

# 教育用プログラミング実行環境「ますめ」の設計と実装

荻野 哲男<sup>1,a)</sup> 藤岡 健史<sup>2,b)</sup>

**概要:** 情報教育の問題解決型学習の中でプログラミングを行う際に、C 言語や Java 言語のようなテキストベースのプログラミングを利用することは、その言語自身の学習が問題になることが多い。一方、Squeak eToys のようなビジュアルベースのプログラミングは、初学者に対し効果的であるものの、高度なアルゴリズムの実装には向いていない。そこで、高等学校における情報教育での利用を目的とし、テキストベースとビジュアルベースの特徴を取り入れた新しい教育用プログラミング実行環境である「ますめ」を設計し、実装を行った。

## Design and Implmentation of MASUME: Programming and Runtime Enviroments for Imformation Science Education

TETSUO OGINO<sup>1,a)</sup> TAKESHI FUJIOKA<sup>2,b)</sup>

**Abstract:** For learning information science and practical skills, experience of the programming is efficacious in high schools. However, text-based programming languages, such as C or Java, are not suitable because of difficulties to master those languages. Therefore, graphic-based programming languages, such as Squeak eToys, are utilizable for beginner of programmings. In this study, we develop new programming environment called "MASUME" to accept both advantages of text-based and graphic-based programming laungages. In this paper, we explain it's concept of design and implementaion for new programming enviroment.

### 1. はじめに

ソーシャルネットワークの形成やネットワーク犯罪など、社会が情報技術に大きく影響されるなか、初等中等教育においても「情報」を学習することが重要になってきている。一方、初等中等教育段階における情報教育は発展途上の段階にあり、その多くが、計算機の操作方法を学習したり、情報に関する知識を学習することにとどまっている。そこで、情報と情報技術の科学的理解とその活用能力の育成を目標として、情報教育にPBLを導入している実践も報告されている [1]。PBLは問題解決を目的としているため、問題を解決するための道具として必要な、情報に関する知識を習得し、情報を活用する能力を育成するという点におい

て効果的であるが、さきほどの実践では、既に存在する情報システムをどう活用するかに主眼が置かれている。筆者らは、情報技術の科学的理解とその活用などの実践力を育成するためには、PBL型の学習で、プログラミングを用いたモデル化やシミュレーションにより問題解決が出来ることを実践させるのが効果的であると考えている。

初等中等教育におけるプログラミングを取り入れた授業の実践として、論理的な思考と問題解決能力を育むことを目的とした Squeak eToys の活用 [2] や、オブジェクト指向の学習が可能な「ドリトル」[3]、フローチャートを用いてプログラミングを容易にした初学者用プログラミング学習環境 PEN[4] などがあげられる。このようなプログラミング環境は、いずれもプログラミング初学者を対象とし、学校教育での利用が容易であるが、モデル化とシミュレーションによる問題解決を目的としたものは少ない。一方、問題解決としてプログラミングを伴うシミュレーションを行う際の環境としては、C 言語や Java 言語を扱う事例が多く、学校教育では実施が困難である。そこで、モデル化

<sup>1</sup> 神戸大学情報基盤センター  
Information Science and Technology Center, Kobe University

<sup>2</sup> 京都市立西京高等学校  
Kyoto Municipal Tonan Senior High School

a) togino@port.kobe-u.ac.jp

b) t-fujioka@edu.city.kyoto.jp

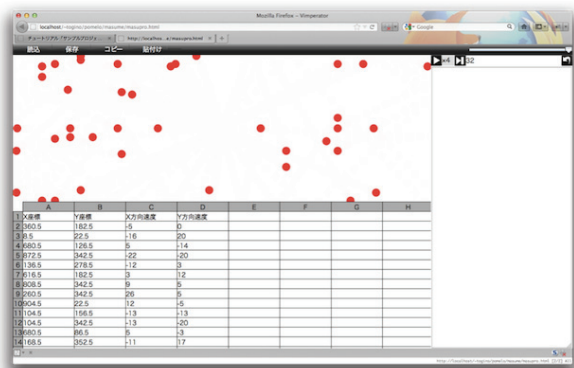


図 1 教育用プログラミング実行環境「ますめ」

とシミュレーションによって問題解決を目指す授業の実現を目的として、初等中等教育で実施可能な新しいプログラミング実行環境が必要であると考え、教育用プログラミング実行環境「ますめ」の設計と実装を行った。

## 2. 教育用プログラミング実行環境「ますめ」

情報教育におけるモデル化とシミュレーションを実現する新しいプログラミング実行環境として図1にあげる「ますめ」を開発した。この「ますめ」の基本的な設計は、表計算アプリケーションに着目し、そのセル1つ1つにプログラミングを行うという点である。「ますめ」の特徴を以下にあげる。

- 個々のセルにプログラミング
- 最小限のプログラミング言語仕様
- サーバクライアント型のブラウザアプリケーション
- グラフィックやサウンドのサポート
- 詳細な粒度での操作活動の記録

これらの特徴により、Squeak eToys などの環境で逐次実行や条件分岐などのプログラミングの基本を学んだ生徒が、C 言語や Java 言語などの実践的なプログラミングとのギャップに躓くことなく、モデル化とシミュレーションを用いて計算機を活用した問題解決を経験させることが容易になる。

さらに、この環境では、生徒の躓き等の学習プロセスを分析するため、生徒の操作履歴を詳細に記録できるようにしている。

### 2.1 個々のセルにプログラミング

高等学校の教科情報などにおいて、計算機を用いたシミュレーションの学習を行う場合、表計算アプリケーションである Microsoft Excel を利用することが多い。その理由として、ほとんどの計算機にインストールされており、授業で使うにあたっての準備が比較的容易であることがあげられるが、それに加え、シミュレーションは多くの数値データを扱う観点から、表計算アプリケーションとの相性

が良いことも考えられる。

例えば、セルに  $= A1 + B1$  と入力すると、数式とみなして A1 セルの値と B1 セルの値を加算した結果にそのセルの値が変更される。また、その数式で参照されるセルの値が変更されると自動的に再計算が行われる。この自動的に再計算される機能はシミュレーションを行う際にとっても有効である。

「ますめ」では、より複雑なプログラミングを実現するために、セルに数式や関数を入力するところを拡張し、セル1つ1つにプログラミングができるようにしている。すべてのセルは値とプログラムの2つを保持し、プログラムの中で参照している他のセルの値が変更された時と、明示的に指示した場合にプログラムが実行され、プログラムの実行結果がセルの値となる。

シミュレーションにおいて、パラメータとなる変数の値のそれぞれに、その値が変化する条件をプログラミングでき、多くのプログラムが並列的に実行されるという環境は、エージェントベースの考え方をプログラミングするのに適している。

### 2.2 最小限のプログラミング言語仕様

プログラミング言語は、それぞれの目的に応じてさまざまな機能や仕様が定められている。計算機の低レベルな操作を可能にするポインタ操作や、プログラムの保守性や再利用性を高めるためのクラスやその継承、安全で効率的なプログラムを実現するための型や、逆に自由で拡張性の高いプログラミングを目指して型の概念がないものなどである。「ますめ」では、これらの言語仕様の多くが、実際の問題解決のためのシミュレーションを行う上で不要であり、また、このような仕様が存在することでプログラミング言語自身の習得が目的になりがちが点を考え、必要最小限のプログラミング言語仕様を1から設計することにした。

「ますめ」を使う上で必要と考えられる言語仕様は、下記にあげるものであり、プログラミング言語としては、汎用のものと比べ、非常に基本的な仕様のみである。

- 四則演算や条件判断などの演算子
- 数値・文字列・真偽値およびオブジェクト型
- 変数およびオブジェクトへの識別子による値の保存と読込
- 関数呼び出し
- 条件分岐や繰り返しなどの制御構造

### 2.3 ブラウザベースのオープンで柔軟な実行環境

高等学校のような計算機環境において、教師が環境を整えるためには、計算機に対する豊富な知識と経験を必要とする。このことが情報教育の中で、モデル化とシミュレーションを生徒に体験させることを困難にしている一つの理由でもある。そのため、以下の点をふまえて「ますめ」の

設計を行った。

- 使用する計算機の OS などに影響を受けないこと
- インストールなどの作業が不要であること
- 学校だけでなく、自宅などでも利用できること

計算機の OS としては、Windows をはじめ Mac OS X や Linux などさまざまなものが存在している。多くは Windows であるが、Windows 7 や Windows XP などそのバージョンによって影響を受けることも少なくない。このような環境の影響を最小限に抑えるため、HTML/CSS/JavaScript を用いたブラウザベースの実行環境を採用した。ブラウザは多くの計算機に導入されており、ブラウザの種類によって違いが生じることもあるものの、規格が統一されていく方向にあるため、最も適したプラットフォームであると判断した。

また、ブラウザベースであっても、Flash や Silverlight のようなフレームワークを導入することで、よりリッチな GUI を提供することが可能になるが、逆にフレームワークの導入や管理などの追加の作業が必要になるため、このようなフレームワークを採用しなかった。

ブラウザベースの実行環境とし、その作業をサーバ側に保存できるようにすることで、授業で作成したプログラムを自宅の計算機で開いて続きの作業が行えるようになる。また、サーバ・クライアント方式にすることで、このような自宅での作業に対する操作履歴も記録できるので、分析の対象を広げることができる。

## 2.4 グラフィックオブジェクトの利用

ドリトル [3] のようなプログラミング環境では、プログラムの実行結果が視覚的に得られるため、何が起こったのか理解しやすいという特徴がある。具体的には、「かめ太! 100 歩く」というプログラムを実行した場合、内部的には「かめ太」の座標を示す値が 100 加算される訳であるが、学習者には、「かめ太」の画面上での表示位置が変化するという動きで理解することとなる。

このように、プログラミング変数が扱う値を数字で表示するだけでなく、グラフィックオブジェクトの位置座標や大きさ・色などに関係付けて表示できる機能が必要である。

## 3. 実装

「ますめ」はブラウザ上で動作するアプリケーションであり、JavaScript で書かれた (1) 仮想マシン (2) コンパイラ (3) GUI フレームワークから構成される。

### 3.1 仮想マシン

「ますめ」に実装された仮想マシンは、コンパイラによってコンパイルされたセルのソースコードを下記の規則に従って実行する。

- 実行規則 1: ソースコードの編集が終了した時に実行

する

- 実行規則 2: ソースコード内で参照した他のセルの値が変化した時に実行する

多くのプログラミング環境では、実行ファイルを生成してそのファイルの実行を指示したり、実行ボタンやコマンドを入力するなど、プログラムは明示的に実行される必要がある。一方、「ますめ」では、プログラムが実行されるべきタイミングをあらかじめ定義し、プログラムは暗黙的に実行される。この設計により、プログラムの変更が、実行結果にどのような変化をもたらすのか、リアルタイムに視覚化され、プログラムの間違いなどに気が付きやすくなる。

#### 3.1.1 実行順序の問題

「ますめ」はセル 1 つ 1 つにそのセルの値を決めるソースコードが記述でき、それらが独立して実行可能である。そこで、実行規則 2 によって実行されるセルが複数ある場合、その実行順番によって異なる結果になる場合がある。例えば、B1 セルと C1 セルがともに A1 セルの値を参照し、かつ B1 セルが C1 セルの値を参照している場合、A1 セルの値が変化した時、B1 セルと C1 セルが実行されることになる。理想的には B1 セルと C1 セルは独立して並列実行されるべきであるが、実際の仮想マシンは 1 つの処理装置しかないため、どちらかが先に実行される。B1 セルが先に実行されると、C1 セルが実行される前の値を参照し、逆に C1 セルが先に実行されると、実行後の値を参照することになるため結果が異なってしまう。

この問題を解決するために、仮想マシンの実行環境にターン番号を導入し、下記のような実行規則を追加した。

- 実行規則 3: 各セルはターン番号毎に値を保持し、ターン  $N$  での実行時には、値の参照はターン  $N$  として行い、値の変更はターン  $N + 1$  に対し行う。
- 実行規則 4: 実行規則 2 で実行されるセルは、ターン  $N + 1$  で実行する。

このターン番号により、同一ターンでは実行順に関係なく同じ値が参照されるため、同じ結果が得られる。

#### 3.1.2 循環参照の問題

循環参照とは、あるセルのソースコードが他のセルの値を参照していることを、セル間の有向パスで表現した場合に、そのグラフが木構造ではなく、ループ状態が含まれている状態を指す。なお、あるセルのソースコードが自分自身のセルの値を参照するという自己参照は、一番シンプルな循環参照の例であると言える。このような循環参照がある場合、「ますめ」が実行規則 2 に従ってプログラムの実行を行うと、プログラムの実行が終了しないことが起こる。

循環参照とプログラムの実行について、表 1 にあげた例 1 や例 2 では、1 回もしくは何回かの実行でプログラムの実行が終了するが、例 3 では、値は収束していくものの終了することはない。また例 4 は値が振動するため、これも終了することはない。

表 1 循環参照とプログラムの実行

例	A1 のソースコード (初期値 1)	実行
1	= A1	1 回で終了
2	= If A1 < 10 Then A1 + 1 ELSE A1 END	10 回で終了
3	= A1/2 + 1/A1	$\sqrt{2}$ に収束
4	= A1 * -1	1 と -1 を振動

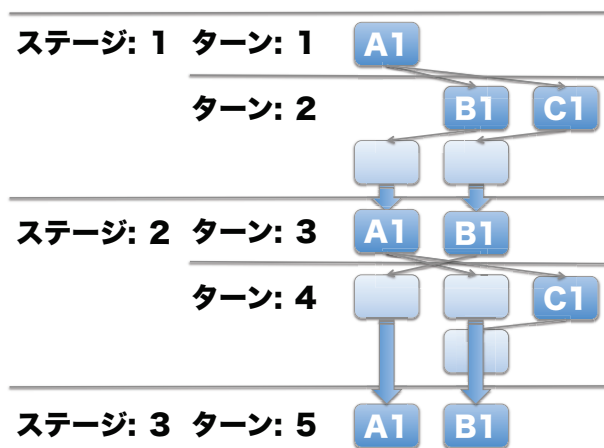


図 2 ステージ番号とターン番号による実行制御の例

この問題について、表計算アプリケーションである Excel では、反復回数や値が変化するとみなすきい値を設定し、上限のある複数回の計算を行うことで対処している。しかし、「ますめ」でのプログラミングにおいて、循環参照を意図的にプログラムする理由を考えると、例 2 のように、シミュレーションにおけるタイマーとしてステップ実行をさせたい場合と、例 3 のように、値を取束させて求めたい場合があり、両者の区別を機械的につけることはできない。そこで、実行したくないのに実行してしまうことと、実行したいのに実行してくれないことを比較し、意図しない実行を防ぐという観点から、実行環境にステージ番号を導入し、下記のような実行規則を設け、手動でプログラムの実行を制御することとした。

- 実行規則 5: 同一ステージでは、同じセルは 1 度しか実行されない。実行規則 2 にて 2 度以上実行する必要がある場合は、実行キューにセルを追加する
- 実行規則 6: ステップ実行ボタンが押された時、ステージおよびターンを 1 つ増やし、実行キューにあるセルを実行する

以上の実行規則に従い、例えば A1 セルの値を B1 セルと C1 セルが参照し、また、B1 セルの値を A1 セルが、C1 セルの値を B1 セルが参照する関係あるものを実行した様子が図 2 となる。

A1 セルのソースコードが編集されると、実行規則 1 によりターン 1 で A1 セルが実行される。その結果、実行規則 2 により B1 セルと C1 セルを実行することになるが、実行規則 4 により、それはターン 2 となる。C1 セルの実

行によって値が変化しても、実行規則 3 により、値の変更はターン 3 として保存されるため、B1 セルが参照する C1 セルの値はターン 2 となり、その影響を受けない。

B1 セルと C1 セルの実行の結果、実行規則 2 により A1 セルと B1 セルを実行することになるが、実行規則 5 により、実行されずキューに追加されることになる。この時点で、実行するセルがすべてなくなり、実行処理が終わる。次に、ステップ実行ボタンが押されると、実行規則 6 により、ターン 3 で、実行キューにある A1 セルと B1 セルが実行される。

#### 4. おわりに

本研究では、プログラミング初心者にも扱いやすいグラフィカルなプログラミング環境と、テキストベースの実践的なプログラミング環境の間を埋める新しいプログラミング実行環境である「ますめ」の設計と実装を行った。この環境を利用することで、高等学校の教科「情報」で求められる、問題解決においてモデル化やシミュレーションを活用するという目的を実現することが可能になると考えている。

今後の課題としては、「ますめ」を利用した授業実践を行い、具体的な問題解決の教材や授業設計を積み重ねていくことである。また、「ますめ」上での活動はすべて自動的に記録される設計になっているため、それらを分析することで、生徒の学習プロセスを解析し、つまづき発見や生徒の状況に応じた動的な対応を可能にしていきたいと考えている。

#### 参考文献

- [1] 明井上: PBL 情報教育の学習効果の検証 (特集: 若手の会, PBL(Project-Based Learning)), 情報処理学会研究報告. 情報システムと社会環境研究報告, Vol. 2007, No. 25, pp. 123-130 (オンライン), 入手先 (<http://ci.nii.ac.jp/naid/110006249700/>) (2007).
- [2] 卓弥稲垣, 和広阿部, 謙介山崎, 耕二横川: 「教具」としての Squeak eToys とその小学校算数教育への適用, 情報処理学会研究報告. コンピュータと教育研究会報告, Vol. 2009, No. 15, pp. 57-63 (オンライン), 入手先 (<http://ci.nii.ac.jp/naid/110007131905/>) (2009).
- [3] 兼宗 進, 御手洗理英, 中谷多哉子, 福井真吾, 久野靖: 学校教育用オブジェクト指向言語「ドリトル」の設計と実装, 情報処理学会論文誌. プログラミング, Vol. 42, No. 11, pp. 78-90 (オンライン), 入手先 (<http://ci.nii.ac.jp/naid/110002726068/>) (2001-11-15).
- [4] 知博西田, 章原田, 亮太中村, 友介宮本, 敏雄松浦: 初学者用プログラミング学習環境 PEN の実装と評価 (教育用および初心者用言語, 特集: 情報教育～理念・理論・実践～), 情報処理学会論文誌, Vol. 48, No. 8, pp. 2736-2747 (オンライン), 入手先 (<http://ci.nii.ac.jp/naid/110006386503/>) (2007).