

修士論文

プログラム作成状況に応じた
アドバイスを可能にする
プログラミング演習システムに
関する研究



平成 22 年度修了

三重大学大学院工学研究科

博士前期課程 電気電子工学専攻

伊富 昌幸

目次

第1章 はじめに	1
第2章 プログラミング演習の現状	2
2.1. 演習室の現状	2
2.2. プログラミング演習支援システムの現状	3
2.2.1. 三重大学（望月システム）	3
2.2.2. HerculeSS	4
2.2.3. CAPES	5
第3章 提案	6
第4章 要素技術	8
4.1. EDITAREA	8
4.2. AJAX 技術	9
4.3. WEBTTY	10
第5章 提案するプログラミング演習システム	12
5.1. 学習者用画面	12
5.1.1. エディタ	13
5.1.2. プログラムの保存, コンパイル	13
5.1.3. プログラムの実行	14
5.1.4. 講師呼び出しボタン	15
5.2. 講師用画面	16
5.2.1. 演習室状況画面	17
5.2.2. 学習者状況一覧画面	18
第6章 本システムを用いた演習	19
6.1. 本システムの実験	19
6.1.1. 演習内容	19
6.1.2. 分析結果	19
6.2. 実験後行った演習	21
第7章 まとめ	22
参考文献	24

実績一覧	25
謝辞	26

第1章 はじめに

コンピュータの発展に伴い、さまざまな分野でプログラミング能力が必要とされている。プログラムを作成する能力を身につけるためには、実際にプログラムを作成して学ぶことが最適である。そこで、実際にプログラムを作成してプログラムへの理解を深めるプログラミング演習が行われている。しかし、現在行われているプログラミング演習は講師の数が学習者の数に比べて少なく、十分な指導を行えていないと考えられる。そのため、本研究ではプログラミング演習に対し、何らかの支援を行うことにする。

プログラミング演習においてどのような支援を行うことができるかを考察するため、プログラミング演習の現状を知る必要がある。そこで、プログラミング演習を観察したところ、プログラムの作成をあきらめたり演習以外のことをしたりしている学習者が演習の早期からいることがわかった。このような学習者は課題のプログラムを作成するために何をしたいかわからない状態であると考えられ、講師が適切かつ早い時点でアドバイスを行う必要がある。また、現在様々なプログラミング演習支援システムが研究されている。しかし、一般的な演習支援システムは学習者がプログラムを提出した後で支援を行っていた。本研究では、講師が学習者に適切かつ早い時点でアドバイスを行うことができる事を目的とし、学習者のプログラムの作成状況を講師が把握できるようにする演習システムを提案する [a-f]。

本論文の構成は以下の通りである。第2章でプログラミング演習の現状について述べる。第3章で本研究の提案について、第4章で要素技術について、第5章で本研究で提案する演習システムについて、第6章で本システムを用いた演習について、第7章で本システムの今後の展望について述べる。第8章で本論文のまとめを述べる。

第2章 プログラミング演習の現状

一般に、プログラミング演習は、講師が学習者にプログラムを作成する課題を与え、学習者が実施する。この時、講師および TA (Teaching Assistant) はプログラムの作成状況を把握するため演習室内を巡回し、プログラムの作成が止まっていたりどのようにプログラムしたらよいか困っていたりする学習者がいればアドバイスをを行う。

2.1. 演習室の現状

どのような支援を行うことができるかを行うことができるかを考察するためプログラミング演習の現状を知る必要がある。そこで、プログラミング演習の観察を行った。観察したプログラミング演習の状況を以下に示す。

- 日時：2008 年 6 月 26 日
- 場所：プロジェクタと 100 台ほどのパーソナルコンピュータのある演習室
- 学習者と講師の人数：学習者 86 名、講師 1 名、TA 1 名
- 演習時間：13:00 ～ 14:30 (90 分)
- 演習の流れ
 - (1) 講師が出欠をとる。(10 分)
 - (2) 講師がプロジェクタを使ってプログラミングの説明をする。(10 分)
 - (3) 講師が学習者にプログラムを作成する課題を与える。
 - (4) 講師及び TA は演習室内を巡回し、アドバイスを必要とする学習者にアドバイスをを行う。(70 分)

15 分おきに見回りをし、目視で判断した結果、表 2-1 がわかった。

表 2-1. 観察結果

	プログラム作成	Web 閲覧	動画閲覧	居眠り	その他
13:30~13:40	78 人	4 人	0 人	4 人	0 人
13:47~13:55	73 人	2 人	4 人	7 人	0 人
14:00~14:05	73 人	3 人	2 人	5 人	1 人

この結果から、学習者が課題に取り組み始める時からプログラムの作成以外のことをやっている学習者が 8 人いることがわかった。また、プログラムの作成以外のことをしている学習者が最も多かったのは、13:47~13:55 と 14:00~14:05 の 13 人であった。これらは課題のプログラムを作成するために何を書けばいいかわからない学習者ややる気がない学習者であると考えられる。このような学習者は演習の始まりから存在するため、講師および TA は適切かつ早い時点でアドバイスをしなければならない。しかし、現状のプログラミング演習では学習者に対し講師らが少なすぎるため適切かつ早い時点でアドバイスできておらず、学習者に対するきめ細かいサポートが行われているとはいえない。

2.2. プログラミング演習支援システムの現状

プログラミング演習を支援するために、文献 [1-7] にもあるようにさまざまなシステムが考案され運用されている。学習者が提出したプログラムを解析し、結果を学習者に提示するシステム [1-5]、学習者が提出したプログラムからプログラミング演習の進捗を講師に提示するシステム [6,7] といったものがある。

2.2.1. 三重大学（望月システム）

過去に三重大学で研究されたプログラミング演習を支援するシステム [1,2] を図 2-1 に示す。

- (1) 演習を始める前にあらかじめ講師は課題をシステムに登録しておく。
- (2) 演習中に学習者はシステムに登録された課題に取り組みプログラムを作成し、プログラムを完成させると学習者はシステムに提出する。
- (3) システムは提出されたプログラムを構文チェックや動作チェックなど間違いがないかを調べ、その結果を学習者に教える。

このように、学習者がプログラムを提出した後を対象としている。そのため、2.1. であげた適切かつ早い時点でアドバイスをする必要を満たすようには実装されていない。

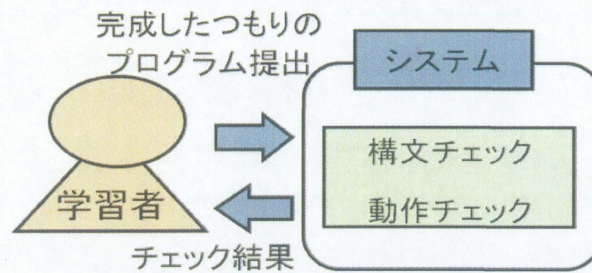


図 2-1. プログラミング演習支援システム

2.2.2. Herculeless

大阪工業大学で研究されているシステムで、講師が学習者の状況をモニタリングするシステム [6] である。Herculeless のクラスの進捗ページを図 2-2 に示す。課題における学習者全体の進捗状況が割合で表示される。学習者の進捗状態は○は正しいプログラム、△は冗長なコードもしくは出力データが正確でない、▲は実行エラー、×はコンパイルエラー、★はファイルが存在しないという状態を表す。進捗ページを講師に提示することで、クラス全体の進捗状況を伝える。進捗ページは学習者が NFS (Network File System) を使って提出したプログラムに対していくつかのテストを行い、そのテスト結果をもとに作られる。学習者がプログラムを NFS を使って提出するまでシステムに情報が届かないため、学習者がコンパイルする以前の状況を講師に伝えることができない。

p1 クラスの進捗

- 課題名をクリックすると、その課題のエラーが多い順に表示されます。

課題	時間	変更	○	△	▲	×	★
work71	10:50	0	54%	11%	30%	21%	12%
work72	10:50	2	78%	4%	2%	8%	10%
work73	10:51	13	84%	0%	0%	5%	10%
work74	10:52	13	53%	0%	7%	15%	23%
work75	10:52	5	0%	0%	0%	53%	48%
work76	10:53	17	26%	0%	2%	15%	55%
work77	10:49	14	0%	0%	8%	17%	72%
work78	10:51	5	0%	0%	0%	4%	94%
work79	10:52	1	0%	0%	0%	0%	99%

このフレームは30秒毎に自動的に更新されますが、学生情報は自動的に更新されません。「更新」ボタンをクリック

[p1 クラスの work77 のエラー集計]

- 57:000000 ★ 課題のファイルが作成さ
- 22:000002 ★ lec07 ディレクトリが存在し
- 11:132402 ▲ (1月)数[7 3 4 3 4]の出
- 9:013210 × work77.c 中に構造体 [poi
- 4:002202 × work77.c 中の関数 fint m

座席	学生	work71	work72
PC1001	e1a06xx1	8-10:08 work71.txt ○	8-10:23 work72.c ○
PC1002	e1a06xx2	8-10:10 work71.txt ○	8-10:22 work72.c ○
PC1003	e1a06xx3	8-10:08 work71.txt × 0f2004	8-09:42 work72.c △ 0b2206
PC1004	e1a06xx4	8-10:10 work71.txt × 0f2004	8-10:23 work72.c ○

図 2-2. Herculeless のクラスの進捗ページ

2.2.3. CAPES

CAPES (Computer-Aided Programming Exercise System) 名古屋工業大学で研究されているシステムで、プログラムの自動テスト [4] と受講者のモニタリング機能 [7] がある。CAPES は学習者をより正確に、より早期に、見落としなく発見するために、作成履歴を習得し保存する。さらに、講師がアドバイスを必要としている学習者を発見するための機能を提供する。学習者の状態は課題毎に表示され、何回目の QA サイクルか、自動テストの結果、編集した関数名がそれぞれ表示される。しかし、システムにプログラムを提出する前といった早い段階において、学習者の状況を提示できているとは言えない。操作履歴からアドバイスを必要としている学習者を見つけ、講師へ提示する方法はまだ実装されていない。

第3章 提案

本研究では講師が学習者に適切かつ早い時点でアドバイスを行うことが出来るようにすることを目的とし、学習者のプログラム作成途中の状況を講師が把握できるようにするためのシステムを提案する。提案するシステムには次の点が必要である。

- (1) 講師がアドバイスを必要とする学習者を即時に把握できる。その学習者について次の2点を講師が把握する必要がある。
 - (ア) 演習室内のその学習者の座席の位置はどこか。
 - (イ) その学習者は作成している途中のプログラムはどのような状況か。
- (2) 学習者が適切かつ早い時点でアドバイスを受けられる。
- (3) 学習者がプログラムの作成、実行がシステムだけでできる。
- (4) システムの導入が容易である。

本研究の目的である講師が早い時点で学習者のアドバイスを行うようにするために、講師がアドバイスを必要とする学習者を即時に把握できるようにする。講師に提示する情報は様々なものが考えられるが、本研究ではアドバイスを必要としている状態を学習者のプログラムが更新されていない時間から判断することとする。学生がプログラミング以外のことをしている場合や、プログラミングが止まっている場合を見つけることができる。具体的に講師に提示する情報は、講師が直接アドバイスに行くときにどこに座っているかを判断するために演習室内でアドバイスを必要としている学生がどこかと、講師が学習者の画面を見てプログラムの状況を判断しなくてもいいように学習者の作成途中のプログラムがどのようなかである。

学習者がアドバイスを受けるには、講師がアドバイスを必要としている学習者を発見をするか、学習者が挙手などで講師を呼び出さなければならない。本システムは学習者が挙手しなくても呼び出せるようにする。

学習者がプログラミングについて学ぶ必要があるため、通常の開発環境と同じようにプログラムの記述、保存、コンパイル、実行ができるようにする必要がある。本システムでは学習者の利便性を考え、すべて本システム上でできるようにする。

プログラミング演習で使用するため、本システムを演習室に導入をしなければ

ばならない。そのため、本システムを導入する手間をできる限り減らし、演習室のシステムに極力手を加えないことが望ましい。

上記を満たすシステムとして、図 3-1 に示す構成を持つシステム PROPEL (PROgramming Practice Easy for Learners) を提案する。このシステムは、自動送信機能を持つエディタを学習者に、学習者のプログラム作成状況の一覧画面を講師に提供する。図 3-1 のシステムを用いた演習の流れは以下の通りになる。

- (1) 学習者はシステムのエディタでプログラムを書く。
- (2) システムのエディタは学習者の作成途中のプログラムを定期的に Web サーバへ自動で送信する。
- (3) Web サーバは受信した作成途中のプログラムを保存する。
- (4) Web サーバは保存されたプログラムからプログラムの作成状況の一覧画面を作成する。
- (5) 一覧画面を講師が閲覧する。

本システムでは、授業への導入のしやすさを考慮し Web ブラウザをユーザーインターフェイスとして用いることにする。これにより、本システムをインストールした Web サーバを用意するだけで、Web ブラウザを使用し、プログラミング演習で本システムを利用することができる。

本システムをプログラミング演習で用いることでシステムにより、講師が早い時点で学習者のコーディング状況に応じたアドバイスを行うことができることが期待できる。

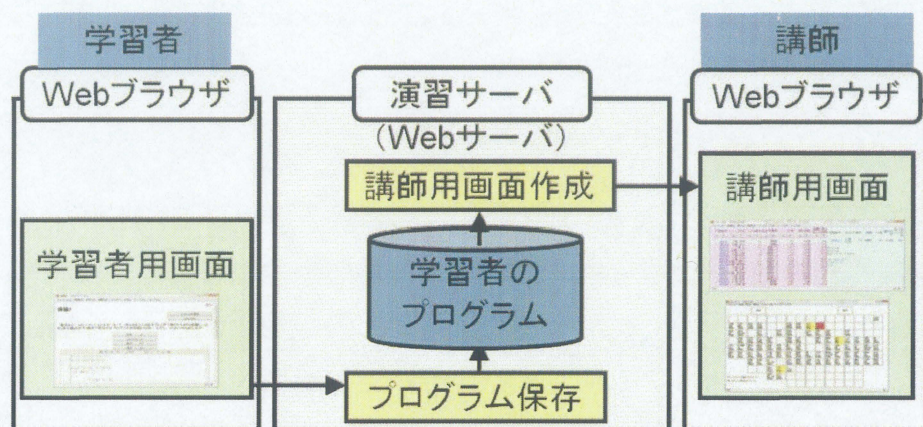


図 3-1. システム構成

第4章 要素技術

本システムは学習者用画面と講師用画面の二つで構成される。学習者用画面はプログラムを記述、保存、コンパイル、実行といったプログラムを作成する機能と学習者が作成している途中のプログラムの保存する機能が必要となる。講師用画面は表示される情報を自動更新する機能が必要となる。

学生用画面はプログラムを記述するためにエディタ、コンパイルするためにコンパイラ、実行するための仕組み、学習者が作成している途中のプログラムを自動で保存する仕組みがそれぞれ必要となる。

エディタに求められる要素は Web ブラウザで動作することとプログラムの記述に適していることである。これらを満たすエディタを一から作り上げるのは難しい。そこで本システムでは EditArea [8] という改変可能でフリーで配布されているエディタを組み込むこととする。

学生がプログラムを保存する機能も通常の開発環境と同じようにする必要がある。通常の HTML (Hyper Text Markup Language) ではプログラムをサーバに送信するたびに画面の遷移が行われ、画面の読み込みの遅さの原因や、エディタのリセットが行われるなど、学習者のストレスとなる。そこで、画面のセインが行われないように保存を行え、Web ブラウザにプラグインを導入する必要があるものとして Ajax 技術 [9] を用いる。画面を表示したまま通信を行い、画面の遷移が行われないようにプログラムの保存を行う。Ajax 技術はほかにも、講師用画面の更新に使用する。

実行するための仕組みとして、サーバで学習者が作成したプログラムを動かす CGI を実行し、その実行結果を Web ブラウザで表示することが考えられる。しかし、この方法だと対話型のプログラムを実行することができない。そこで本システムは対話型のプログラムが実行できるように WebTTY [10] を本システムに組み込む。

4.1. EditArea

EditArea は、JavaScript で書かれた Web ブラウザ上で動作するプログラムのソースコードを編集するためのエディタである。製作者は Christophe Dolivet。プログラム開発のための何らかのシステムのコンポーネントとして利用できるように作られている。ソースコードの保存機能は組み込まれておらず、このコ

ンポーネントを利用する者がカスタマイズして利用できるようになっている。 LGPL ライセンス [11] で配布されている無償かつ改変可能なソフトウェアである。 EditArea を組み込んだシステムの画面を図 4-1 に示す。 EditArea にはソースコードの編集に適した以下の機能がある。

- C 言語の予約語に色がつく。
- 行数を表示する。
- 自動で字下げする。

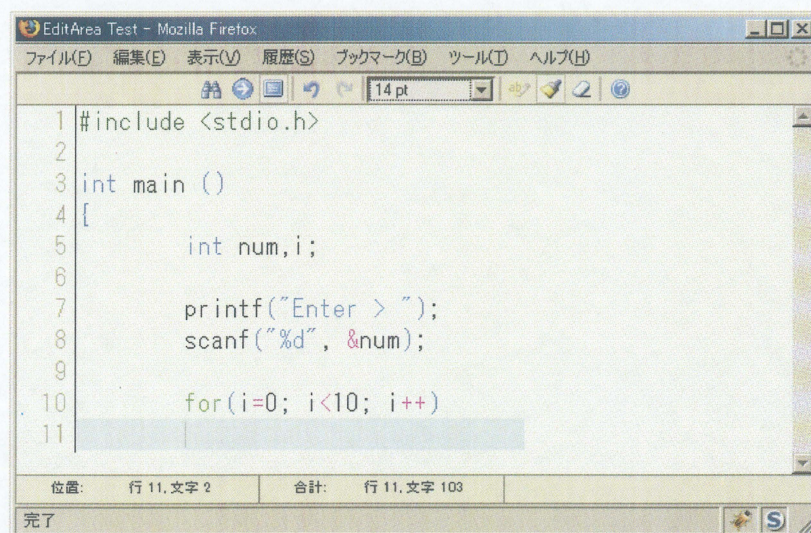


図 4-1. EditArea を組み込んだシステムの画面

4.2. Ajax 技術

Ajax 技術は Web ブラウザ上で動作するアプリケーションにおいて、画面遷移なしに非同期通信を行う方法の一つである。最近の Web ブラウザは、表示画面を生成するレンダリングエンジン、ユーザの操作を受け取るユーザーインターフェイス、非同期通信をサポートする Ajax エンジンを持つ。Ajax 技術が組み込まれた Web ページで行われる処理を図 4-2 に示す。

- (1) ユーザーインターフェイスが Web ページ内で行ったユーザのアクションを受け付け、Ajax エンジンにアクションに応じた指示をする。
- (2) Ajax エンジンがアクションに応じた HTTP リクエストを Web サーバのサーバサイドアプリケーションに送信する。
- (3) Web サーバのサーバサイドアプリケーションが受け取った HTTP リクエストに応じた処理を実行し、結果をデータとして返す。

- (4) Ajax エンジンが Web サーバのサーバサイドアプリケーションから返されたデータを加工し、表示をレンダリングエンジンに指示する。
- (5) レンダリングエンジンが Ajax エンジンによって指示されたとおりに Web ページを表示する。
- (6) (1)~(5) がユーザのアクションごとに実行される。

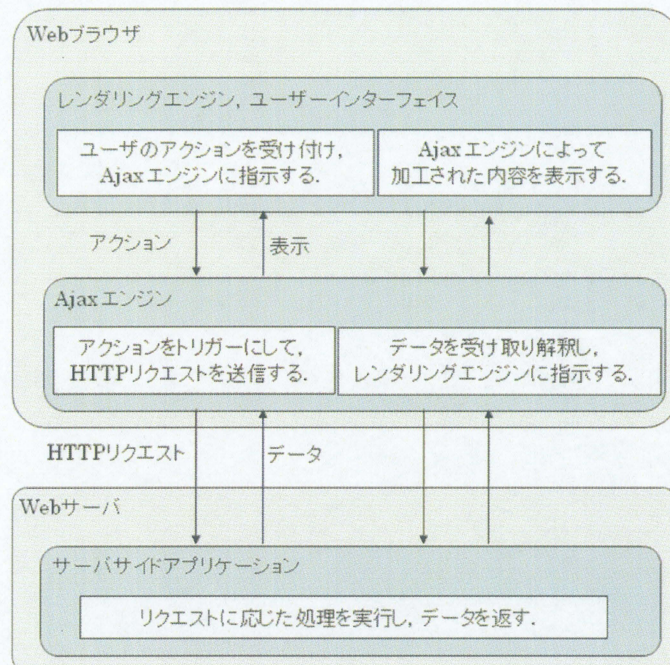
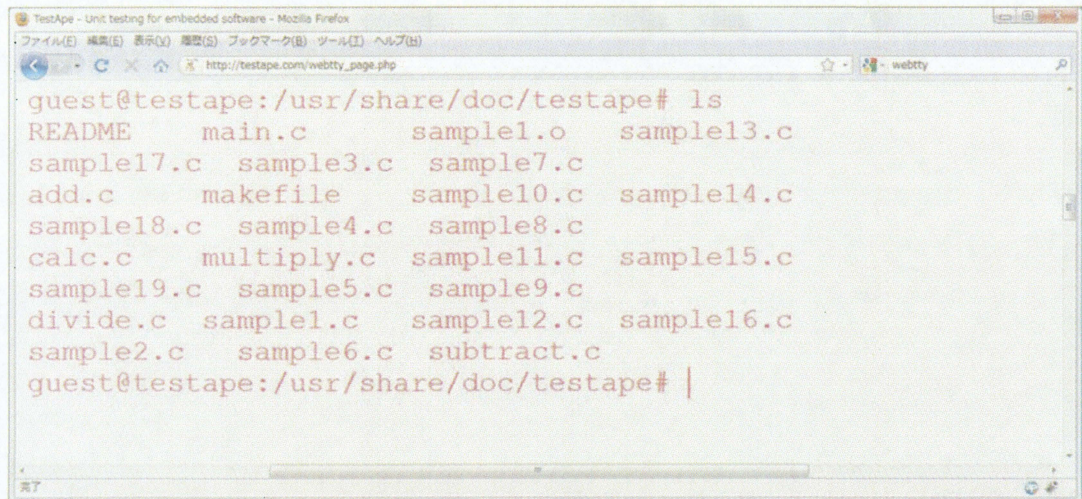


図 4-2. Ajax 技術による処理の流れ

4.3. WebTTY

WebTTY は Web ブラウザからサーバの仮想端末を操作するコンポーネントである。製作者は Martin Steen Nielsen であり、LGPL ライセンスで配布されている無償かつ改変可能なソフトウェアである。図 4-3 に WebTTY が動作する Web ページを示す。このように Web ブラウザから、サーバ上の端末を対話的に操作できる。



The screenshot shows a Mozilla Firefox browser window titled "TestApe - Unit testing for embedded software". The address bar displays "http://testape.com/webtty_page.php". The main content area shows a terminal window with the following text:

```
guest@testape:/usr/share/doc/testape# ls
README      main.c      sample1.o  sample13.c
sample17.c  sample3.c  sample7.c
add.c       makefile   sample10.c sample14.c
sample18.c  sample4.c  sample8.c
calc.c      multiply.c sample11.c sample15.c
sample19.c  sample5.c  sample9.c
divide.c    sample1.c  sample12.c sample16.c
sample2.c   sample6.c  subtract.c
guest@testape:/usr/share/doc/testape# |
```

図 4-3. WebTTY 動作例

第5章 提案するプログラミング演習システム

本研究で提案するコーディング状況に応じたアドバイスを可能にすることを目的としたプログラミング演習システムについて説明する。本システムは自動送信機能を持つエディタを学習者に、学習者のプログラムの作成状況の一覧画面を講師に提供する。

5.1. 学習者用画面

本システムの学習者用画面を図 5-1 に示す。本システムは、講師に新しい情報を提示するため、学習者の作成途中のプログラムを常に保持する必要がある。本システムのエディタは一定時間ごとに学習者が作成している途中のプログラムをサーバに送信する。また、学習者用画面は学習者が通常の開発環境でプログラムを作成すると同等の機能、プログラムの保存、コンパイル、実行ができることが必要となる。学習者はこの学生用画面をつのエディタを使ってプログラムを作成し、通常の開発環境と同じように、プログラムの作成に区切りがついた時、学習者の操作でプログラムを保存する。保存したプログラムのコンパイル、動作確認のためコンパイルしたプログラムの実行、動作確認を行ったプログラムを講師に見てもらうために提出する。

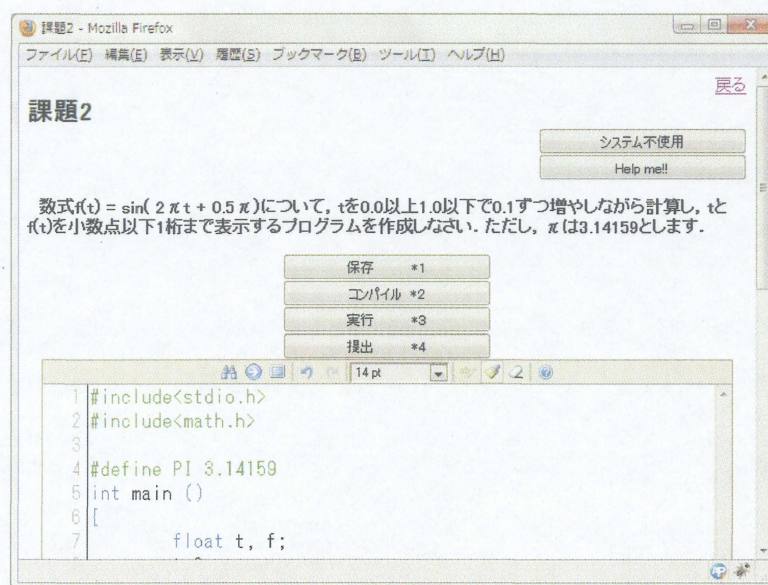


図 5-1. 自動送信機能を持つエディタ画面

5.1.1. エディタ

学習者の作成途中のプログラムを自動的にサーバに送信するエディタを Ajax 技術を用いて EditArea に付け加えることで実現した。

EditArea で記述されている途中のプログラムを自動で送信する方法を図 5-2 に示す。以下の通りにプログラムを自動で送信する。

- (1) 学習者が EditArea でプログラムを記述する。
- (2) エディタを表示する Web ページに埋め込んだ JavaScript のコードが定期的に EditArea で書かれた内容を取得し Ajax 技術を使用してサーバへ送信する。
- (3) サーバ側では送られてきた作成途中のプログラムを保存用 CGI で学習者のプログラム作成状況を記録するためのデータベースに保存する。

Ajax 技術を用いることにより、エディタが動作している Web ページを表示している状態でサーバとの通信が行えるため、学習者がエディタを使用している間でも作成途中のプログラムをサーバに送信することができる。

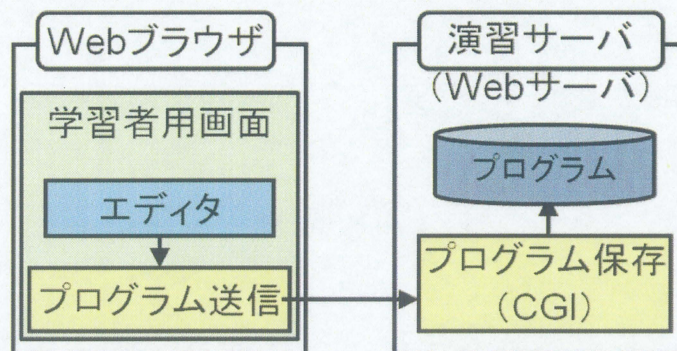


図 5-2. 作成途中のプログラムを自動送信する方法

5.1.2. プログラムの保存, コンパイル

学習者用画面で作成したプログラムを、プログラム作成の区切りがついたところで、学習者の操作で保存、保存したプログラムのコンパイルができるように、Ajax 技術を使用して動作をするようにする。保存ボタンが押されたときの動作は以下のようになる。

- (1) 保存ボタンが押されたイベントを取得する。
- (2) エディタの内容を Ajax 技術を使用してサーバに送信する。
- (3) サーバ側では送られてきた作成途中のプログラムを保存用 CGI で学習者のプログラム作成状況を記録するためのデータベースに保存する。

- (4) 学習者用画面でウィンドウを開き結果を表示する。
コンパイルボタンが押されたときは以下になる。
- (1) コンパイルボタンが押されたイベントを取得する。
 - (2) コンパイル命令をサーバに送信する。
 - (3) サーバ側で gcc を使って保存ボタンを押して保存されたプログラムをコンパイルし，エラーメッセージやコンパイルの成否を学習者用画面に返す。
 - (4) 学習者用画面でウィンドウを開き結果を表示する。

5.1.3. プログラムの実行

学習者用画面でコンパイルしたプログラムの実行ができるように，Ajax と WebTTY を使って学習者用画面に実装した．プログラムを実行している画面を図 5-3 に示す．このように学習者は対話型プログラムを実行することができる．

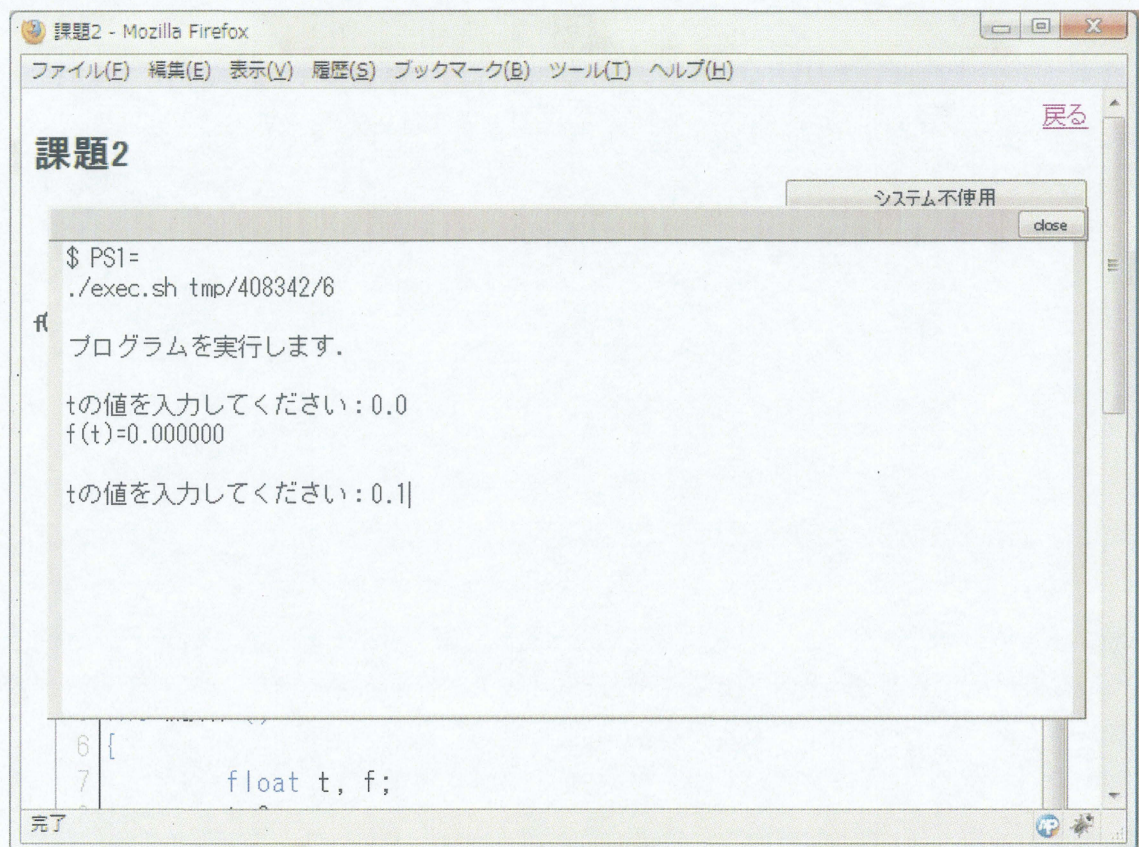


図 5-3. プログラムを実行した画面

以下通りに WebTTY を使って、学習者用画面でプログラムの実行を行う。また、プログラムの実行を行うときのシステムの構成を図 5-4 に示す。

- (1) 実行ボタンを押されたイベントを習得する。
- (2) 演習サーバに WebTTY を実行する命令を送る。
- (3) WebTTY は学習者がコンパイルしたプログラムを実行し、出力を返す。
- (4) 実行ウィンドウを開き、WebTTY からの出力を表示する。
- (5) 学習者の入力があれば、演習サーバ上の WebTTY へ入力を送る。
- (6) WebTTY からの出力を表示する。
- (7) (5)(6)をプログラムが終了するまで、繰り返す。
- (8) WebTTY を終了する命令をおくる。

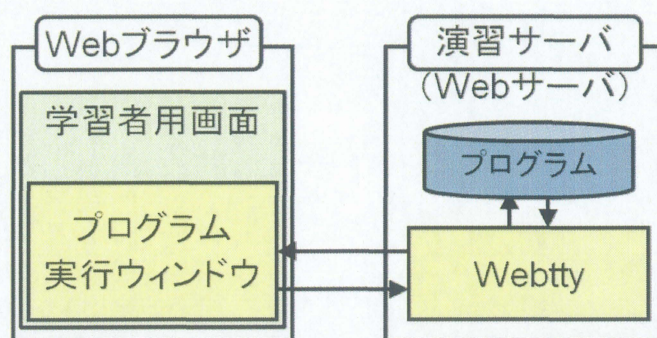


図 5-4. プログラム実行の実現方法

5.1.4. 講師呼び出しボタン

学習者が講師を呼びたい時、講師は速やかに学習者へアドバイスを行う必要がある。そのため、本システムの学習者画面は講師呼び出しボタンがある。講師呼び出しボタンが押されると、その状態がサーバに保存され、講師用画面に反映される。図 5-5 に講師用画面を示す。この画面では講師呼び出しボタンが押された学習者の座席が紫色に表示され、講師がアドバイスを必要とする学習者を把握をすることができる。

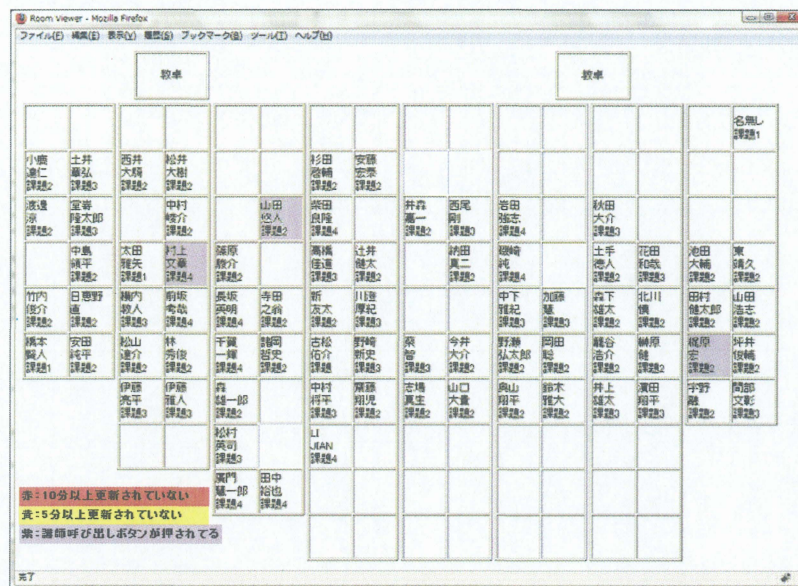


図 5-5.講師呼び出しボタンが押された講師用画面

5.2. 講師用画面

講師用画面はアドバイスを必要としている学習者を講師が即座に把握できることが必要となる。また、この学習者の演習室内での座席位置や現在作成しているプログラムの状況も知る必要がある。そこで、本システムでは座席位置を表示する演習室状況画面と詳しい情報を表示する学習者状況一覧画面をそれぞれ提供する。

講師用画面で常に最新の状態を表示させる方法を図 5-6 に示す。以下の通りに講師用画面を表示させる。

- (1) 講師用画面に組み込まれている JavaScript のコードが Ajax 技術を使用してサーバの画面作成用 CGI と通信を行う。
- (2) サーバの画面作成用 CGI はサーバに保存された学習者の作成途中のプログラムから、行数と更新時を計算し、学籍番号とそれらのリストを作成する。
- (3) 講師用画面に組み込まれている JavaScript のコードが画面作成用 CGI が作成したリストからプログラムの作成途中の状況を集約した画面の右側を生成する。
- (4) 講師用画面に組み込まれている JavaScript のコードが画面の左側を更新するために定期的に画面作成用 CGI を呼び出す。このとき Ajax 技術を使用して、データの通信量を減らし、画面を表示しつつ通信を行い

ちらつきを抑える。

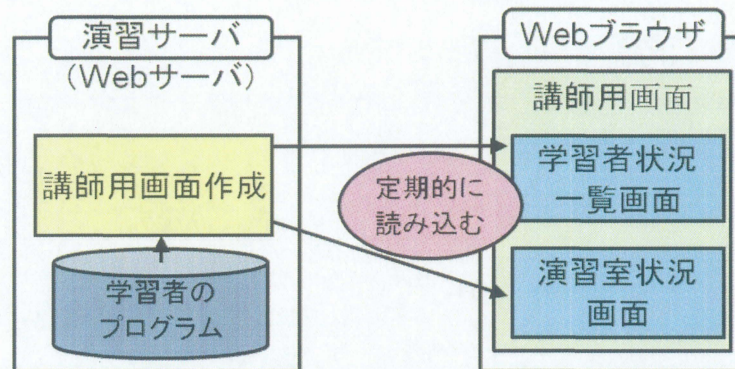


図 5-6. 一覧画面を表示する方法

5.2.1. 演習室状況画面

講師に提供する演習室の状況を表す Web ページの画面を図 5-7 に示す。画面は演習室全体を表し、座っている学習者の学籍番、現在作成中の課題がそれぞれ表示される。また、10 分以上プログラムに変更がない場合赤色、5 分以上変更がない場合黄色、講師呼び出しボタンが押された場合紫色で、色が変化する。講師は色の変化を見てアドバイスを必要としている学習者を判断することができる。

Room Viewer - Mozilla Firefox
 ファイル(F) 編集(E) 表示(V) 書置(S) ブックマーク(B) ツール(T) ヘルプ(H)

2008/08/26 14:00

教卓										教卓										名無し										
小島 達仁 課題2	土井 康弘 課題3	西井 大輔 課題2	松井 大樹 課題2			杉田 浩輔 課題2	安藤 宏幸 課題2	宮林 由佳 課題2	DRU KAOPEI 課題2																					
志遠 淳 課題2	室津 隆太郎 課題3		中村 雄介 課題2		山田 悠人 課題2	所田 良隆 課題4	辻井 敏太 課題2	井森 高一 課題2	西尾 剛 課題3	池田 雄志 課題4		秋田 大介 課題3																		
	中島 雄平 課題2	太田 寛矢 課題2	村上 文章 課題4		藤原 敬介 課題2	高橋 信道 課題3	辻井 敏太 課題2	井森 高一 課題2	西尾 剛 課題3	池田 雄志 課題4		秋田 大介 課題3																		
竹門 雄介 課題2	日野野 雄 課題2	堀内 敬人 課題3	前坂 秀樹 課題4		高橋 敬明 課題2	寺田 之典 課題2	新次太 課題2	川原 博紀 課題3		中下 雅紀 課題3	九郎 雄 課題3	森下 雄太 課題2	北川 慎 課題2	田村 健太郎 課題2	山田 志志 課題2															
橋本 麗人 課題1	宮田 純平 課題2	松山 達介 課題2	林 秀彦 課題2		千葉 一輝 課題4	藤岡 哲史 課題2	古松 佑介 課題2	野崎 新史 課題2	原 啓 課題2	今井 大介 課題2	野瀬 弘太郎 課題2	岡田 聡 課題2	藤谷 浩介 課題2	藤原 健 課題2	佐原 宏 課題2	坪井 俊輔 課題2														
	伊藤 光平 課題3	伊藤 雅人 課題3	森 雄一郎 課題3		森 雄一郎 課題3	中村 雄介 課題2	藤岡 哲史 課題2	志場 真生 課題2	山口 大貴 課題2	奥山 裕平 課題2	鈴木 雅大 課題2	井上 雄太 課題2	濱田 翔平 課題3	宇野 雄 課題2	前田 文彰 課題3															
			八田 裕司 課題3		八田 裕司 課題3	八田 裕司 課題3	八田 裕司 課題3																							

赤: 10分以上更新されていない

黄: 5分以上更新されていない

紫: 講師呼び出しボタンが押されている

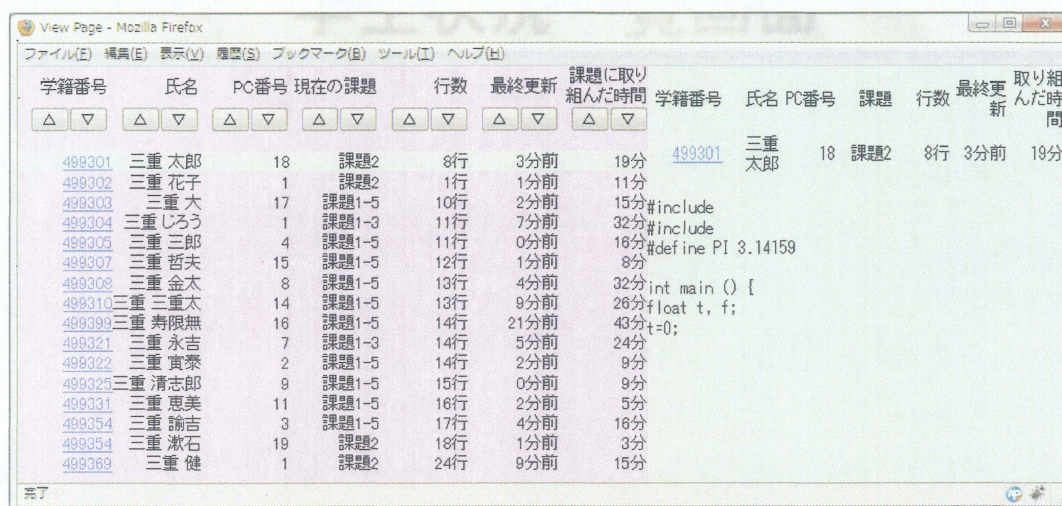
元

図 5-7. 演習室状況画面

5.2.2. 学習者状況一覧画面

講師に提供する学習者のプログラムの作成途中の状況を集約して一覧として表示した Web ページの画面を図 5-8 に示す。講師には、プログラムを作成中の学習者の学籍番号と氏名、その学習者が作成しているプログラムの行数と、学習者が最後に入力したのが何分前なのかという情報の四つを一覧にしたものを提供する。講師が一覧の中から学習者の学籍番号を選択すると、その学習者が作成している途中のプログラムが画面の右側に表示される。一覧は講師が操作しなくても、自動的に最新の状態に更新される。

学習者が作成している途中のプログラムを画面の右側に表示させるために Ajax 技術を使用し、から要求があるたびに Web サーバにある学習者が作成している途中のプログラムを読み込む。一覧を表示したままでも作成途中のプログラムを表示できるようにしている。



学籍番号	氏名	PC番号	現在の課題	行数	最終更新	課題に取り 組んだ時間	学籍番号	氏名	PC番号	課題	行数	最終更新	取り組 んだ時 間
499301	三重 太郎	18	課題2	8行	3分前	19分	499301	三重 太郎	18	課題2	8行	3分前	19分
499302	三重 花子	1	課題2	1行	1分前	11分							
499303	三重大	17	課題1-5	10行	2分前	15分							
499304	三重 じろう	1	課題1-3	11行	7分前	32分							
499305	三重 三郎	4	課題1-5	11行	0分前	18分							
499307	三重 哲夫	15	課題1-5	12行	1分前	8分							
499308	三重 金太	8	課題1-5	13行	4分前	32分							
499310	三重 三重太	14	課題1-5	13行	9分前	26分							
499399	三重 寿限無	16	課題1-5	14行	21分前	43分							
499321	三重 永吉	7	課題1-3	14行	5分前	24分							
499322	三重 寅泰	2	課題1-5	14行	2分前	9分							
499325	三重 清志郎	9	課題1-5	15行	0分前	9分							
499331	三重 恵美	11	課題1-5	16行	2分前	5分							
499354	三重 諭吉	3	課題1-5	17行	4分前	18分							
499354	三重 淑石	19	課題2	18行	1分前	3分							
499369	三重 健	1	課題2	24行	9分前	15分							

図 5-8. プログラムの作成状況の一覧画面

第6章 本システムを用いた演習

本研究では講師が学習者に適切かつ早い時点でアドバイスを行うことが出来るようにすることを目的とし、学習者のプログラム作成途中の状況を講師が把握できるようにするためのシステムを提案した。本システムの有効性を確かめるため、実際のプログラミング演習で運用した。2010年1月19日に三重大学で行われたプログラミング演習Ⅱと2010年度に三重大学で行われたプログラミング演習Ⅰで本システムを運用した。

6.1. 本システムの実験

2010年1月19日に三重大学で行われたプログラミング演習Ⅱで運用をした。演習は1年生を対象に、講師1名、TA1名で行われた。演習は1コマ目と2コマ目で学習者を入れ替えて行い1コマ目は34名、2コマ目は44名に対して行った。

講師には本システムに関する説明を行い、本システムの演習室状況画面のみを使用してもらい、TAは通常の演習と同じように演習室内を巡回してもらった。実験では提案したシステムから自動アドバイス機能をなくしたシステムを使用した。システムは講師用PCに表示させ、講師はこの画面を見て演習室内の状況を把握できるようにした。演習中の様子をビデオで撮影し、それを分析し、サーバに保存されたプログラムを分析した。

6.1.1. 演習内容

演習時間は90分で行われ、はじめの40分に講師から本日の演習について、プログラムに関することなどを説明し、15分で演習システムの利用方法を説明し、25分で学習者が実際にプログラムを作成する演習を行った。演習は配列に関する課題をおこない、参考書に記載されているプログラム例を書き換え、課題で指定されたプログラムを作成するというものである。演習で行う予定の課題は4題用意し、それ以上に出来る人のためにチャレンジ問題を用意した。

6.1.2. 分析結果

表1に撮影したビデオを分析した実験結果を示す。講師がシステムを見てすぐに指導に行ったのは、すべて黄色表示（5分以上停止）されていた学習者に

対してであった。

1 コマ目は、講師はまずシステムを見て、黄色表示されている学習者に教えに行き、黄色表示がなくなると演習室内を巡回し、一通り巡回し終わるとシステムを見るといった形式だった。

2 コマ目は、講師が1回巡回しただけで、ほとんどシステムの画面を見て指導を行う学習者を見つけていた。また、講師がシステムを見た後、演習室全体への指導を行うことがあった。講師が学習者ひとりに指導する時間は30秒から1分であった。

	クラス1 (34人)	クラス2 (44人)
システムを見た回数	11	12
システムを見て指導	6	8
提出以前	6	6
初回のコンパイル以前	2	1
初回のコンパイル以後	4	5
提出以後	0	2
演習室全体に指導した	3	3
巡回中に指導	8	1
学習者に呼ばれて指導(挙手)	1	2

表 6-1. 実験結果

講師がシステムを見て指導した中で、学習者が本システムにプログラムを提出する前だった事例は、1コマ中では6例、2コマ中では6例あり、以前のプログラミング演習支援システムでは支援できなかった段階で本システムが支援できていた。そのなかでも、学習者がコンパイルのためにシステムに保存する前だった事例は、1コマ目では2例、2コマ目は1例あり、非常に早い段階でのアドバイスができていた。本システムを使用することによって、講師は早い時点で学習者にアドバイスをすることができたと考えられる。

その他にも、1コマ目に比べて2コマ目が、システムを見て指導した回数が多く、講師のシステムへの習熟が進んでいたと考えられる。しかし、講師が指導を行った回数はほとんど変わっていない。本システムを使って指導を行うことができる学習者の数は限りがあるので、システムが本当に困っている学習者

を見つけることができるようにする必要がある。

演習終了後に、講師にインタビューを行った。以下にインタビューで聞くことができた意見を示す。

- 進行状況がわかるのがいい。
- システムの説明を短い時間でしなければならない。

講師はシステムの画面を見て演習室全体へ指示を行うことが、1 コマ目と 2 コマ目をあわせて 6 回あり、システムを使って進捗状況を確認していた。また、システムを学習者に説明するのに 15 分かかったため、通常の演習よりも、学習者がプログラムを作成する時間が減った。次回からは説明が短くなるため、学習者がプログラムを作成する時間を確保できると思われる。

6.2. 実験後行った演習

2 年生を対象に行っているプログラミング演習 II において、本システムを運用した。講師 1 名と TA1 名が学習者 76 名を対象に行った。現在の演習では 2 つのモニタに学習者状況一覧画面と演習室状況画面をそれぞれ表示させて、講師は演習室状況画面を見て指導を行った。演習室状況画面見て色の変化がある場合（5 分以上更新がない場合黄色、15 分以上更新がない場合赤色、Help ボタンが押された場合紫色）、講師は速やかに学習者の席へ行き、アドバイスをを行う。色の変化がない場合、講師は学習者状況一覧画面を閲覧し、作成途中のプログラムを見たり、ひとつの課題に取り組んでいる時間を見たりする。このとき、字下げが正しくされていないなどの問題とった学習者全体への共通したアドバイスや演習の進捗状況を把握し、演習室全体への指示をだすなど、演習室内を巡回せずに演習を行うことができた。

第7章 まとめ

プログラミング演習を観察したところ、プログラムの作成をあきらめてしまう学生がいることがわかった。このような学生に対し講師は適切かつ早い時点でアドバイスをする必要があるが、現状は学生の人数に対し講師及びそれをサポートする TA の人数が少な過ぎるために、きめ細かいアドバイスができていない。一方、学生や講師を支援するためにプログラミング演習システムの研究 [1-7] がされているが、その多くは、学生がプログラムを提出した後を対象としているため、学生がプログラムを記述している途中でシステムが支援することができない。

本研究は、講師らが学生に適切かつ早い時点でのアドバイスできるようにすることを目的とし、学生がプログラムを作成している途中でも講師らがその状況を把握することができるシステムを提案した [a-f]。アドバイスを必要としている学習者を、かつ、操作の負担をかけずに講師に提供することが要求される。本研究で提案するシステムは、講師らに演習室全体の状況を即座に把握できる演習室状況画面と、各学生が作成している途中のプログラムを表示するためのプログラム作成状況一覧画面を提供する。一方、学生には自動かつ定期的に作成途中のプログラムをサーバに送信する機能をもつエディタを含む学生用画面を提供する。学生は、この学生用画面で、プログラムの作成に必要な、プログラムの記述、保存、コンパイル、実行の全ての作業を行うことができる。システムは、送信された作成途中のプログラムをもとに、講師用画面を更新し、講師に常に最新の学生らのプログラムの作成状況を伝える。

本システムの有効性を確かめるため、実際のプログラミング演習で運用を行った。その結果、プログラミング演習において本システムを使用することにより、講師らは適切かつ早い時点でのアドバイスができることを確認した。

また、本システムに以下のような改善をすることが考えられる。

- システムが自動でアドバイスする。
- 講師のアドバイスの質をよくする。

講師がアドバイスできる数には限りがあるので、ひとつのアドバイスの質を高めるために、講師が本当に困っている学習者へアドバイスができる工夫が必要である。それにともない、システムが自動でアドバイスできる問題に関しては、システムが自動でアドバイスを行い、講師の負担を軽減することが必要で

ある。また、講師のアドバイスの質を向上させるために、講師用画面から、アドバイスを必要とする度合いを考慮するなど、本当に困っている学習者に速やかにアドバイスできるようにする。

システムを表示させている PC を見なければ、演習室の状況を把握できなかった。講師が学習者にアドバイスをした後、システムを使って演習室の状況を知るためには表示させている PC まで歩かなければならなかった。そこで、iPad [12] や iPod touch [13] にシステムを表示させることが考えられる。これらは無線 LAN 機能やブラウザが標準で搭載されており、演習室内は無線 LAN が使用できるので、これらを使えば演習室内のどこでも、システムを使って演習室の状況を把握できるようになる。図 7-1 に本システムの演習室情報画面を表示させた iPod touch と iPad を示す。

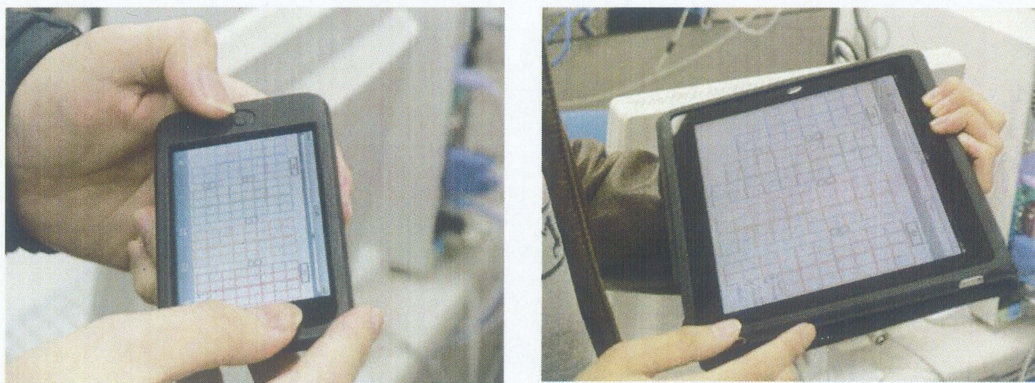


図 7-1. iPod touch (左) と iPad (右) での利用例

参考文献

- [1] 望月将行, 森田直樹, 北英彦, 高瀬治彦, 林照峯, 自動テスト機能を備えたプログラム提出システム, 2003PC カンファレンス (2003)
- [2] 南山幸紀, 山家伊織, 高瀬治彦, 北英彦, 林照峯, プログラミング初心者に間違いの原因に気付かせるためのプログラムの動作の分析結果を用いたアドバイス, 電気関係学会東海支部連合大会 (2008)
- [3] 長慎也, 箕捷彦: proGrep – プログラミング学習履歴検索システム, 情報処理学会研究報告. コンピュータと教育研究報告, Vol.2005, No.15(20050218), pp.29-36 (2005) No.104(20051022), pp.65-71 (2005)
- [4] 中島秀樹, 宮地恵佑, 高橋直久: プログラミング演習支援システム CAPES のための答案評価機構の実現, 情報処理学会研究報告. コンピュータと教育研究報告, Vol.2006, No.16(20060217), pp.127-134 (2006)
- [5] 田上恒大, 安部公輝: 比較的大きなプログラミング課題のための自動採点システム, 情報処理学会研究報告. コンピュータと教育研究報告, Vol.2006, No.16(20060217), pp.135-140 (2006)
- [6] 内藤広志, 斉藤隆: プログラミング演習のための進捗モニタリングシステム, 情報処理学会研究報告. コンピュータと教育研究報告, Vol.2008, No.13(20080216), pp. 33-40 (2008)
- [7] 片桐由裕, 立岩佑一郎, 山本大介, 高橋直久, 受講者の操作履歴の分析機能を用いたプログラミング指導者支援システムの実現, 情報処理学会研究報告. 教育工, vol. 109, No.335(200912), pp. 181-186 (2009)
- [8] EdiaArea(code editor), <http://www.cdolivet.com/editarea/>
- [9] Ajax, http://en.wikipedia.org/wiki/Ajax_%28programming%29
- [10] TestApe(sample of WebTTY), http://testape.com/WebTTY_sample.php
- [11] GNU Lesser General Public License – GNU Project – Free Software Foundation (FSF), <http://www.gnu.org/copyleft/lesser.html> (2007)
- [12] アップル-iPad, <http://www.apple.com/jp/ipad/> (2010).
- [13] アップル-iPod touch, <http://www.apple.com/jp/ipodtouch/> (2010).

実績一覧

- [a] 伊富昌幸, 北英彦, 高瀬治彦, 林照峯, プログラムの作成状況が把握できるプログラミング演習システム, アカデミックフェア 2009, 2009
- [b] 伊富昌幸, 北英彦, 林照峯, プログラムの作成状況が把握できるプログラミング演習システム, 2009PCカンファレンス, 2009
- [c] Masayuki Itomi, Hidehiko Kita and Terumine Hayashi, A Programming Practice System with Function of Grasping Progress of Students' Program Writing, The First International Workshop on Regional Innovation Studies, 2009
- [d] 伊富昌幸, 小島佑介, 高橋功欣, 北英彦, 林照峯, プログラム作成状況の把握機能を持つプログラミング演習システム, 平成 21 年度三重地区計測制御研究講演会講演論文集, B-03, 2009
- [e] 伊富昌幸, 小島佑介, 高橋功欣, 北英彦, プログラムの作成状況を把握する機能を持つプログラミング演習システム, 2010PCカンファレンス, 2010
- [f] Masayuki Itomi, Kouki Takahashi, Yusuke Kojima, Tadashi Ogawa, Daisuke Nishiguchi and Hidehiko Kita., A Programming Practice System with Function of Grasping Progress of Students' Program Writing and Its Experimental Use, Proceedings of the Second International Workshop on Regional Innovation Studies, pp.97-98, 2010

謝辞

本論文は、著者が三重大学大学院工学研究科博士前期課程に在籍中に行った研究をまとめたものである。本研究を進めるにあたり、懇切丁寧な御指導と御督励を賜った三重大学工学部電気電子工学科北英彦准教授，林照峯特任教授に感謝いたします。

また，本研究の実験に協力をいただき，さらに貴重な時間をさいて本研究論文を査読して頂いた，三重大学工学部電気電子工学科森香津夫准教授に深く感謝いたします。

最後に，日頃熱心に討論していただいた計算機工学研究室の皆様方にお礼申し上げます。