

# 論文紹介: The camel has two humps (ドリトル・アンプラグド会合)

久野 靖\*

2008.5.5

## 1 はじめに

- 「ビューティフルコード」の翻訳本が発売されたので反響が気になって色々見ていたら見つけた論文

<http://www.cs.mdx.ac.uk/research/PhDArea/saeed/paper1.pdf>

- 「プログラミングができるようになるか否かを一発で調べるテスト」そんなのありか?
- あちこちのブログで議論沸騰しているらしい
- まあ、読んでみてから。

## 2 Introduction

- コンピュータサイエンス専攻の入学者の 30~60%はプログラミングができるようにならずに落第し続けている
  - 1950年代にコンピュータが発明されてからずっと!
  - 何十年にも渡って、コンピュータサイエンスは進歩しているのに!
  - 教授方法についても多くの先人がいろいろと工夫してきているのに!
  - これは英国だけでなく世界中でそうなのだ!
- 学習者の能力や意欲の問題ではない…意欲は十分あるし大学に入学するだけの能力もちやんとあるのに!
- 教授方法についても多くの進歩がある --- グラフィクス、AI、論理プログラミング、OO、C、C++、関数型。が、無益!
- せめて、学習開始前に適性を判別することはできないか? → 未だかつて有効なテストは見つかっていない

## 3 Related work

- 非常に多くの先人の業績がまとめられている → 一見の価値あり
- とくに: Johnson-Laird の提唱した個人の推論構築モデル

- Step 1. ある選言が真であるような状況を想像する (メンタルモデルの構築)
- Step 2. そのモデルで言えるような結論を導き出す
- Step 3. その選言が真であるような別のモデルで、上記の結論が偽であるようなものがないかチェック → 見つからなければその結論は選言に基づく正しい推論結果と判断

- これを (よさげだったので) コンピュータサイエンス/プログラミング学習に適用した例を探したけれど見つからなかった

## 4 The experiment

- 初心者を (プログラミング入門で) つまづかせる 3 点
  - 代入と順次実行
  - 再帰と反復
  - 並行性
- 「代入と順次実行」なんて簡単じゃないかと思えるようだが、実際にこれが障害になっている → これをまず見ようと思った
- 学生が順次実行と代入についてどういうメンタルモデルを持っているかを調べるアンケート
  - 予想: 「最初はモデルはバラバラだが、学習に成功する学生については正しいモデルに収束していくだろう」
  - 作成した問題 → 図 1 (コピーモデル、移動モデル、交換…)
  - 対応するメンタルモデル → 表 1
- 紆余曲折はあったが、結局約 60 人の学生に対して実施
  - テスト 1、テスト 2 → 先に述べた同一のもの
  - テスト 1 → 本当に最初、プログラミングに接する前
  - テスト 2 → 授業開始後 3 週間たってから
  - 最終試験 → 実際にコードが書けるかどうかで評価 (図 5 ~ 図 7)

\*筑波大学大学院経営システム科学専攻

## 5 Reults

- 3つのグループに明確に分かれた
  - consistent --- 44%。ほぼすべての質問に同じモデルで回答
  - inconsistent --- 39%。質問毎にモデルが変化
  - blank --- 8%。空白回答
- 上記は最初のテストの比率。
  - 2回目では inconsittent のうち半分 (11名) が consistent に移動
- 最終試験の結果 (書けるようになったかどうか) との相関
  - 当初 consistent のグループははっきり書けるようになっている (図 12)
  - inconsistent のままのグループははっきりダメ (図 13)
  - inconsistent → consistent のグループはいまいち (図 13)

## 6 Analysis and speculation

- consistent グループにプログラミングを教えるのは無問題
- inconsistent/blank グループに教えるのは極めて困難
- 「プログラミングができるようになるか」の判定試験ができちゃった!!