

プログラミング演習におけるプログラム共同開発支援

酒井三四郎[†] 高岡 大介[†] 出口 博章^{††} 市村 洋^{†††} 水野 忠則[†]

[†] 静岡大学情報学部 〒432-8011 浜松市城北 3-5-1

^{††} 八戸大学商学部 〒031-0544 八戸市大字美保野 13-98

^{†††} 東京工業高等専門学校 〒193-0997 八王子市梶田町 1220-2

あらまし プログラミング演習において、数人のグループで一つのプログラムを共同で開発することは、プログラミング技能を習得する上で効果がある。参加者間の議論とソースコードの版管理支援に重点をおいた Web ベース協調学習支援システムを提案する。本システムは参加者達が自由に内容を変更追加できるコラボレーションツールである「Wiki」システムを基礎としている。学習者達はソースコードを含んだ Web ページを自由に変更してプログラムを完成させていく。このページは版管理されており、任意の版を表示したり差分を確認したりできる。議論はソースコードを含んだページでソースコードの変更と同時にできる。さらに、掲示板型の Web ページで行うこともでき、ソースコードを参照する時に版を指定できる。

キーワード グループウェア, プログラミング教育, Web アプリケーション, 版管理

Support of cooperative program development in programming exercises

Sanshiro SAKAI[†], Daisuke TAKAOKA[†], Hiroaki DEGUCHI^{††}, Hiroshi ICHIMURA^{†††}, and Tadanori MIZUNO[†]

[†] Faculty of Information, Shizuoka University

^{††} Faculty of Commercial Science, Hachinohe University

^{†††} Tokyo National College of Technology

Abstract It is effective in the master of the programming skill that one program is jointly developed in the group which consists of several learners. The authors propose the Web base cooperation learning support system which gives priority to the communications between participants and support of the version control of the source code. The base of this system is collaboration tool Wiki by which participants can freely change and add the content. Learners freely change the Web page which contains the source code and complete programming. This page is version controlled. Learners can display an arbitrary version, and confirm the difference. The discussion is done at the same time as changing the source code on the Web page which contains the source code. In addition, it is possible to discuss it on the Web page of the bulletin board type. When the source code is referred, the learners can specify the version explicitly.

Key words group ware, programming education, web application, version control

1. はじめに

個人学習で基礎知識を得ることができるが、独善的な意見や、一方的な見解に偏ってしまう恐れがある。また、独りで学習する個人学習では疎外感や孤独感を感じやすく、学習意欲の低下につながってしまう。そこで、適切に他者とコミュニケーションをとりながら学習することも必要である。そこで、本研究では、プロジェクトベース学習をプログラミング演習に適用した。

プロジェクトベース学習に参加する学習者は教授者か

ら出題された課題を共有している。学習者は相互に共通目的達成のために作業を行い、最終的な成果物（解決案）を得ている。このことから、プロジェクトベースの学習には自然な形で協調作業が組み込まれているといえる [1]。

大学の情報系学科ではプログラミングに関する授業が行われている。これは、プログラミング能力を獲得させるのが目的であるが、問題点がいくつか存在する。まず、プログラミング言語の文法を習得することを中心としている点がある。演習では、その文法事項の復習および機能の動作確認で終わってしまう。そのため、実際にシス

テムを作成する能力を養成する段階まで至らない。次に、講義および演習は集合教育に終始する傾向にある。プログラミング能力の個人差は大きいといわれており、プログラミング演習を多数の集合教育で実施すると、能力が低い学習者は実習についていけない場合が多い[2]。

インターネット技術の発展に伴い、電子教科書、遠隔教育など教育支援ツールが多くある。しかし、プログラミング教育に対するインターネットの利用は、文法やアルゴリズムの教育がほとんどである。このような現状において、プログラム設計、プログラムの文書化の教育を中心においた、グループプログラミング支援システムの研究などがなされている[3]。しかし、実際のプログラミング演習においては、ネットワーク上のグループで共同開発するための支援ツールが少ないのが現状である。

また、ソースコードを CVS 等で管理し、議論をメーリングリストや掲示板で行う方法もある。提案するシステムは、この方法に比べて、操作が簡潔で密度の濃い議論が可能になることを目指している。そして、グループで協調してソースコード開発を行う過程において、プログラミングに関する経験的知識と問題解決能力を高め、プログラミング能力を高めることを目的としている。また、プログラムの設計ではなく、むしろコーディングの手法や技法の学習を支援することに重点をおいている。

2. 協調学習システムの提案

2.1 概要

本システムはプログラム言語に対してある程度知識のある学習者が、よりプログラミング能力をつけるために用いる。想定するプログラム演習は以下のような手順で行われる。

- (1) プログラミングの講義を受ける
- (2) 課題が出され、グループで取り組む
- (3) 仕様書の作成
- (4) 開発(コーディング)
- (5) デバッグ, テスト
- (6) 課題の提出

本システムでは、プログラミング演習の(3)~(5)の部分の支援を想定する。特に重点をおいたのは(4)のコーディングの部分で、プログラム共同開発のための機能に主眼を置いた。

本システムは通常の Web ページではなく、Web アプリケーションとして設計され、ページという概念の集合で構成されている。学習者はこのページをあたかも通常の Web ページのように見ることができるが、これはシステムが必要な情報を集めて自動的に生成して表示したものである。

グループのメンバー全員でこのページを共同で編集し、開発していくというものである。各ページには、編集という機能がついており、ページの内容を誰もが編集する

ことができる。編集して保存した内容はそのページにすぐに反映される。また、変更前との差分を表示したり、変更前に戻したりすることも簡単にできる。

このような機能を応用してプログラムの共同開発を行う。プログラムソースのバージョン管理、行番号やキーワードのハイライト表示などでプログラム開発を支援する。また、円滑な議論を行うために、掲示板のようなページで質疑応答を行うこともできる。その際に、ソースのバージョンとの対応を取るためにバージョン指定リンクなどを用いることができる。これによって、後から学習に参加する学習者にとっても議論とプログラム開発の推移が分かりやすい。

本システムは、Linux 上の perl で開発した CGI プログラムである。動作環境としては UNIX 系 OS で動作する Apache Web Server 上を想定している。利用環境は、学習者のプラットフォームを選ばず、通常のブラウザが動作する端末があれば、いつでもどこでも利用することができる。

2.2 Wiki

本システムは、Wiki [4] の理念を基盤に作られた。Wiki とは、Ward Cunningham が 1994 年ごろに提唱した Web システムで、そのコンセプトは、いつでも、どこでも、だれでも、なんでも追加できるというコラボレーションツールである。もとは、Pattern Language の研究で用いられていたものが一般に広がり、様々な WikiClone という Wiki に似たシステムが作られ、協調学習を行う研究も行われている[5]。

これは、一種の Web 掲示板のようなもので、その内容を誰もが編集できる。また、簡単な独自のマークアップによってページを表すことができる。この言語で記述されたページは CGI によって解析され、表示する HTML を動的に生成する。本研究では、この Wiki の考え方を基に、プログラミング演習のための機能を拡張した。

3. システムの設計と実装

3.1 基本機能

システムにアクセスするには、最初にログインする必要がある。ここでは、ユーザー名とパスワードで認証する。これは、あらかじめシステムにユーザーとパスワード、グループを登録しておき、それと比較し認証する。ここで、ユーザー名とパスワードが一致すればそのユーザーのグループが特定され、グループに応じたページが表示される。以下で述べる表示や編集の機能は Wiki を継承している。

3.1.1 表示

システムにログインすると、図 1 のような top というページが表示される。これはすべてのページのルートになるページで、必ずどのグループにも存在する。

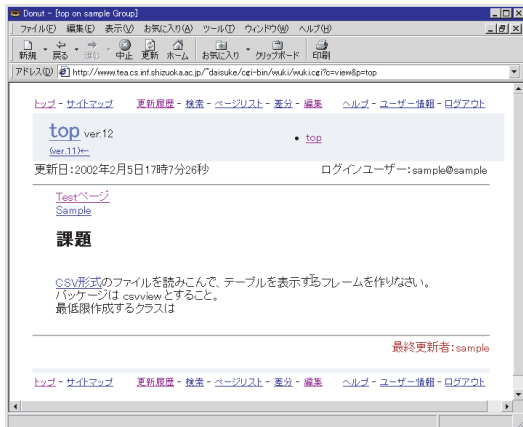


図 1 top ページ

図 1 の上部がヘッダ部で、ここはシステムが自動的に作成するメニューやタイトルが表示される。最上部のメニューはシステムの機能のリストで、ここをクリックして各機能を使用する。その下の背景が付いているエリアはページの情報で、左側に大きく表示されているのがページ名、その横には現在のバージョンが示され、さらに前もしくは次のバージョンへのリンクが表示される。このエリアの右側には現在表示しているページの深さが top ページからの階層で表される。図 1 の場合、top ページは最上部の階層なので、top だけが表示されている。また、その下にはこのページの更新日、ログインユーザーとそのグループが表示される。中程の、罫線で囲まれた部分がメイン部となる。ここはユーザーが自由に編集できるエリアで、基本的にこの内容をメンバーで更新していく。

3.1.2 編集

メニューから「編集」を選ぶと、ページのメイン部がテキストエリアに展開される。ユーザーはこれを自由に編集することができる。ここに書いた内容は、スペースや改行もほぼそのままメイン部へ反映される。また HTML 文を埋め込むこともできる。さらに、本システム独自のマークアップがいくつかある。例えば、'[[' と ']]' で囲まれた文字はページリンクとみなされる。

編集対象のページは編集しているユーザーによってロックされる。もし他のユーザーがロックされているページを編集しようすると、そのページは誰によってロックされているかを提示し、しばらく待つように要求する。これによって、誰かが編集中に、他のユーザーが別の編集を行い、変更過程がずれてしまうというのを防止する。

このように、ページの編集や新規ページの追加、さらに変更を加え、ユーザー同士で協調的にドキュメントを作成していく。

3.2 変更管理機能

3.2.1 差分表示

学習者達はページを次々と更新していく。そのとき、前

のバージョンから、どこが変更されたのかということを知りたい場合にこの機能は役立つ。差分を表示すると、削除された行は薄い色で、取り消し線がかけられて表示される。追加された行は行頭に '+' が付き、太字で背景も強調して表示される。これによって、ページ内のどこが変更されたのかが一目瞭然である。

3.2.2 ページ変更履歴リスト

特定のページのすべての変更履歴リストを表示する機能である。このリストには、各バージョンの更新日、バージョン、指定したバージョンとその直前のバージョンとの差分、指定したバージョンと現在のバージョンとの差分、更新者が表示される。これを用いて、任意のバージョンや、その差分などを表示することができ、特にあとから学習に参加してきた学習者が現在までの流れを把握するためなどに利用できる。

3.2.3 更新情報表示機能

関連ページ群の中でどのページが変更されているかを知るために、更新情報表示の機能がある。更新された日時、更新されたページ、更新したユーザーをテーブルで表示する。更新されたページにリンクが張られているので、目的のページへジャンプすることができる。

システムを利用する一般的なモデルとして、学習者はログイン後にまずこの更新情報を見てどこが更新されたかをチェックする。そして、その更新されたページの差分や、ページ変更履歴リストを見ることによってどういう変更がされたのかを把握する。次に、何かを追加する場合はページの編集を選択し、内容を変更、または加筆修正をする。こういう一連の流れを、学習者それぞれが行い、協調作業によってドキュメントの整備や、プログラムを組み立てていくことを本システムでは想定している。

3.2.4 サイトマップ他の機能

学習者はグループごとに用意される空間に自由にページを追加、変更を加えていく。また、ページは階層構造をもって構成される。そうしたグループごとの学習空間に作られた全てのページを表示するのがサイトマップ機能である。

その他、指定したキーワードを含むページを検索する機能、学習者の個人的なメモやアクセス数、投稿数などの学習履歴の表示機能などがある。

3.3 バージョン指定リンク

ページリンクにおけるバージョンの指定方法は次の 3 つがある。

- [[ページ名]] : 常に最新版を指定
- [[ページ名\$番号]] : 特定のバージョン番号を指定
- [[ページ名\$]] : 現在のバージョンを指定

[[サンプル]] のような普通のページリンクは、そのページの常に最新版へジャンプするようになっている。通常はこれで問題ないが、あるバージョンに対する意見を述

べたいときなど、バージョンを指定できたほうが便利である。そういう場合は、[[ページ名\$バージョン番号]]で、そのページのバージョンを指定してリンクすることができる。例えば「サンプル」ページのバージョン 10 を指定したい場合は、[[サンプル\$10]]と記述する。

通常、あるページに対してなにか意見などを述べたいときは大概、その対象のページの現在のバージョンである。バージョンを指定してリンクしないと、後から見た時に、対象ページのバージョンがあがっていたりすると、話がまったくわからない状況に陥ってしまう恐れがある。しかし、対象ページのバージョンを一々確認するのは大変なので、対象ページの現在のバージョンを指定してリンクできる。[[ページ名\$]]のように、バージョン番号を指定しないと、現在のバージョンへリンクされることになる。例えば「サンプル」ページの現在のバージョンが 12 だとして、[[サンプル\$]]と書いておくと、このリンクを書いたページが保存される際に、システムは「サンプル」ページの現在のバージョンを調べ、[[サンプル\$12]]と書き換えて保存する。こうすることによって、後から参加したユーザーもどういった会話・議論がなされていたかが把握しやすくなる。

また、階層構造に基づいたリンク指定、システム外へのリンク作成、リスト作成、表作成などのために、html に比べて簡易なマークアップ方法を用意している。

3.4 議論支援

現在利用できるページの種類は二つある。一つは、これまで説明してきたような一般的なページで、誰でも編集できる、従来の Wiki と同じようなタイプのページである。このページを作成するには、[[新規ページ]]のように新しいページ名を[[]]で囲み、そこに表示される「?」マークをクリックすれば一般的なページが作られる。

この新しいページへのリンクを作成するときに、ページ名の語尾に“BBS”という文字を付け加えることによって、図 2 のような BBS 型ページを作成することができる。BBS 型のページは、ヘッダ部とメイン部の間に、投稿フォームが表示される。ユーザーはここに名前、メールアドレス、タイトル、コメントを入力し投稿する。投稿された内容は、新しい順に上に積み上げられていく。

このように、BBS 型のページはまさに掲示板のように動作し、議論や報告などに向いているといえる。これによって円滑な議論をサポートする。この BBS 型ページは一般的なページと同じように管理されている。書き込みが増えるに従って、バージョンが上がっていく。実はこの BBS 型ページは通常のページに BBS 型のフィルタをかけて表示しているだけなのである。

現在ではこのようなフィルタ機能は BBS 型しか用意していないが、他にもカレンダー型のフィルタを適用して、

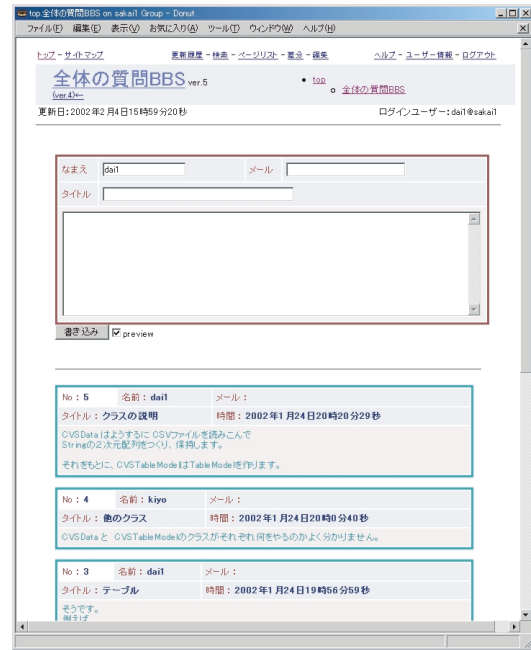


図 2 BBS 型ページ

グループでのスケジュール管理などを行えば、より有用な協調作業支援となるであろう。

3.5 プログラミング支援

プログラミング支援に関する主な機能は、プログラムのキーワードのハイライト表示、行番号表示などのソースコードの表示に関する機能と、プログラムの抽出、版管理、変更コメントなど、ソースコードの管理に関する機能がある。

これらを用いて実質的なプログラムの開発を行い、BBS 型ページとバージョン指定リンクを併用して開発のための議論を行い、コミュニケーションを取る。特にプログラムに関する質問や議論を BBS 型ページで行う場合、バージョン指定リンクで対象ページをリンクすることが重要である。こうすることによって、刻々と変化するソースコードに対しても、どのバージョンに対するの意見なのか明確になり、スムーズな議論を支援することはもちろん、後からの学習者が参照する場合にも非常に役に立つ。

3.5.1 プログラム定義

プログラムリストを表示させたいときは、メイン部のなかにそのまま埋め込む。この時、プログラムだということを定義するために、プログラムリストの先頭行に「{{{プログラムファイル名}}」と宣言する。そして、次の行からプログラムを並べ、最後に「}}」を挿入する。

また、文中からソースコードを抜き出して指定したソースファイルに抽出し、それにリンクを張っている。学習者はこのソースファイルをダウンロードすることでローカルにソースファイルを得る。この時、最新のソースコードは指定された名前前で保存される。

ページを編集して、ソースを書き換えるとページの

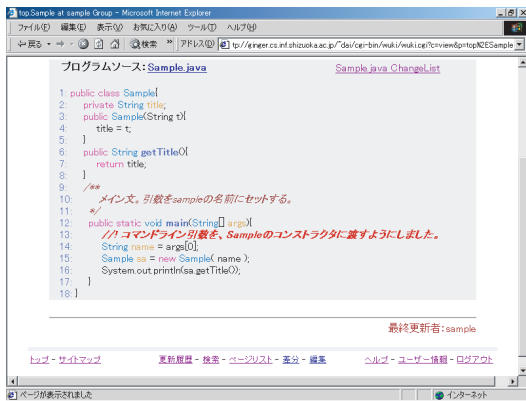


図 3 プログラム変更後

バージョンがあがり、同様に、ソースコードのバージョンもあがる。バージョン 10 の時のソースコードは、Sample.java.10 のように保存され、バージョン 10 の「サンプル」ページから、ソースコードリストとしてリンクされる。こうすることによって、学習者は常に最新のソースコードも入手できる。

3.5.2 変更コメント

Java 言語のインラインコメントは「//」で表すが、本システムでは「変更コメント」というものを「//!」で定義する。変更コメントとは、以前のバージョンからどこをどのように変更したのかという意味要素や、なぜ変更したのかという原因要素を記述するためのものである。差分機能を用いれば、どのような変更がなされたはすぐにわかるが、こうしたソースコードなどの変更の意味はなかなか理解しづらいため、この変更コメントを用いる。

変更を保存し、再び表示したものが図 3 である。変更コメントは通常のコメントよりも強調されて表示されている。また、プログラムリストの右上に、“ChangeList”というリンクが追加されている。これは、先ほどプログラム中に埋め込んだ変更コメントを抜き出し、変更点の一覧を表示するページへのリンクである。

ChangeList ページでは、対象プログラムの今までの変更履歴を一覧で表示する。ここでは、変更を加えたユーザー、バージョン、日時、変更行、コメントを表示する。学習者はこのページを見ることで、今、変更された箇所を見られるだけでなく、過去にどのような開発の推移があったのかわかる。また、ここから簡単に任意の版のソースコードにアクセスできる。

4. 評価と考察

4.1 実験方法

本システムを用いたプロジェクトベース学習の有効性を評価するために、実際に数人のグループで学習に取り組む実験を行った。実験方法を表 1 に示す。

表 1 実験方法

サーバー	Apache-1.3.22 on Linux 2.2.18-0v14.2
グループ数	2
1 グループ	4 人
学習者の前提	Java 言語の基礎は勉強済み
課題	GUI を用いて CSV ファイルの内容を表形式に表示する Java プログラム
クラス数	4
予想行数	約 200 行
期間	1 週間
環境	時間と場所は自由

実験では、プログラムソースの協調開発支援としての評価に重点をおいたので、クラスのインターフェースはすでに定義済みで行った。被験者にはそれぞれ担当のクラスを振り分け、担当者はそのクラスを積極的に開発するようにした。もちろん担当クラス以外の開発にもかかわるべきなのだが、責任を感じてもらうため、担当もつけることにした。

また、議論を活発にしたり、質問などに答えるためにアシスタントを両方のグループに一名参加させた。開発が進まない場合などは、アシスタントが催促したり、サポートして、担当者や他の学習者の参加を促す。このような設定で 1 週間、実際にプログラムを共同開発する実験学習を続けてもらった。

4.2 学習効果の評価

実験終了後に 3 つのカテゴリー（学習効果、基本機能、共同プログラミング支援機能）についてのアンケートを行った。アンケートの質問に対して次の選択肢で答えてもらった。1 から 5 の番号をそのまま評価の得点としてスコアリングし、このスコアの平均と、標準偏差を求めた。

1. 絶対にそうは思わない。
2. あまりそうは思わない。
3. そう思う。
4. 大体そう思う。
5. 非常にそう思う。

学習効果に関する評価結果を図 4 に示す。ここでは、本システムの学習効果を調べるためだけではなく、このようなプロジェクトベース学習支援システムについての意識調査も兼ねて行った。概ねこのような協調作業支援システムを協調学習に用いることは有用だとみられている。また、学習効果が十分あり、また学習に対する動機付けにも十分役立つと考えられる。協調学習だけで全ての学習はまかなえるかに対しては否定的な意見が目立った。

4.3 基本機能の評価

基本機能に関するアンケート結果を図 5 に示す。本システムの操作が複雑でないか、二通りの方法で聞いたが、本システムの操作は簡単であるということが肯定された。

編集方法など基本的な機能に関しては大体中央値より

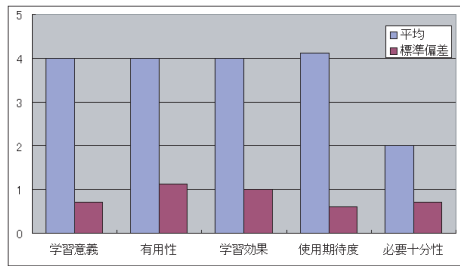


図 4 学習効果の評価

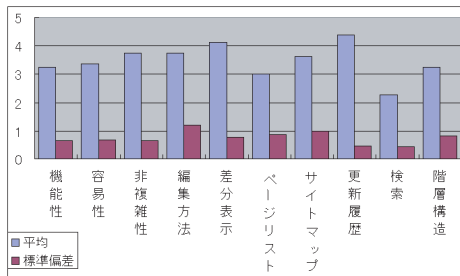


図 5 基本機能の評価

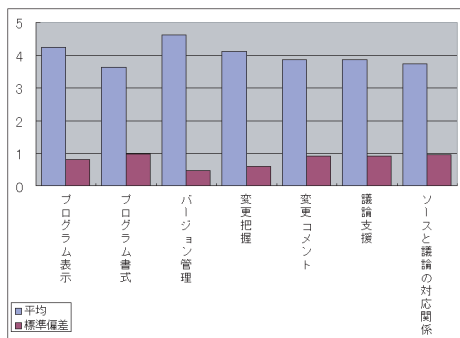


図 6 共同プログラミング支援機能の評価

も上回り、機能全般の平均得点は 3.5 とまずまずな結果となった。最も高得点だったのは更新履歴機能で、次に差分機能となっている。これは最も使用頻度が高いであろう機能が使いやすいということなので、システムとしては十分実用的であると思われる。

4.4 共同プログラミング支援機能の評価

プログラム開発支援機能に対する評価をグラフにしたものを図 6 に示す。非常に高い評価を得、このカテゴリの平均得点は 4.02 であった。しかし、プログラムの書式や変更コメントなど編集に関する項目にはばらつきがあり、それが若干スコアを下けている。また、議論支援や、ソースと議論の対応関係に関して、本システムで使用する BBS タイプでは十分とは言えず議論にはスレッド式の掲示板のほうが良いという意見もあり、そういう意見で評価に多少揺れが生じている。

4.5 学習効果との相関関係の考察

アンケート項目の学習効果のカテゴリと、共同プログラミング支援機能のカテゴリ、それとアクセス情報のそ

表 2 相関関係

項目 1	項目 2	相関係数
学習効果	必要十分	0.883
学習効果	ログイン数	0.647
学習効果	変更数	0.647
必要十分	変更数	0.728
プログラム書式	ログイン数	0.730
プログラム書式	変更コメント	0.900
変更コメント	ソースと議論の対応関係	0.801

れぞれの項目の相関係数を計算した。目立ったものを表 2 にまとめる。表 2 より、本システムに多くアクセスし、変更を加えた人が学習効果を感じていることが分る。さらに、学習効果があった、つまりより多く本システムを利用した場合は、本システムの機能は必要十分だと答える傾向にある。

5. おわりに

評価実験の結果、本システムを用いた協調作業によるプログラミング学習は有効であることが分かった。しかし、本システムのような協調学習システムでは、システムがどんなによい出来でも、学習者自身の参加意欲や積極性によって、学習効果が大きく左右されてしまうことが改めてわかった。学習者の学習への参加を促すために、アシスタントを用いたのは多少効果的ではあったが、それでもなおそういう問題のある学習者は存在する。今回行った数名の被験者の中にも存在するという事は、もっと規模が大きくなると、そういった学習者の数はさらに増えると考えられる。

現在、様々な協調学習支援システムが考えられているが、このような学習に取り残される学習者の扱いに重点をおいたものはあまりない。このような学習にあまり積極的でない学習者をいかにして減らすかが協調学習の今後の大きな課題である。

この研究の一部は平成 13 年度科学研究費補助金基盤研究 (B),13480051 および平成 14 年度科学研究費補助金基盤研究 (C),14580213 の援助を受けた。

文 献

- [1] 井上久祥, 陸田靖行, 岡本敏雄: “非同期型バーチャルセミナーの実現: 協調作業に基づいた非同期学習の構成,” 第 62 回情報処理学会全国大会論文集, 7B-02 (2001).
- [2] 川口清志, 竹田尚彦: “プログラミング教育における演習問題サーバの開発,” 信学技報, ET98-98, pp.41-48 (1998).
- [3] 宮地庸造, 横山節雄, 樋山淳雄: “グループプログラミング教育支援システムの開発,” 信学技報, ET98-103, pp.81-88 (1998).
- [4] Ward Cunningham: “PortlandPatternRepository and WikiWikiWeb front page,” Cunningham & Cunningham, Inc., <http://c2.com/cgi/wiki>
- [5] Mark Guzdial: “Supporting Learners as Users,” College of Computing Georgia Institute of Technology, <http://www.cc.gatech.edu/fac/mark.guzdial/learnerusers/>