

HMMMML3:他人を意識した モチベーション向上を考えたプログラミング環境

中橋 雅弘[†] 宮下 芳明[†]

本稿で提案するプログラミング言語 HMMMML3 における超好意的解釈コンパイラは、スペルミス大幅に許容するとともに、複数の解釈が成り立つ場合はその全ての解釈を実行し、タブ切り替えによってユーザーに選択させる。プログラムの正確さをスコア化しランキングサイトにアップロードする機能も持つ。全てのプログラムは HTML5 上で実行されるので、スマートフォンでも実行可能である。これにより、プログラミングする過程においても、その成果を見せるときも、ユーザーはより他人を意識し、それがモチベーション向上につながると考えている。

HMMMML3 : Programming Environment that Makes Users Conscious of Others and Motivates them

MASAHIRO NAKAHASHI[†] HOMEI MIYASHITA[†]

"Super-generous compiler" in programming language HMMMML3 proposed in this paper allows misspelling greatly, that the user can select from all executed interpretations by switching tabs. It calculates the score of the program from the accuracy, and uploads it to the ranking-site. Moreover, all programs can be executed on HTML5 so that they are executable on smart phones; we consider that motivate the user higher.

1. はじめに

本稿で提案するプログラミング言語 HMMMML3 はこれまで開発してきた HMMMML の機能を HTML5 で動作するように拡張したものである。これによりスマートフォン上で自作プログラムを動かすことができ、他人に見せることも容易になった (図 1)。



図1 スマートフォン上で動作する HMMMML プログラム

安村はユーザ指向のプログラミング言語デザインの指針として Programming 2.0 を提唱し、これからのプログラミング言語は人間の特性をふまえて設計されなければならないと述べている[1]。しかし現在のほとんどのプログラミング言語は、極めて厳密な構文指向であり、守らなければならないルールが非常に多くだけでなく、それらをひとつでも間違えると実行結果を表示すらしてくれない。こうした制約はプログラミングを学び始めたばかりの人にとって大きな障害であり、

学習に対するモチベーションの低下につながると考えている。著者らはこうした思想から「モチベーションを向上させる言語」としてこれまで HMMMML[2][3](Homei Miyashita's Motivating Multi-Lingual Markup Language)を開発してきた。本稿では、この言語をさらに発展させた HMMMML3 について述べる。

2. 関連研究

プログラミング教育といっても、概念の理解を重視することもあれば、C 言語や Java などの汎用的な言語に繋げられるような理解を重視することもある。その違いによって、対象の年齢層、指導方針は大きく変わってくる。目的に沿った形で、プログラミング教育の研究はこれまでに数多く行われてきた。

井後らは学生が C 言語に対して苦手意識を持つのは、プログラムが英単語で記述されているからと考え、日本語プログラミング言語「なでしこ」を用いて C 言語への導入を図った[4]。長らも対象を大学生とし、Java への移行を考えたプログラミング言語 Nigari の開発を行った[5]。Nigari は宣言を省略し、プログラムを自動的に可視化することで、プログラムの流れの直感的理解を促した言語である。これら 2つの研究とは違い、よりビジュアルなインターフェースを用いて子供へのプログラミングの概念教育を考えたプログラミ

[†] 明治大学 理工学部 情報科学科

Department of Computer Science, Meiji University

ング言語が Scratch である[6]. Scratch はあらかじめ用意された簡単なスクリプトを, ブロックを組み立てるように組み合わせることでプログラミングを行うものである. これにより子供でも楽しんでいるうちに条件分岐, 繰り返し, オブジェクト指向といった考え方が身につく. 原田が開発したビジュアルプログラミング言語 Viscuit は, 絵を描きマッチングを指定するだけでプログラミングを行うことができる[7]. マッチングには Fuzzy Rewriting を用いており, これまでのビジュアルプログラミング言語にはみられなかった柔軟な動きを実現した. これにより子供でもプログラミングの楽しさを容易に知ることができる. Viscuit はコンピュータの凄さを伝えることを目的としている.

3. HMMMML

HMMMML (Homei Miyashita's Motivating Multilingual Markup Language)は単純かつ直観的な文法表現を工夫したものである. AT 車的な支援思想に則り, 前章で述べた Scratch[6]や Squeak[8], MAX/MSP/Jitter[9]に代表される直感性を追及したビジュアルなインタフェース, あるいは例示プログラミング手法[10][11]までは検討せず, あくまでテキストを入力してコンピュータを動作させる範疇で支援のデザインを行ってきた.

HMMMML で3回 Hello と表示するときには, 「forタグ」を使用して以下のように書ける.

```
<for 3> print "Hello" </for>
```

カウンタとしての変数の宣言や初期値設定;条件式;増分処理を指定する必要がない.

マウスカーソル位置に半径 20 の円を描画するには
`draw.circle(mouse.x, mouse.y, 20);`

と書ける. エディタに備わっている無限ループボタンをオンにして実行すると, 図2のようなペイントツールを1行で実現できる. 無限ループは ESC キーやウィンドウの閉じるボタンでプログラムを終了することができる.

さらに, セミコロンの有無やドット抜け, 括弧の閉じ忘れについても許容するコンパイラを搭載しているため, 先ほどのプログラムは

```
drawcircle(mousex,mousey,20
```

と書いても同じ結果を得ることができる.

if 文の条件式内で=の代わりに==を使用してしまっても==と解釈し, 大文字小文字の混同や配列変数を含む宣言忘れについても許容して動作する.

say "Hello"と実行すると人工音声で"Hello"と発音し, google"Hello"と実行すると"Hello"の検索結果をブラウ

ザに表示する. MIDI で音を鳴らすには, play.note(ノート番号), 指定したサイズの Google マップを表示するには draw.map(緯度, 経度, 横幅, 縦幅)と1行で行える. 動画ファイル・音楽ファイルも全て play"ファイル名"で実行する.



図2 HMMMML を用いた 1 行プログラミングの例

4. HMMMML2

HMMMML は「好意的な解釈によってとりあえず実行する」ものだったが, HMMMML2 では「好意的な解釈によってとりあえず実行し, その後でコンパイラがどこを修正したのかをユーザに教える」ものに発展させた. 「そのこのゲッキョク駐車を右に曲がって」と指示されたタクシー運転手に例えるなら, 「そんな駐車場はありません」と言って正しく言い直すまで右折しないのがこれまでのコンパイラ的思考であり, 「月極駐車場のことを指しているんだな」と好意的に解釈して右折するのが HMMMML 的思考である. しかしこれでは, 乗客は漢字の読み方を間違えたままであり, 学習の促進を考えたとき, 最も理想的なのは, とりあえず右折してから「ほんとはツキギメって読むんですよ」と小声で教えてくれる運転手ではないだろうか.

HMMMML2 では, if 文の条件式内で「a=1」と書いた場合, これを「a==1」と好意的に解釈して実行した上で, ソースコードには「a=1 を a==1 に修正しました」とコメント文が入るようにした (図3).

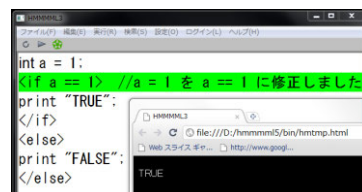


図3 解釈結果の実行後提示

これまでのプログラミング言語やその開発環境では, エラーとなっているところに下線を表示し「実行前に直させる」インタラクシオンデザインとしていたが,

HMMMML2 は「早く実行結果が見たい」というユーザの気持ちを優先させ、まず実行し、そのあとで正しい表現が何であったかを伝えるのである。

4.1 超好意的解釈コンパイラ

HMMMML2 では、さらに好意的な解釈を行う「超好意的解釈コンパイラ」を搭載した。ひとつのミスも許さない超頑固なコンパイラの対極の存在を生み出すことが出来れば、「超好意的」から「超頑固」までの広い範囲で支援度合の段階的調整ができるので、プログラマのスキル向上を促すインタラクションデザインにつなげられると考えた。

はじめてプログラミングに挑戦する人に多いのが、全角スペースを入力してしまった場合である。HMMMML2 では、プログラムに直接関係ないと思われる全角スペースや2バイト文字は全てコメントとして好意的に無視するようにした。

HMMMML2 では、スペルミスに対しての許容幅を拡張した。予約語にない命令、例えば `ptint "Hello"` という文字列があったとき、これは `print` を意図して記述されている可能性が高いので、コンパイル時に `print` と置き換えて実行するようにした。コンパイラ内部のアルゴリズムとしては、予約語との一致率を計算して類似した予約語に置換している。

HMMMML2 の言語仕様では、`p` で始まって文字列が続く命令が `print` しかないため、`p "Hello"` と記述しても `print "Hello"` と解釈される。

これに対し、`s "Hello"` と書いた場合は、`say`(人工音声で読み上げ) と `show`(ダイアログ表示) の2通りが考えられる。HMMMML2 では、複数通りの解釈がなりたつ場合は、その全ての解釈を実行することとした。それぞれタブを切り替えることで実行結果を見ることができ、「採用」ボタンを押すことでその解釈を選択することができる(図4)。

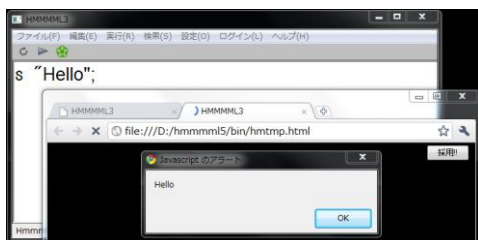


図4 複数解釈に対するタブ表示

たとえば図4において、タブ1を選択すると「Hello」が読み上げられ(`say`としての解釈)タブ2を選択すると「Hello」というダイアログが表示される(`show`としての解釈)。タブ1を選択して採用ボタ

ンを押すと、ソースコードは

```
say "Hello" //s をsay に修正しました
```

という内容に修正されている。

また、現状のシステムで好意的に解釈できないような特殊なエラーが発生した場合は、そのプログラムをサーバ上にアップロードする機能を備えた。これにより集められた多くのプログラムからエラーの典型を導き出し、超好意的解釈コンパイラの解釈機能強化に役立てることができる。

5. HMMMML3

HMMMML2 までは、あくまでもユーザ個人に対してのモチベーション向上だった。HMMMML3 では他人を意識したユーザのモチベーション向上を考えた。

人間は自分の実力を他人に自慢したい、認めてもらいたいという欲求を少なからず持っている。例えば、著者らの研究室で毎年行われているプログラミング発表会では、研究室のメンバーを驚かせたいという理由だけで、異常なモチベーション向上が起きている。また、中田らの調査によると、Linux 開発に携わるボランティアのユーザのモチベーション向上には、他人に認められたいという欲求が大いに影響していることが示されている[12]。人工的なシステムがどれほどユーザに対して配慮したとしても、他人による「生きた」フィードバックには敵わないと著者らは考えた。そこで以下の拡張を行った。

5.1 ランキング

HMMMML3 では、プログラムの正確さの評価を行い、スコアとして換算し、ユーザの成績をサーバにアップロードする機構を設けた。図5のランキングサイトで全国のユーザのプログラミング成績のランキングが閲覧できる。HMMMML エディタに `twitter` のアカウントでログインしていれば、ユーザ名にはそのアカウント名が表示され、それをを用いてユーザ同士で健闘を讃え合うこともできる。また、これによりプログラマのコミュニティが容易に形成できるのではないかと考えている。

	User Name	Score
1	J	4400
2	tettetete	1100
3	seikosama	1000
4	The_Lus	900
5	kumasun	700
6	tmtact	700
7	Irakkuma	600

図5 ランキングサイト

5.2 スマートフォンへの対応

また、HMMMMLのほとんどの命令をHTML5に移植することに成功した。例えば、前章でも紹介したsay命令の場合、Google Translate APIの読み上げ機能(非公開API)を用い、ここから返ってくるmp3ファイルをhtml5の<audio>要素で再生することで実現した。今までのHMMMMLとインタフェースは同一で、実行ボタンを押すとChromeが立ち上がって実行する仕組みになっている。

以下に図1で紹介したプログラムのソースコードを示す。このプログラムは、ランダムな場所に球を配置し、その球にぶつからずに長い軌跡を描くほど得点が高くなるゲームである。ゲームオーバーになると、指定したURLにジャンプするようになっている。

```
score=0;
<for 99>
  draw.sphere(random(400),random(400),random(20));
</for>
<while>
  <if get.red(mouse.x,mouse.y)!=0>
    draw.box(0,0,400,400)
    show "SCORE = "+score
    go "http://miyashita.com"
  </if>
  score = score+10;
  set.color(0,255,0);
  draw.circle(mouse.x,mouse.y,3);
</while>
```

そして完成したプログラムは、エディタの実行ボタンのとなりにある星形のボタンを押すだけで、特定のURLへとアップロードされる。プログラムは実行ボタンからももちろん実行できるが、ブラウザにそのURLを打ちこむことでも実行できる。すなわちスマートフォンを所持した人が近くに一人でもいれば、自作のプログラムをいつでも、どこでも見せることができるのである。このようにプログラムを容易に他人に見せられる環境を提供したことで、ユーザは自分が作ったプログラムを用いて誰かを喜ばせたり、楽しませる体験ができる。そこから得られるものこそユーザが一番知ってもらいたいものであり、その体験はユーザのプログラミングに対するモチベーションを著しく向上させると考える。

6. 展望

今後は、超好意的解釈に依存しないための追加機能

として、超好意的解釈-超頑固解釈を調整できるスライダを考えている。このスライダを用いて、好意的解釈に依存しないほど高得点を得ることができるシステムにすれば、支援のための補助輪をユーザが自ら外し、より高度なプログラミングの世界に飛び出してくれるはずである。

本システムは著者らが主観的に定めたデザインポリシーに基づいて試作されたものにすぎないため、今後さらに改良を重ね、これまで数多く提案されてきているエンドユーザプログラミング支援手法とも比較した実験やワークショップを行っていきたい。

参考文献

- 1) 安村通晃. Programming2.0: ユーザ指向のプログラミング, 情報処理学会夏のシンポジウム2006.
- 2) 宮下芳明. プログラミングに対するモチベーションを向上させる新言語 HMMMML の開発, 第51回プログラミング・シンポジウム予稿集, pp.57-64, 2010.
- 3) 中橋雅弘, 宮下芳明. HMMMML2: 超好意的に解釈するコンパイラ. 夏のプログラミングシンポジウム2010, 長野, 2010年.
- 4) 井後宏康, 原佑輔, 松江信太郎, 吉末千紘. 「なでしこ」によるプログラミング言語の導入, <http://mis.edu.yamaguchi-u.ac.jp/kaisetu/sotsuron2007/n-ihmy-resume.pdf>
- 5) 長慎也, 甲斐宗徳, 川合晶, 日野孝昭, 前島真一, 笈捷彦. Nigari-Java 言語へも移行しやすい初学者向けプログラミング言語, コンピュータと教育研究会報告, pp.13-20, 2003.
- 6) Scratch: <http://scratch.mit.edu/>
- 7) 原田康徳. 子供向けビジュアル言語 Viscuit とそのインタフェース, ヒューマンインタフェース研究会報告, pp.41-48, 2005.
- 8) Cardelli, L. and Pike, R. Squeak: a language for communicating with mice. In Proceedings of the 12th Annual Conference on Computer Graphics and interactive Techniques SIGGRAPH '85. 1985.
- 9) Max/MSP/Jitter <http://cycling74.com/products/maxmsp/jitter/>
- 10) 増井俊之. インターフェイスの街角(47) - 例示プログラミング. Unix Magazine. Vol. 16, No. 11, 2001.
- 11) Hartmann, B., Wu, L., Collins, K., and Klemmer, S. R. Programming by a sample: rapidly creating web applications with d.mix. In Proceedings of UIST2007. 2007.
- 12) 中田智之, 串田洋介, 二本柳寿史, 佐久間明彦, 古澤理. Linux ハッカーのモチベーション. http://www.waseda.jp/sem-inoue/file/archives/2004_Motivation_of_Linux.pdf. 20004