

九州大学における 一般情報処理教育支援システムについて

峯恒憲¹ 佐藤周行² 正代隆義³ 廣川佐千男³ 有村 博紀³

森雅生³ 篠原歩³ 竹田正幸³

九州大学大学院システム情報科学研究科

¹ mine@al.is.kyushu-u.ac.jp, ² schuko@c.csce.kyushu-u.ac.jp,

³ {hirokawa,shoudai,arim,masa,ayumi,takeda}@i.kyushu-u.ac.jp

あらまし

九州大学では、毎年およそ 2,300 人の新生が情報処理の入門教育を受講する。これらの学生は、Windows95 の動くパソコン上で、プログラミングを始めとする情報処理の基礎を学ぶ。個人差が大きくなりがちな情報処理を学ぶ学生の習熟度に対処するため、講義をサポートし、かつ、講義時間外に利用できる自学自習用のホームページを用意した。このページでは、我々のこれまでの情報処理教育の経験から得た講義のエッセンスを再現 (Lecture Capture) している。しかし、プログラムが如何に動作するかを理解させることは難しかった。

そこで、我々は、プログラムの動作を理解を助ける機能として、「プログラムアニメーションコンパイラ」と呼ぶ Pascal プログラムを Java Applet に変換するシミュレータを開発した。これにより、Web のブラウザを通してプログラムの動きを体験し、実感してもらえるようになった。

本稿では、本システムのアイデアと概要、並びに今後の開発予定について報告する。

On-Web-Teaching 2300 Students Information Processing and Computer Literacy at Kyushu University

Tsunenori Mine, Hiroyuki Satoh, Takayoshi Shoudai, Sachio Hirokawa,
Hiroki Arimura, Masao Mori, Ayumi Shinohara, Masayuki Takeda
Graduate School of Information Science and Electrical Engineering,
Kyushu University

Abstract

We are developing a Web system for the introductory course of programming and computer literacy. 2300 students take the course with this Web system at Kyushu University. Our main object of this system is to reappear our lecture in front of students in their private studies. Teaching how programs run is a hard problem to deal with in information processing educations. In order to deal with such a difficult subject easily, we propose a program animation compiler for assisting students to understand a motion of programs. This compiler systematically translates tiny Pascal programs into Java Applets. In this paper, we state our idea and future plans of the system.

1 はじめに

九州大学では、情報処理の入門教育を入学したての学生およそ2,300名に対して実施している。この講義を、理系の学部(理、工、農、医、歯、薬)に所属する学生のうちのほとんどのものが必修科目として受講し、また、文系の学部(文、教育、法、経済)に所属する学生も、選択科目としてではあるが、ほとんどの学生が受講している。

この情報処理入門は、約60名を一クラスとして、週1回90分の実習で半期(13回)の演習として開講されている。このようなクラスが、年間39クラス開講され、常勤、非常勤を合わせ17人の教官と、各時間に1人以上のTA(Teaching Assistant)がついて、この講義を行っている。このような講義形態になったのは平成6年度からであるが、それ以前にも我々は、1年次、2年次の半分以上の学生に情報処理入門の講義を行って来た。この過程で蓄えられた大量のレポートや質問は、講義をする上での貴重な資料として使われている。

1996年4月からWindows95をOSとするパソコンが導入されるに伴い、九州大学情報処理教育センターは、全ての学部学生にパソコンとワークステーションを使用するためのアカウントを配付した[1]。これら全てのパソコンはインターネットに接続され、九州大学の学生であれば、特別な設定をすることなくインターネットの機能を利用することができる。特に、電子メールやホームページ作成は、我々が行っている情報処理入門の重要なテーマの一つである。

インターネットの普及は、カリキュラムの改編だけでなく、教育そのものを変えつつある。我々は、これまでの情報処理教育の経験を踏まえ、Webのプラグイン機能をフルに活かした情報処理入門のためのホームページ(Computer Literacyのページ)を作成した¹。このホーム

ページはアクセス制限をつけていないため、インターネットに接続しているどの端末からでも見ることができる。このシステムの開発の要点は、これまで行なって来た一般情報処理教育についての知識と技術を、いかにインターネットとWindowsの時代に対応させるかであった。我々は、教官が行う講義のエッセンスを出来るだけ再現しようと試みた(Lecture Capture)[2, 3]。

昨年一年間を通して利用してみた感想を講義担当教官に尋ねたところ、講義における資料として、ならびに自習用の教材として有用であるとの意見が多く、役立っていることがわかった。しかし、このページから得られる情報は学生にとっては受身的なものであり、プログラム作成に於て最も重要なことの一つである、「プログラムが如何に動作するかを理解させる」ための情報が不足しているとの意見もあった。プログラムの動作を説明するには、実際にパソコンでデモンストレーションする必要があるからである。しかし、そのためには教官(あるいはTA)がついている必要がある。さもなければ、自分で苦勞して実践しなければならない。

そこで、プログラムの動作理解を助ける機能として、まず、Pascalプログラムから、そのプログラムの動作のシミュレーションをするJava Appletを作る方式を開発した。これにより、インターネットワークに接続された端末でJavaを使えるブラウザを持つ端末であれば、Pascalプログラムの動きを体験し、実感できるようになった。

以下では、このPascalプログラムのシミュレータを加えた本システムの構成、及び今後の開発予定について記す。

¹<http://www.rc.kyushu-u.ac.jp/Literacy>

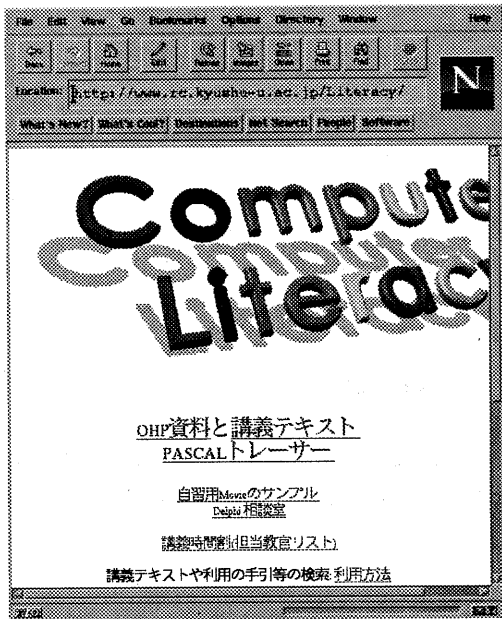


図 1: OHP システムと講義資料を中心とする Computer Literacy のページ

2 Computer Literacy のページ

我々は、情報処理教育のために必要なエッセンスをまとめた OHP スライドと講義資料を用意している。OHP の内容は、講義中に板書されるべき内容そのものである。これらは半期ごとに見直され、改善されてきた。我々は、まず、この OHP と講義資料の持つアイデアや教えるための「こつ」をなるべく損なわないことを目標に、システムを作成した。

このシステムは、講義を再現した OHP システムと、それをサポートする講義資料の 2 つの部分から成る。

OHP 資料

講義は半期 12 回を想定している。各回ごとに OHP スライドの現物が 10~16 枚程度用意されている。これらは、我々が昨年度まで実

際に使用してきた OHP スライドである。ほんの少しではあるが冗談めいたものもある。

166 枚の OHP スライドは日本語 LaTeX で記述されている。この各スライドを Web 上で公開するにあたり、各々をまず Post Script 化し、それを gif 化した。これにより、教官が実際に使用する OHP スライドと全く同じものを Web 上で提供できた。更に、講義時間外でも講義のエッセンスが掴めるよう、以下の機能を加えるように工夫した。

1. 非常に簡単なアニメーション

例えば、実際の講師が講義するのと同程度のスピードで、指先や小さいボールの絵が、OHP の文章上を動くようにした。実際の講義では、講師の手や足の小さな動きが、学生の注意力を促すためである。九州大学情報処理教育センターでは、WWW のブラウザとして、主に Netscape を使用している。これを実現するために Shockwave を Netscape のプラグインとして用いた。

2. 非常に簡単で短い音声

講師自身が吹き込んだ 5~10 秒ほどの生の声を聞けるようにした。アニメーションと同様、学生の注意力を促すのが目的である。

3. 非常に簡単なクイズ

各講義ごとに非常に簡単な選択式の習熟度テストを付けた。マウスで候補をクリックすることで、解答することができ、さらに採点結果を報告する。Web の機能である cgi を使用している。

4. 日本語による検索

後述する講義資料や、情報処理教育センターの手引などの中で調べたいページを探すために、検索機能を組み込んだ。この検索システムは、自然な日本語文を入

力とし、係り受け情報を利用した解析を行うなど、高機能なものとなっている[4]。ただし、OHP 資料は、gif 化していたため、検索対象から外していた。

最近の講義では、情報処理関係の講義以外でも、OHP スライドを利用することがある。OHP を利用することで、講義の流れが良くなることが多い。しかし、一方、学生から見れば、「講義が速い」、「字が見えない」などの不満も起こりうる。こういった不満は、OHP をデジタル化し、Web 上で公開したことにより、消滅させることが出来たようである。

当初、OHP 資料は gif 化していたために、OHP スライドそのままの形態で、公開することができたが、スライド上の文字やプログラムをコピー&ペーストすることができなかつたり、検索システムの対象外となるなどの不便があった。そのため本年度は、できるだけテキストファイル化し、これらの問題を解消した。その作業では、まず latex2html のプログラムを利用し各ページとページ間のリンクを作成した。その後、OHP スライドとして不適な部分を手作業によって修正を加えた。

講義資料

従来、講義に用いてきた資料集をホームページ上で公開した。この講義資料は元もと MS-DOS 上のワードプロセッサで作成しており、デジタル化されてはいたが、資料としての意味合いが濃く、目次や索引などが付けられていなかったり、文体が統一されていなかった。

昨年度、情報処理教育センターでの機種更新に伴い、機種依存部分の内容変更を余義無くされたこともあり、この資料の TeX 化に踏みきり、講義用教科書としての体裁を整えると共に、これを HTML 化した。HTML 化に当たっては、latex2html を利用したが、本の TeX 化に当たって特殊な命令を定義し利用していたため、section 以外の十分な変換はでき

ず、殆どを手作業で行った。このため、本年度改訂した本の修正内容が、いまだ反映されずにいる。

この講義資料は、OHP 資料の所で述べたように、検索システムの対象となっており、必要な箇所を自然な日本語で検索することができる[4]。また、OHP 資料を gif ファイルから、テキストファイルに直したことに伴い、OHP 資料と連動したシステムとして利用可能となった。

3 利用統計

どの程度、我々のシステムが利用されているのかを調べるために、統計機能を組み込んだ。

月別

平成 8 年 5 月から平成 9 年 4 月までの 1 年間の利用統計を表 1 および図 2 に示した。

月	ヒット数	ファイル数	サイト数	情報流出量
97/4	31719	23693	414	192699
97/3	2641	2281	106	13830
97/2	776	705	44	6734
97/1	30901	24367	332	201064
96/12	17778	14914	318	101066
96/11	24675	21005	317	140032
96/10	21037	18152	305	126032
96/9	3033	2328	124	18445
96/8	2360	1771	68	11913
96/7	5112	4371	209	27412
96/6	10145	8972	248	46300
96/5	18067	15667	294	76976
Total	168244	138226	2779	962503

表 1: 平成 8 年度月別利用統計

我々の講義では、半期に 3 回レポートを課している。5 月から 6 月にかけて、第 2 回目のレポート回収が行われる。また、第 3 回目のレポートの回収は、講師に依り夏休み前と後に分かれている。7 月と 9 月のヒット数はその事実を顕著にあらわしている。グラフならびに表より、10 月頃からこの Web のページの利用が定着してきたことがわかる。更に、平成 8

年12月初旬より九州大学以外に公開を始めたことで、1月ならびに4月のヒット数が増加してきている。学外からの利用がトップだった時期もあった。

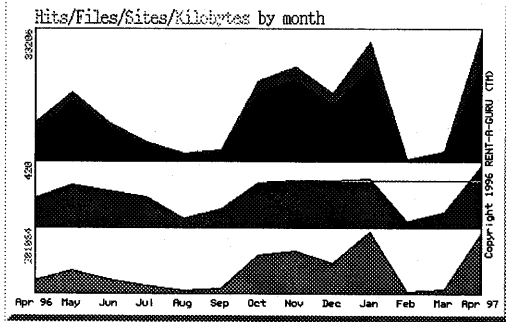


図2: 平成8年度月別利用統計: グラフは上からアクセス数、ファイル数、アクセスしたサイトの数、情報の量 (Kbyte) である。

日別

利用状況が顕著である5月と8月(夏休み中)をあげる。5月は特定の日、特に、火曜日の利用頻度が高い。一方、夏休み中である8月は、10日前後と18日前後にピークがある。学生の帰省状況を反映していると思われる。

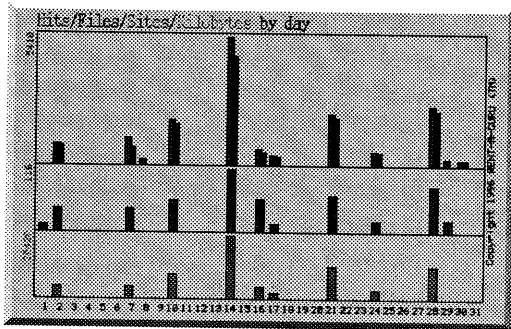


図3: 平成8年度5月利用統計 (横軸は日付けである)

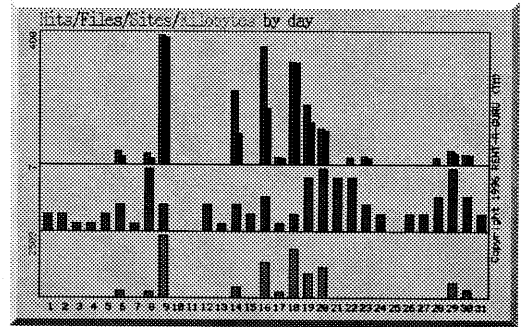


図4: 平成8年度8月利用統計 (横軸は日付けである)

これらのデータから、OHP資料が、講義資料に比べ比較的使用されていること、また、講義内容の推移に伴い、アクセスされる箇所が変わり、講義の自習に利用されていることがわかる。

4 プログラム動作の可視化

先に述べたシステムでは、単に、資料集の公開に留まっており、必ずしも「プログラムの動作を理解させる」ための情報を与えているとは言えなかった。そこで、プログラムの動作の理解を助ける機能を加えることを検討した。

プログラムの動きを教える方法としてはいくつか考えられる。その一つとして、shock-waveなどの機能を利用した、講義内容に関するアニメの作成が上げられる。このための作業としては、まずパソコンを使った実際の講義もしくはデモのビデオ撮りがある。撮影したビデオそのままでは、自習には利用できないため、次に編集作業が必要となるが、この手間は膨大である。しかも、その作業内容は、講義内容や、講義を行う教官に依存する部分が多いため、種々の場合に適用することができず、講義内容などの修正・変更の度に、この作業が必要となるなど、労働単価と比べて割

りに合わない。この作業の自動化については、いろいろと研究されているようであるが、現状では難しいように思われる。

別の方法として、講義そのままではなく、講義内容の中身であるプログラムの実行自体を可視化し、プログラムが如何にして実行されているのか、その動きの理解を助けるシステムの作成が考えられる。このシステムは、言うなれば、商用の言語処理系などの上で実現されているトレーサであるが、このような既存のトレーサは、ほとんどがプログラム開発者向けに作成されており、それ自体、単体のアプリケーションとして動作するものが多い。このようなトレーサは主にプログラム開発者にとって使いやすいように作成されているため、その使い方はプログラムの初学習者には難しく、その使い方自体を新たに学ぶ必要があるなど、負荷が大きいものとなっており、「プログラムの動作を理解する」という目的だけのために利用するには機能が多すぎる。

この目的のためには、簡易型のトレーサなどが Web 上で実現されていることが望ましい。

しかし、プログラムの視覚化では多くの研究がなされている (eg. [5, 6]) が、このような我々の目的に沿ったものは、まだほとんど見当たらない。そこで、Web の使い方がわかれば十分利用できるなど、初学習者にとって優しいシステムを目指して、Pascal プログラムの動作シュミレータを Java 言語を利用して開発することとした。

5 プログラム・アニメーション コンパイラ

Pascal プログラムの Web 上の動作シュミレータを作成するツールを PAC(Program Animation Compiler) と呼ぶ。PAC としては、本来、Pascal プログラムから Java プログラムへ自動変換できることが望ましい。これが実現

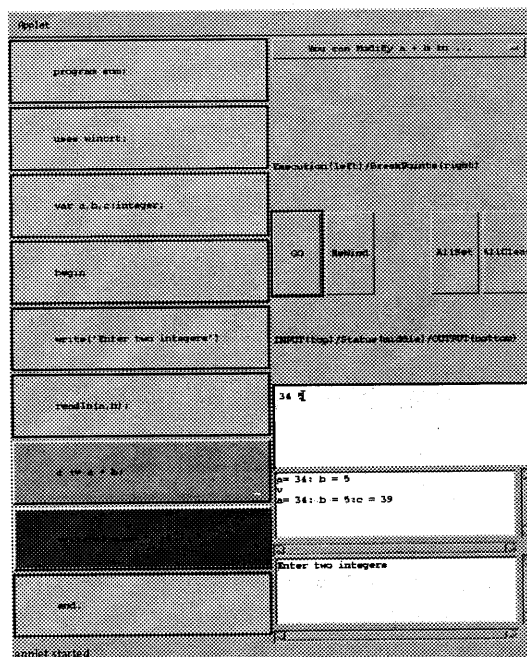


図 5: 2 数の和を計算する Pascal プログラムのトレーサ

できれば、学生に自分が作ったプログラムを入力させ、その実行を Web 上で確認させることができるし、Web 上で行うため、各学生が作成したプログラムを、その学生の希望により登録することも容易であり、Web のページを発展させて行くことが可能となる。しかし、現在のスタッフでは、その実現に時間がかかり過ぎるという問題がある。

しかし、本来、学生に、プログラムがどのように実行されるのか、その実行の様子を確認させられればよいので、まずプロトタイプとして、人間が手作業で作成することとした。ただし、人間が手作業で変換する部分は、できるだけ少なく、かつほぼ機械的に変換できるように設計した。このシュミレータの一例を図 5 に示す。これにより、Web のブラウザを通してプログラムの動きを体験し、実感してもらうことが可能となることを確認した。

更にプログラムの実行を確認するためには、

一つの固定されたプログラムだけでなく、プログラムの中の一部を色々と書き替えて、それを実行して見るのが望ましいため、メニュー方式により、一部の式を替えられるようにしている。

さて、このシミュレータのプロトタイプでは、一旦 Pascal 言語でプログラムを書いた後、それを Java 言語で書き直す必要があるため、作成者の負担が大きい。そこで、その負担を軽減するため、言語に依存しないプログラムの構造(逐次、条件、反復、変数名、入力、種々の関数命令など)を中間言語として規定し、それから、Pascal と Java の両方のプログラムを生成する方法を取ることを考えた。この方式で作成する PAC を、中間言語型 PAC と呼ぶ。この中間言語型 PAC で扱う Pascal 言語の文法の一般形、及び、この形式に基づいて書いたプログラム例を、図 6, 図 7 に示す。

この中間言語型 PAC を利用すれば、プログラムの抽象的な構造を考えながらプログラムをすることができる。また、この中間言語型 PAC は、当初、プログラム構造の概念を十分に理解している Web 上のシステム作成者を対象に設計したものであったが、これを視覚化することにより、

- プログラミングを学ぶ学生自身も利用可能となる。
- 言語に依存しない、プログラム生成器となる。
- プログラム作成に関わる本質的概念の早期理解に役立つ。

などの利点も期待できる。

6 おわりに

本稿では、九州大学で行われている情報処理入門教育の講義ならびに自習用に作成した Computer Literacy のページについて記した。また、プログラムの実行動作を理解させるた

めの Pascal プログラムの動作の可視化シミュレータについて述べた。

このシミュレータのプロトタイプ版では、作成者の負担は軽いとは言えないが、現在開発中の、中間言語型 PAC を実現できれば、その負担も激減され、シミュレータの作成時間が数段短縮されることが期待される。更に、その PAC を視覚化することで、プログラムを作成する上で本質的に必要なプログラムの構造化や、その他の抽象的概念を理解させるツールとして役立つことも期待される。

更に、Pascal から Java 言語への、自動変換 PAC ができれば、学生の作ったプログラムを、容易にアニメーション・アプレットに変換でき、学習意欲を湧かせることに役立つことが期待できる。これらの早期実現が、今後の課題である。

参考文献

- [1] 下園幸一, 原坂佳子, 古川善吾. 九州大学における教育用システムの利用者情報管理について. 情報処理教育研究会講演論文集, pp. 337 - 340, 12 1996.
- [2] Sachio Hirokawa, Tetsuhiro Miyahara, Tsunenori Mine, Takayoshi Shoudai, Masao Mori, Hiroyuki Sato, Ayumi Shinohara, and Masayuki Takeda. Teaching 2300 students with www - practice and experience at kyushu university. In *Proc. of ERI'96*, pp. 59 - 63, 1996.
- [3] 正代隆義, 廣川佐千男, 峯恒憲, 森雅生, 佐藤周行, 篠原歩, 竹田正幸. 九州大学における一般情報処理教育. 情報処理教育研究会講演論文集, pp. 48 - 51, 12 1996.
- [4] 麻生和昭, 峯恒憲, 雨宮真人. 係受け情報を利用した www 上での日本語テキスト検索システム. 言語処理学会第 3 回年次大会論文集, pp. 253-256, 3 1997.
- [5] J. T. Stasko, L. Ford, and M. H. Brown. *Software Visualization Pointers*. <http://sudo-www.ics.es.osaka-u.ac.jp/theme/V/svlinks.html>.
- [6] B.A. Price, R.M. Baecker, and I.S. Small. A principled taxonomy of software visualization journal of visual languages and computing. Vol. 4, No. 3, pp. 211-266, 1994.

```

(program
 プログラムの名前
 (const 定数名 "式")
 (var (型 変数名 ... 変数名)
      ...
      (型 変数名 ... 変数名))
 (begin
  文
  文
  ...
  文))

(begin 文 ... 文)
(if-then-else "式" 文 文)
(if-then "式" 文)
(for 変数名 式 式 文)
(repeat 式 文 ... 文)
(while 式 文)
(read/readln 変数名 ... 変数名)
(write/writeln "式" ... "式")
(assignment 変数名 "式")

```

註) 現在、式をどのように表現するかについては検討中である。たとえば、 $a+b/\sin(x)$ のような式をどのようにして表すかについては、次の2通り考えられる。

1. Lisp
 - の表現を使う。 (+ a (/ b (sin x)))
 - 欠点 : このようにカッコを使って書くのは煩わしい。
 - 利点 : Pascal あるいは Java への変換は楽。
2. 文 字
 - 列としてそのまま表す。 "a+b/sin(x)"
 - 利点 : Pascal の表現のままですむ。
 - 欠点 : 構文解析を行う必要がある。

図 6: 中間言語型 PAC で扱う中間言語表現の一般形

```

; 算術式 (2 数の和の計算)
(program
 sum
 (var (integer a b c))
 (begin
  (write "Enter two integers.")
  (readln a b)
  (assignment c "a+b")
  (writeln "sum = " "c")))

; 条件判定式 (お酒が飲める?)
(program
 drink
 (var (integer n))
 (begin
  (write "How old are you?")
  (readln "n")
  (if-then-else "n<20"
   (writeln "You are too young to drink.")
   (writeln "'Cheer!'"))))

; 定回反復 (九九の計算)
(program
 kuku
 (var (integer i j))
 (begin
  (for i "1" "9"
   (begin
    (for j "1" "9"
     (write "i*j"))
    (writeln))))))

; 不定回反復 (ニュートン法)
(program
 newton
 (const eps "1.0e-6") ; ?????
 (var (real a b))
 (begin
  (assignment a "2")
  (repeat "abs(a-b) <= eps"
   (assignment b "a")
   (assignment a "(b+2/b)/2"))
  (writeln "sqrt(2)=" "a")))

```

図 7: 中間言語型 PAC で利用するプログラム例