。

本研究ではこれを、次の2つの動作を行うシステムとして具体化した。①まず簡易モーションキャプチャと無線型心拍センサの2つのハードウェアから情報を取り込み、参加者の運動状況を解析する。②参加者の運動状況に応じて変化する視覚情報をフィードバックする事で、運動状況への理解促進と適切な運動への誘引を行う。

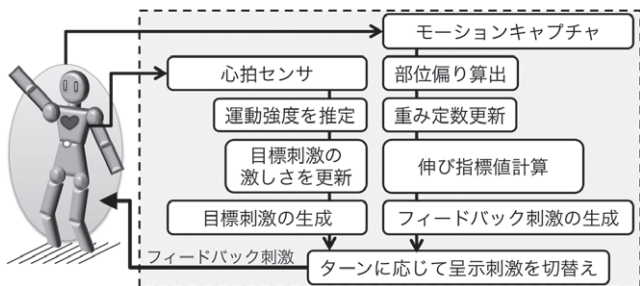


図1. 開発中の運動習慣化支援システム

システムに必要な機能、つまりセンサ入力の処理、ディスプレイ表示内容の生成といった処理の実装に関数型言語Clojureを採用した。本稿では、その概要について報告する。

2. 関数型言語Clojureによる開発

2.1 開発環境 本稿のプログラムは以下の環境で動作

を確認した。マシンはMacBook Air (OS : Mac OS X 10.7.5, プロセッサ : Intel Core i7 1.8 GHz, メモリ : 4 GB 1333 MHz DDR), エディタとしてEmacs 23.3にSLIMEプラグインを用い、言語はClojure 1.4.0である。また簡易モーションキャプチャはMicrosoft社製モーションセンサKinectを使った。

2.2 Clojureの概要 Clojureとは、Java Virtual Machine (JVM) 上で動作するLisp⁽³⁾⁽⁴⁾、Clojureを採用する主なメリットは次の5つである。(i)JVMで動作し、Windows, Mac, Androidなど多くのOSで使用可能。(ii)Javaへ透過的にアクセスでき、過去に蓄えたJavaライブラリ等の資産を活用可能。(iii)Read Eval Print Loop (REPL) によって、開発中のソフトウェアを実行しながら機能を付加可能。(iv)文字列、リスト、ベクタ、セットといった、論理的にリストとみなせるデータの抽象「シーケンス」と、シーケンス処理のための関数が用意されており、文字列やリストの処理を統一的に記述可能。(v)Lispの備えるマクロや高階関数など高い抽象化機能を用いてプログラムを簡潔に記述可能⁽⁵⁾。

2.3 高階関数による簡潔な記述 運動解析では、腕や足が剛体リンクであると単純化して、その長さや向き、角度を計算した。今回、Clojureの高階関数(引数や戻り値に関数をとる関数)を使う事で、これらの計算を簡潔に記述できた。例としてまずベクトル $a=[a_x, a_y, a_z]$ と $b=[b_x, b_y, b_z]$ の内積をあげる。図2に示したJava実装例では自由変数tmp, iが登場する(2, 3行目)のに対し、図3に示したClojure実装例では高階関数applyやmapを使う事で、同じ内容を、自由変数を使わずに、より短く簡潔に記述した。Clojureでは数式と同様、内側の括弧から評価される。まず、2行目のmapでは第2, 3引数のa, bから要素を1つずつ、第1引数の無名関数へと渡し、各要素を掛け算した結果を集めて要素数3のシーケンスを得る。つぎにapplyでは第1引数の関数(+)をこのシーケンスへ適用し、3要素の和、即ち内積を返す。

```

1 double InnProJava (double a, double b) {
2   double tmp;
3   for (int i=0; i< a.length; i++) {
4     tmp += a[i] * b[i];
5   }
6   return tmp;
7 }

```

図2. 内積を返す関数のJavaでの実装例

```

1 (defn InnProClj [a b]
2   (apply + (map #(* %1 %2) a b)))

```

図3. 内積を返す関数のClojureでの実装例

モーションセンサで計測した姿勢データは、参加者の肩や肘など全身で14の特徴点についてそれぞれ3次元座標を格納している。手首をa, 肩をb, 肘をOrigとした時(図4), 例えば前腕の長さは式1により計算できる。

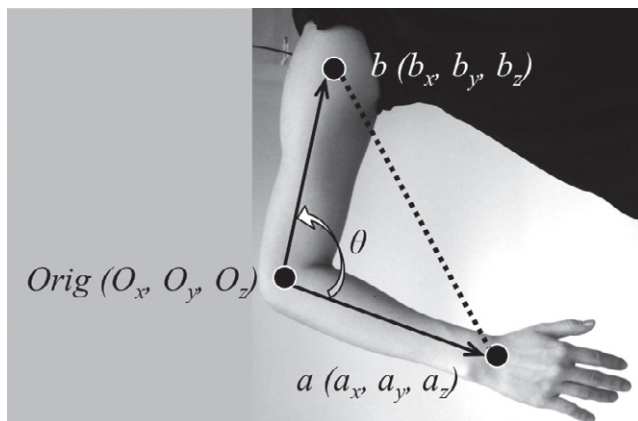


図4. 肩, 肘, 手首を例とした解析のイメージ

$$Abs(Orig, a) = \sqrt{(O_x - a_x)^2 + (O_y - a_y)^2 + (O_z - a_z)^2} \dots\dots (1)$$

高階関数を使う事で式1は図5の様に実装できた。ただしMath/sqrtは平方根である。

```

1 (defn calcAbsClj [o a]
2   (Math/sqrt
3     (apply + (map #(Math/pow (- %1 %2) 2) o a))))

```

図5. リンクの絶対値を返す関数のClojureでの実装例

2. 4 REPLによる対話的開発の効果 REPLはプロンプト形式のインタフェースを備えており, これを活用すると, ①引数を与えて戻り値を検証するテストを関数単位で対話的に行ったり, ②ソフトウェアを実行中に関数の変更・追加といった編集が可能となる。①, ②の特長により, プログラムに小さな変更を加えるたび, すぐにその妥当性を確かめながら, プログラムの機能を拡張できる。

例えば, 図4中の肘の角度 θ を余弦定理により計算する関数calcAnglClj(図6)において, 戻り値の単位をラジアンから度へ変更する事を考える(図7)。ここでは2つの変更を行った。(1)ラジアンを引数に取って度を返す関数rad2degを新たに定義し, (2)calcAnglCljの戻り値にrad2degを適用する。この間, プログラムに変更を加えてはREPLで検証する工程を繰り返し, 少しずつ確実に機能を拡張する事で, 結果として迅速にソフトウェアを開発できた。

```

1 (defn calcAnglClj [a orig b]
2   (let [v-oa (map #(- %1 %2) a orig)
3         v-ob (map #(- %1 %2) b orig)
4         abs-oa (calcAbsClj orig a)
5         abs-ob (calcAbsClj orig b)
6         oaob (innProClj v-oa v-ob)]
7     (Math/acos (/ oaob abs-oa abs-ob))))

```

図6. 変更前(θ をラジアン単位で返す関数の実装)

```

1 (defn rad2deg [theta]
2   (* 180 (/ theta Math/PI)))
3
4 (defn calcAnglClj [a orig b]
5   (let [v-oa (map #(- %1 %2) a orig)
6         v-ob (map #(- %1 %2) b orig)
7         abs-oa (calcAbsClj orig a)
8         abs-ob (calcAbsClj orig b)
9         oaob (innProClj v-oa v-ob)]
10      (rad2deg (Math/acos (/ oaob abs-oa abs-ob))))

```

図7. θ を度単位で返す様に変更後(ラジアンから度への変換関数を追加し(1行目), calcAnglCljの戻り値に適用した(10行目))

3. まとめ

本稿では, 関数型プログラミング言語Clojureを用いた簡易なモーションキャプチャのデータ解析ソフトウェアの開発について報告した。高階関数を使う事で計算手続きを簡潔に表現できた。また, 対話的なプログラミング環境REPLの活用により, 迅速なソフトウェア開発ができた。

(平成25年7月23日受付, 平成25年8月21日再受付)

文 献

- (1)下方 薫: “これからの生命健康科学を考える: 今なぞ予防が大切か”, 中部大学生命健康科学研究所紀要, vol.2, pp.29-34 (2006)
- (2)後濱龍太: “運動習慣化のための創発的バイオフィードバックシステムの開発”, 地方独立行政法人 東京都立産業技術研究センター 平成25年度 研究成果発表会要旨集, p.61 (2013)
- (3)Clojure公式ページ, <http://clojure.org/>
- (4)Stuart Halloway: “プログラミングClojure”, オーム社(2010)
- (5)Paul Graham: “ハッカーと画家”, オーム社(2005)