

ソースコードに関する協調的な対話機能を有する 分散型ペアプログラミング演習システムの実現

青木 一浩[†] 立岩 佑一郎[‡] 山本 大介[‡] 高橋 直久[‡]

[†] [‡] 名古屋工業大学大学院 〒466-8555 愛知県名古屋市昭和区御器所町

E-mail: [†] aoki@moss.elcom.nitech.ac.jp, [‡] {tateiwa,yamamoto.daisuke,naohisa}@nitech.ac.jp

あらまし ペアプログラミングを演習で実施する場合、ソースコードに関するコミュニケーションの促進が重要である。また、各受講者が答案作成にどの程度寄与したか把握できるようにする必要がある。本稿では、以下の機能を持つ分散型ペアプログラミング演習システムを提案し、その実現法について述べる。(1) チャットと連携してソースコードやコミュニケーションの変化に対応する協調的なアノテーション機能。(2) 指導者が演習途中にアドバイザーとしてペアに加わり、ペアの演習状況を詳細に把握し、必要に応じてペアに助言可能なアドバイザー参加機能。また、提案システムのプロトタイプを開発し、学部の演習に適用した結果について述べる。そして、アノテーション機能が有用であることを明らかにした。

キーワード E-Learning, ペアプログラミング, アノテーション

Implementation of Distributed Pair Programming Exercise System with A Coordinated Communication Function on A Source Code

Kazuhiro AOKI[†] Yuichiro TATEIWA[‡], Daisuke YAMAMOTO[‡], and Naohisa TAKAHASHI[‡]

[†] [‡] Graduate School of Engineering, Nagoya Institute of Technology

Gokiso-cho, Showa-ku, Nagoya, Aichi, 466-8555 Japan

E-mail: [†] aoki@moss.elcom.nitech.ac.jp, [‡] {tateiwa,yamamoto.daisuke,naohisa}@nitech.ac.jp

Abstract A communication on a source code is essential in pair programming exercise. It is necessary for teachers to grasp each of students contributions. In this paper, we propose an implementation method of distributed pair programming exercise system which has the following features: (1) the function to adapt a coordinated annotation to changes of a source code and communications in cooperation with chat and (2) the function to make it possible to grasp each of students contributions and to help that teachers give advice to students in pair programming exercise. We have developed a prototype of the proposal system, and from the results of application to C programming exercise classes in our university, we have shown that the annotation function is useful in pair programming exercise.

Keyword E-Learning, pair programming, annotation

1. はじめに

近年、通信技術の発達に伴い、学校教育や企業教育などの場面で E-Learning システムが盛んに利用されるようになってきた。実際に我々は、初心者向けプログラミング演習を目的としたプログラミング演習支援システム CAPES[1][2][3]を開発し、運用をしている。CAPES とは、主に C 言語やアセンブラ言語 CASL の初級学習者を対象としたプログラミング演習支援システムであり、問題の提示や、答案評価の自動化を行うことで、演習を効率的に行い、個々のペースに合わせてプログラミング学習を支援する特徴を持つ。

しかし、CAPES を用いて演習を行う際、次のような問題が考えられる。

問題点 1 問題の途中で躓き、諦めてサボってしまう。

問題点 2 もともとやる気のない人は、問題を取り組

み始めない。

問題点 3 すでに演習を終了している受講者は時間を持って余す。

このような受講者を少なくし、さらにプログラミングの技術をより向上させるようにしたい。上記の解決法として、指導者が、演習室を巡回することで、サボっている受講者を注意したり、わからないところを教えたりするという解決法があるが、指導者が全員の様子を監視し、注意するのは負担がかかってしまう。

一方、XP(eXtreme Programming)のペアプログラミング[4]では、2人1組となって1つの端末を用いて1つのソースコードを作成する。ドライバ、ナビゲータの役割があり、ドライバは、キーボード、マウスを用いて、ソースコードの編集を行っていく。ナビゲータはソースコードを編集せずにドライバの編集作業の観察や、コメントをして、ドライバの相談や質問に答える。

このようなペアプログラミングを演習に適用すると以下の利点を得られると期待できる。

利点 1 共同作業を行うので、さぼりにくくなる。

利点 2 意見交換を通じて、プログラミングに関する理解が深まる。

これらの利点により、1人でプログラミングを行うよりもプログラミング技術の向上が期待できる。しかし、XPのペアプログラミングをそのままE-Learning形式のプログラミング演習に適用すると、以下の2つの問題が生じる。

問題点 1 ペアプログラミングで演習をする時、ソースコードに関するコミュニケーションを促進させることが重要となる。ソースコードに関するコミュニケーションを促進させるためには、ナビゲータがソースコードの間違いを容易に指摘でき、指摘箇所について活発に議論できるようにする仕組みが必要となる。

問題点 2 演習の採点や指導を効果的に行うためには、各受講者の理解度や演習に対する寄与の度合いを正確に把握し、必要に応じて適切な指導を行えるようにする必要がある。しかし、ペアプログラミングでは、ペアが共同作成して提出した答案プログラムを見るだけでは、その作成過程で、各受講者が記述したプログラムやコメントの箇所や量を把握できない。

本稿では、問題点1、2に対する仕組みとして、ソースコードに対するアノテーション機能を実現する。従来のアノテーションの対象は動画や画像[5]である。一般に、アノテーションの対象物が編集などによって変化することは想定されていない。一方、ペアプログラミングの演習では、ソースコードがアノテーションの対象であり、演習の進捗に従って変化する。

問題点1、2に対応するためには、ペアプログラミング演習でのアノテーション機能は次の3つの要件を満たす必要がある。

要件 1 答案作成過程でソースコードの内容が変化してもアノテーションとソースコードの対応付けが維持可能であること。

要件 2 演習状況を正確に蓄積することによって、誰がどの程度ペアプログラミングの演習に寄与したかを把握できること。

要件 3 演習指導者が演習状況を監視して、必要に応じてペアに対するアドバイスをアノテーションにより記述できること。

本稿では、上記要件を満たすアノテーション機能をCAPESに結合した分散型ペアプログラミング演習システムを提案し、その実現法について述べる[6]。

2. 関連研究

ペアプログラミング支援システム Reppsy[7]は、未踏ソフトウェア創造事業 2007 年度第 II 期末踏ユースに

採択されたプロジェクトである。このプロジェクトでは、ネットワークを介したペアプログラミングを行うための開発環境として提案されている。ペアプログラミング用の共同エディタにより複数のPCから1つのファイルを同時にリアルタイムに編集することができる。また、ユーザ管理機能、プロジェクト管理機能を有する。我々の提案システムは、演習に使用するものとして、受講者の現在の演習取り組み状況を考慮することにより役割交代をさせて、ペアプログラミングの効果によって、質の高い演習を目指す。さらに、ソースコード1行または、複数行に対するアノテーションを入れることができ、指導者が演習中の受講者のペアプログラミングに参加して、アノテーションやチャットを用いてアドバイスが可能となっている。

文献[8]では、開発履歴のコードの理解を支援するシステムを提案している。具体的には、分散ペアプログラミングを行っている際の会話などのコミュニケーションと生成されるソースコードを関連付けることにより、ユーザに対し開発履歴の変更理由の理解を支援する。提案するシステムは、プログラミングの開発履歴を保存する機能、プログラミング中に交わされた会話や共有されたメモなどを保存する機能、開発履歴と保存された会話やメモ情報を対応付けてユーザに提供する機能の3つの機能を有する。これによって、開発履歴の変更箇所をコミュニケーションログとの対応付けを変更日時と発言日時によって行い、プログラムの理解を支援している。我々の提案システムでは、ペアプログラミングでの演習中に行われる会話やコメントをソースコードと関連付けて、演習の質を向上させることを考えている。受講者間でやりとりされるコミュニケーションは、リアルタイムで演習ログに記録される。また、チャットやアノテーションとソースコードとの関連付けは、提案システムから容易に行うことができ、これを用いてペアプログラミングでの演習を支援する。また、ペアのやりとりした内容を演習ログから読み取り、それを再現することで、過去にペアでやりとりしたコミュニケーションで、ソースコードとの関連を持ったチャットやアノテーションの内容を確認することが可能である。

3. XPにおけるペアプログラミング

3.1. 実施方法

XPにおけるペアプログラミングでは、2人1組となって1つの端末を用いて1つのソースコードを作成する。まず、ドライバ、ナビゲータのどちらの役割になるかを決定する。ドライバは、キーボード、マウスを用いて、ソースコードの編集を行っていく。ドライバはソースコードを直接編集できるが、ナビゲータはソースコードを編集せずにドライバの編集作業を観察や、コ

メントをして、ドライバの相談や質問に答える。ドライバとナビゲータは役割を入れ替えながら、ソースコードの編集、コメント、観察を繰り返していく。

3.2. ペアプログラミングの利点

ペアプログラミングでは、次のような利点があるということが、ペアプログラミング—エンジニアとしての指南書[4]に記載されている。

品質 ペアの作成するコードのミスが少なくなる。

時間 1人でプログラミングするとき比べて、約半分の時間で高品質のコードを作成することができる。

士気 ペアプログラミングを楽しんでいる従業員は退職しないので、職場の定着率が上昇する。

信頼とチームワーク チームメイトについてよく知ることができ、信頼を築いてチームワークが向上する。

知識伝達 特にいつも同じ人とペアを組まないプログラマはシステム全体をよりよく理解する。

強化学習 相手を注意深く観察することにより、継続的に学習する。

4. 提案システム

4.1. 提案システムの特徴

提案システムを用いた演習では、2人の受講者が、それぞれの端末で、ドライバとナビゲータの役割を担い、ネットワークを介したペアプログラミングにより演習課題のプログラムを協力して作成する。提案システムの特徴は以下の3つである。

特徴1 ソースコードに対する協調的アノテーション機能

チャットと連携して、ペアが共有するソースコードやコミュニケーションの変化に対応する協調的なアノテーション機能を実現する。これにより、ソースコードが変化しても、ソースコードに対するチャットやアノテーションとの対応関係を維持し、受講者が容易にソースコードを参照しながらコミュニケーションできるようにする（要件1への対応）

特徴2 演習ログ解析機能

受講者の答案プログラム作成過程において、チャット、アノテーション、ソースコードの入力操作をログとして取得する。これにより、各受講者の演習に対する寄与の度合いを正確に把握できるようにする。

特徴3 アドバイザ参加機能

特徴2によって取得されたログを用いて、任意のペアの演習状況を再現する機能を実現する。指導者が、演習途中にアドバイザとしてペアに加わり、ペアの演習状況を詳細に把握し、必要に応じてペアにアドバイスできるようにするアドバイザ参加機能を実現する。これにより、指導者が、各受講者のプログラムの記述量、ペアの会話のやりとり、アノテーションの量や内容を詳細に把握して（要件2への対応）、チャットやア

ノテーションによりアドバイスできるようにする（要件3への対応）。

4.2. 提案システムの概要

提案システムは、図1に示すように、クライアントシステム、中継サーバ、制御サーバからなる。表1にシステムでやりとりするデータを示す。提案システムは、CAPESの答案評価機能により、答案の評価と評価結果を取得する。制御サーバがペアの組み合わせを制御し、ペアデータベースにペア登録情報を保持する。コミュニケーション機能は、受講者間で中継サーバを介してコミュニケーションデータを通信する。中継サーバで中継するコミュニケーションデータを受講者識別IDと共に演習ログデータベースに演習ログとして記録する。演習ログデータベースは、受講者の演習状況を再現する際利用される。

表1 システムでやり取りするデータ

コミュニケーションデータ	チャット、ソースコード、アノテーション
ペア登録情報	受講者識別ID+ペアID+名前+学籍番号+現在ペアが演習中かどうかの状態
演習ログ	受講者識別ID+コミュニケーションデータ

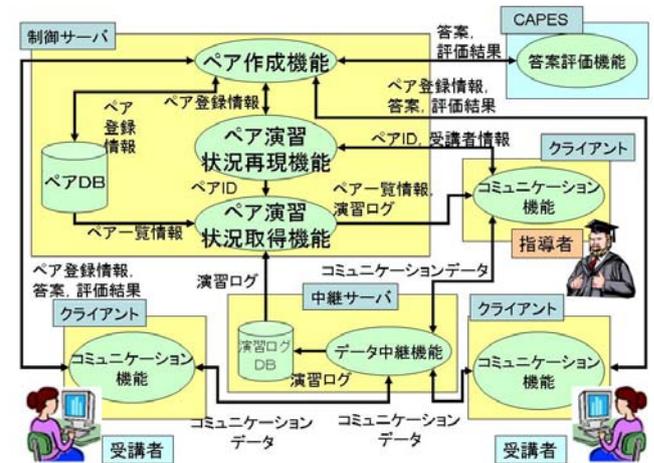


図1 提案システムの構成図

4.3. データベース

4.3.1. ペアデータベース

ペアデータベースは提案システムの制御サーバが保持する。以下に保持するデータを示す。

ペア登録情報 = {受講者識別ID+受講者の氏名+学籍番号+ペアID+中継サーバのIPアドレス+現在演習中かどうかの状態}

ペア演習状況 = {ペアID+単元ID+課題番号}

ペア登録情報の受講者識別IDは提案システムにログインしてきた受講者に対して一意に割り当てられる。ペアIDはどの受講者とどの受講者がペアとなっているのかをあらわすIDである。中継サーバのIPアドレ

スは、ペア間でやりとりするデータがどこを中継して、相手にデータを送信するか決める中継先の IP アドレスである。このペア登録情報はペア作成機能と中継サーバとの通信のために用いられる。ペア演習状況は指導者がペアの進捗度合いを確認するために用いられる。

4.3.2. 演習ログデータベース

演習ログデータベースは提案システムの中継サーバが保持する。以下に保持するデータを示す。

演習ログ = {受講者識別 ID + タイムスタンプ + データの種類 + コミュニケーションデータ}

データの種類には、受講者が入力したソースコード、アノテーション、チャット、役割情報がある。役割情報には、ドライバ、ナビゲータのどちらかになる。コミュニケーションデータには、中継したデータがそのまま入る。演習ログは指導者のためのペアの演習状況の把握や、演習状況の再現のために用いられる。

5. 提案システムの実現法

5.1. ソースコードに対する協調的アノテーション

ペアプログラミングにおいて効果的な演習を実現するためには、ソースコードを共有するだけではなく、ソースコードの任意の部分を参照可能な議論を促進させる必要がある。そこで、ドライバが編集したソースコードのリアルタイム共有機能、協調的アノテーション機能、参照チャット機能を提案する。ソースコード、チャット、アノテーションには図2のような参照関係を持つことで、ソースコードに関する協調的な対話機能を実現する。

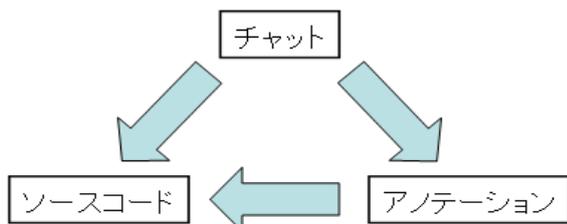


図2 3つのデータの参照関係

5.1.1. ソースコードリアルタイム共有機能

ソースコードの各行はソースコード ID によって管理する。ドライバがソースコードを更新すると、ソースコードの更新前後の差分を取得し、ソースコード ID とともに中継サーバに送信する。ペアとなっている相手は中継サーバから差分情報のみを受け取り、同じソースコードを再現する。

5.1.2. 協調的アノテーション機能

本機能では、ソースコードの任意の行に対応付けられたアノテーションを付与することが可能である。また、ソースコードが更新された際、アノテーションが保持するソースコード ID の位置変化を調べることで、対応関係の維持が可能である。本機能では、ソースコードの任意の1行に対するアノテーションだ

けではなく、複数行に対するアノテーションも可能である。ソースコードの更新前後の変化がアノテーションとの対応関係に影響を及ぼす場合、アノテーションからソースコードへの参照を更新する。対応するソースコードの行が消えると、ソースコード ID も消え、アノテーションとの対応がなくなったことを通知する。

5.1.3. 参照チャット機能

チャット機能を用いたコミュニケーションは単純であり、比較的慣れ親しんだコミュニケーション手法である。しかし、メッセージのみをやり取りするチャットでは、ソースコードのどの部分に言及したメッセージなのかの把握が困難である。

そこで、メッセージを入力する際、ソースコードの参照した部分を明示可能な参照チャット機能を提供する。これにより、チャットの入力メッセージに加えて、参照情報を送信可能にする。参照情報にはソースコード ID が含まれる。参照情報をメッセージと合わせて送信することで、ペアとなった相手は、チャットからソースコードへの関係を共有することができる。

5.2. 演習ログ解析機能

受講者の演習状況を把握するために、全ての受講者の演習での操作を記録する必要がある。また、指導者が、各受講者がどの程度ペアプログラミングでの演習に寄与しているのかを把握できるようにする必要がある。本機能では、中継サーバがコミュニケーション機能によってやりとりされる受講者の入力したソースコード、アノテーション、チャットを演習ログデータベースに受講者の識別 ID 共に記録することで、各受講者の操作取得を実現する。また、演習ログを用いて、各受講者のソースコード、チャット、アノテーションの量、答案提出回数を計算し演習ログ解析結果を出力する。演習ログ解析結果はアドバイザー参加機能において利用される。

5.3. アドバイザ参加機能

指導者がペアの演習途中に加わり、ペアの演習状況を把握するために、現在演習しているペアの把握と、ペアがやりとりした過去のコミュニケーションを取得して演習を再現する必要がある。本機能は、ペア作成機能、ペア演習状況取得機能、ペア演習状況再現機能により実現される。

5.3.1. ペア作成機能

STEP1 受講者は、受講者の名前、学籍番号を用いて提案システムにログインする。

STEP2 本機能は、現在ログインしている受講者がいるか検索し、ペアを作成する。

STEP3 ペアが作成できた場合、相手のペア登録情報を受信し、ペア作成を完了する。ペアが作成できなかった場合、本機能は再問い合わせし、STEP2に戻る。

STEP4 現在演習している状態を、ペアデータベース登録する。

5.3.2. ペア演習状況取得機能

STEP1 指導者は、指導者の名前、学籍番号を用いて提案システムにログインする

STEP2 本機能は、現在演習中のペア一覧、演習ログ解析機能による演習ログ解析結果を取得する。

STEP3 本機能は、STEP2で取得したデータをペアごと一覧で表示させる。

STEP4 現在のペアの演習状況は変化していくため、定期的に、STEP1-4を繰り返し演習状況を再取得する。

5.3.3. ペア演習状況再現機能

ペア演習状況再現機能は、ペア演習状況取得機能により得られる現在演習途中のペア一覧から、指導者が演習状況を再現したいペアを選択して、参加する。選択したペアのペアIDから演習ログを取得し、提案システムでペアの演習状況を再現する。

5.4. 提案システムを用いた演習の流れ

STEP1 受講者はCAPESから提案システムへログインする。

STEP2 提案システムは、現在ログインしている受講者を調べ、ペアとなる相手がいるか検索する。

STEP3 提案システムは、ペアを作成し、提案システムを用いた演習を開始する。ペアとなる相手が見つからない場合、STEP2を行う。

STEP4 受講者は、CAPESにより提示されている問題に対して、提案システム上で答案プログラムを作成する。

STEP5 答案プログラムができたなら、提案システムからCAPESに答案プログラムを提出する。

STEP6 受講者はCAPESから正解と判定されるまで、STEP4,STEP5を繰り返す。

6. プロトタイプシステムの実装

クライアントシステムはFlex[9]を用いて実装した。クライアントシステムをWebページで動作させるために、WebアプリケーションとしてApache Tomcat[10]を用いた。ペアの作成や、制御を行うための制御サーバも同様にApache Tomcatを用いた。クライアントシステム同士が通信をするための中継サーバは、Java言語[11]で実装した。ペアデータベース、演習ログデータベースは、MySQL[12]を用いて実現した。

提案システムのプロトタイプを図3に示す。図3の左上は、チャットインタフェースとなっており、チャットメッセージの送信や、プログラムやアノテーションと関連付けた参照チャットメッセージを送信できる。

図3の左下はプログラムの編集や共有を行うインタフェースになっている。ドライバのみが編集可能となっている。図3の右下にはプログラムと関連付けら

れたアノテーションが表示されるエリアとなっている。アノテーションは折り畳み表示が可能となっている。また、アノテーションに表示されている行番号は、プログラムが更新されると行番号も更新される。

図3の右上は受講者の答案プログラムの評価結果を表示するインタフェースとなっている。受講者が作成した答案プログラムをCAPESに送信することができる。また、CAPESが答案プログラムの評価を終えると結果がインタフェース上に表示される。

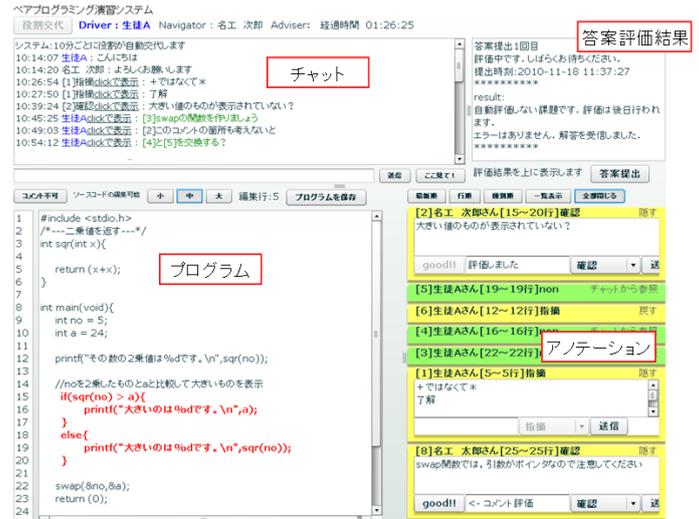


図3 プロトタイプシステムのインタフェース

6.1. アノテーション機能

アノテーションの入力例を図4に示す。

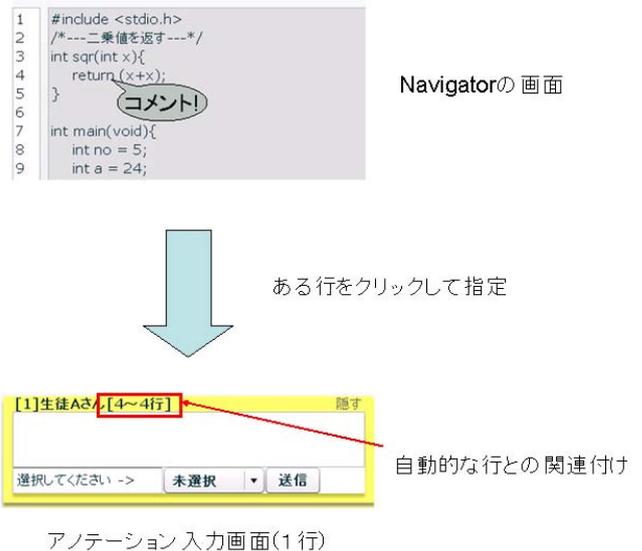


図4 アノテーションの入力例

ナビゲータの画面上で、プロトタイプのプログラム表示エリアにマウスカーソルを移動させると、カーソルがコメントとなり、コメントを入力することができる。コメントしたいプログラムをクリックすると、クリックされた行の行番号とコメントを自動的に関連付ける。アノテーション機能は任意の行に関連付けたコ

メントを入力することができ、プログラムの変化に対して、行番号が対応することが可能である。これによって、プログラムの任意の箇所に関連付けたコメントが容易にできる。また、プログラムとコメントの関連を容易に把握することが可能となる。

6.2. 参照チャット機能

図5に参照チャットの参照対象を示す。チャットを行うインタフェースから、ここ見てのボタンをクリックすることで、参照モードに切り替わり、ソースコードやアノテーションを参照することができるようになる。参照対象をクリックすると、図6に示すようにチャットの入力欄に参照対象とチャットをリンクさせるためのIDが自動的に入力される。受講者は、チャットの入力欄に続けてメッセージを入力し、相手に参照情報が付与されたチャットメッセージを送信することができるようになっている。

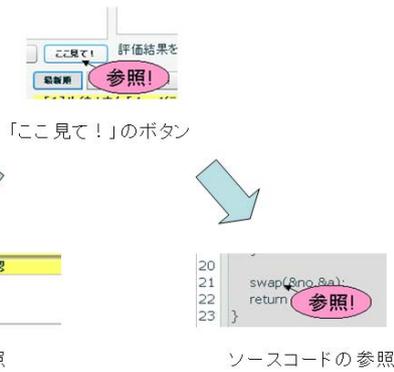


図5 参照チャットによる参照対象

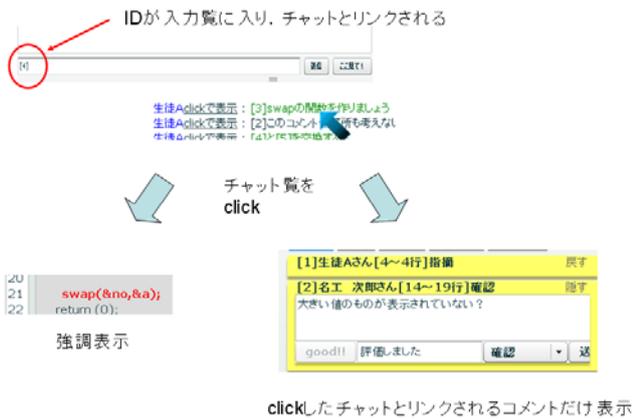


図6 参照チャットによる参照の実行例

図7に示すように参照機能は一度参照を行った後、ここ見てボタンをもう一度押すことで、二度目の参照をすることができる。図7の実行例では、ソースコードの2箇所の部分を参照することで、参照された2箇所の部分が強調表示されている。

参照チャット機能は一般的なチャットメッセージの送信に加えて、ソースコードやアノテーションと関

連付けたチャットメッセージの送信を行うことできる。また、1つのチャットメッセージと2箇所の関連付けが可能となっている。これにより、時間的な話の流れにおいて、ソースコードやアノテーションを参照しながらチャットメッセージが送信可能となる。

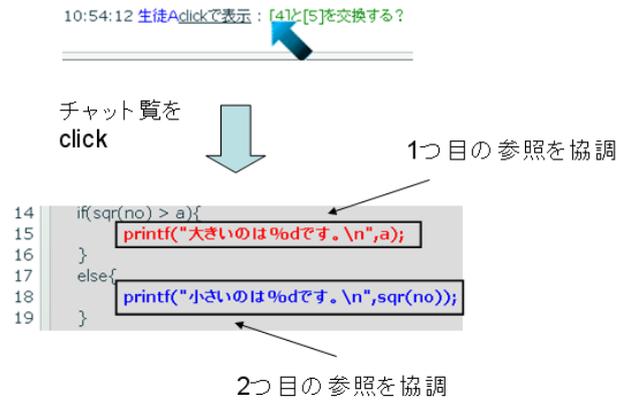


図7 参照チャットによる2箇所の参照

6.3. アドバイザ参加機能

図8にアドバイザ参加機能によって各受講者の演習活動状況の取得実行画面を示す。プロトタイプシステムは、大学の演習授業で、運用を行うため、受講者が取り組んでいる課題や、ソースコードの編集量、アノテーションの数、チャットの発言数は時間と共に変化していく。したがって、定期的に演習ログデータベースから演習状況の取得を行い、更新をする必要がある。プロトタイプシステムでは、図8の左上にあるように手動での更新（再読み込み）、指導者が設定した秒ごとの更新（オート再読み込み）が可能となっている。3回ログインしてペアができていない人を発見すると、図8の右上に表示されるようになっている。

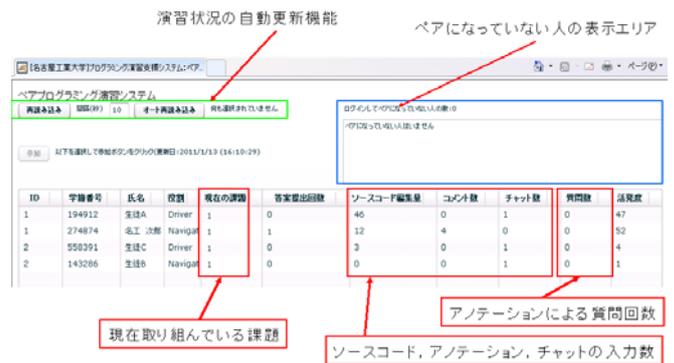


図8 アドバイザ参加機能による各受講者の演習状況一覧表示

指導者は演習中の受講者の一覧から、図9に示すように受講者を選択し、演習中のペアに参加できる。指導者が演習中のペアに参加すると、参加したペアに指導者が加わったことが通知され、指導者の端末では、図4のインタフェースが現れ、演習中状況を再現する。

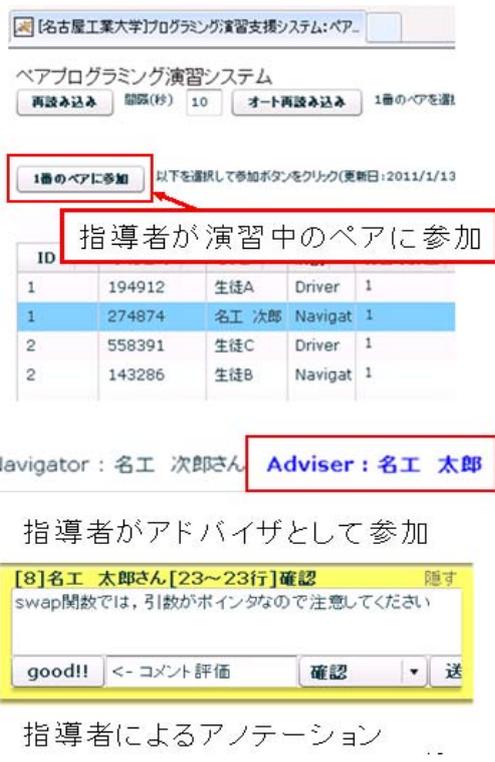


図 9 アドバイザ参加機能による指導者の演習への参加

アドバイザ参加機能は、全ての受講者がペアでの演習でどの程度活動しているのかを把握することができる。また、受講者の演習画面を指導者の端末で再現することができ、指導者は演習中のペアに対して、ナビゲータと同様にアノテーションや、チャットの入力が可能となる。これにより、指導者は見回りをせず、受講者の演習状況を把握した上でアドバイス可能となる。

7. 評価実験

過去に本稿で提案したシステムのプロトタイプシステムを開発し、2008年度より大学の演習授業で運用を行ってきた。今回の評価実験では、名古屋工業大学1部の2010年度プログラミング2の授業3クラス(1,m,n)において、CAPESによる個別演習(以下、従来手法)、1台のPCを2人で共有する従来型のペアプログラミングでの演習(以下、PP)、提案システムによる演習を行った。アノテーション機能の有効性を検証するために、次の3つの評価を実施した。

- (1) 提案システムに関するアンケート調査
- (2) 各クラス、各回での課題が正解するまでの平均時間の比較
- (3) プロトタイプシステムが収集したアノテーション、チャットのログ分類

(1) アンケート結果について

演習終了後、提案システムに関するアンケート調査を依頼し、147人の回答を得た。アンケート項目を以下に示す。また、図10に各アンケート項目に対して

5段階評価を行った結果を示す。

- 項目-1: 相手のアノテーションの内容がプログラム作成に役立ったか
- 項目-2: プログラムの間違いなどをアノテーションで指摘できたか。
- 項目-3: アノテーションによるプログラムの強調表示が役立ったか。
- 項目-4: 提案システムを用いた指導者からのアドバイスは役立ったか。
- 項目-5: 提案システムの使い方を理解できたか。

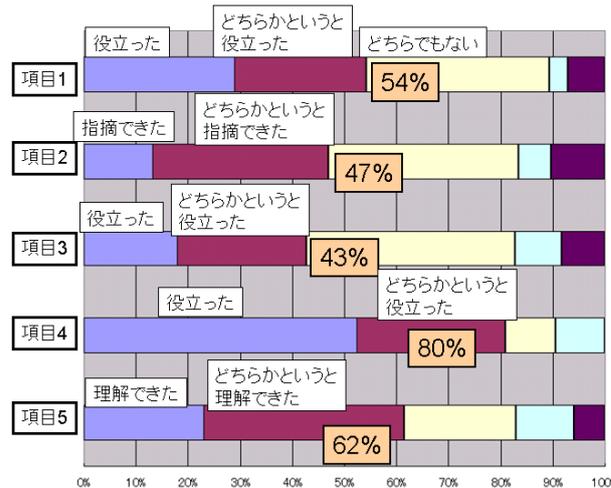


図 10 アンケート結果 1

また、アンケート項目と、5段階評価{1:あてはまらない, 2:どちらかというあてはまらない, 3:どちらともいえない, 4:どちらかというあてはまる, 5:あてはまる}の平均値をまとめたものを表2に示す。項目-5,6,7はPPによる演習後、項目-8,9は提案システムによる演習後に実施したアンケート項目である。

表 2 アンケート結果 2

アンケート項目	平均値
項目-6: ナビゲータの指摘によってプログラム作成を中断させられることがあったか	2.86
項目-7: ナビゲータがコメントで、プログラムを伝えることは難しかったか	3.38
項目-8: プログラムとリンクさせてコメントしたいことがあったか	3.17
項目-9: 相手のチャットやアノテーションによってプログラム作成を中断させられることがあったか	2.45
項目-10: チャット、プログラム、アノテーションとの関係を把握することがプログラムの理解に役立ったか	3.21

表2の項目-6,9を比較結果から、提案システムは、ドライバがプログラムを作成する時、作業を中断させられずに演習を進めることを支援していることがわかる。また、表2の項目-7,8,10より、PPによる演習では、ドライバに変更してほしいプログラムを伝えるのは難しく、プログラムとの関連を持った指摘を行いたかつ

たのに対して、提案システムは発言、指摘の内容がどのプログラムとかかわっているのかを把握させ、プログラムの理解を助けていることがわかった。

以上のアンケート結果から、アノテーションが回答作成のために有効に働いていることがわかった。また、ドライバは編集作業に集中することができ、相手とのコミュニケーションを行う際に何について議論をしているか、理解を助けていることがわかった。

(2)平均正解時間の比較について

事前調査として、3クラス(l,m,n)での正解のばらつきがあるかどうか調査を、CAPESが取得した演習履歴を用いて調査した所、平均正解時間は1クラスが一番短く、1クラス、nクラス、mクラスの順番となった。3クラスでの演習を行い、各クラスの平均正解時間をまとめたものを表3に示す。

表 3 正解平均時間の比較

	従来手法	PP	提案手法
第1回	1:42:03(n)	1:42:01(m)	1:06:04(l)
第2回	0:30:53(l)	0:46:21(n)	0:44:44(m)
第3回	2:02:19(m)	1:29:35(l)	1:59:14(n)

各回において、平均正解時間が事前調査と比較して大きく変化した所はない。提案手法を用いた演習と同等の演習が行うことができた。

(3)アノテーション、チャットのログ分類

プロトタイプシステムが取得した演習ログを用いて、アノテーションとチャット(アノテーション:204個,チャット:1688個)の内容を分類した。

表 4 アノテーション、チャットの分類

表4の分類結果に示すように、プログラムに関する内容(プログラムの質問,プログラムの説明,プログラムの間違い指摘,プログラムの記述)について比較すると、チャットは17.2%であるのに対して、アノテーションは51.4%と高い頻度で利用されている。従って、アノテーション機能はチャット機能に比べてプログラムと関連付けしたコメントを容易に行えるインタフェースであることを示唆する。逆に、挨拶などの雑談はチャット機能の割合が多い。用途に応じてチャット機能とアノテーション機能を使い分けることも重要であると考えられる。

内容	アノテーション (%)	チャット (%)
プログラムの質問	4.4	2.5
プログラムの説明	3.4	3.7
プログラムの間違い指摘	25.5	6.9
プログラムの記述	18.1	4.1
雑談	13.7	23.8
挨拶	2.9	5.8
意味のわからない文字列	11.3	15.4
その他	20.7	37.8

ラムの間違い指摘,プログラムの記述)について比較すると、チャットは17.2%であるのに対して、アノテーションは51.4%と高い頻度で利用されている。従って、アノテーション機能はチャット機能に比べてプログラムと関連付けしたコメントを容易に行えるインタフェースであることを示唆する。逆に、挨拶などの雑談はチャット機能の割合が多い。用途に応じてチャット機能とアノテーション機能を使い分けることも重要であると考えられる。

ト機能とアノテーション機能を使い分けることも重要であると考えられる。

8. おわりに

本研究では、分散型ペアプログラミング演習システムを提案した。また、提案した実現法をもとに、本論文であげた特徴を有するプロトタイプシステムの実装し、評価を行った。評価結果より、アノテーション機能の有効性と、チャット、プログラム、アノテーションの協調動作の利便性を明らかにした。

今後の課題として、どのようなペアの組み合わせが演習の質を向上させるか、ドライバ、ナビゲータの役割交代をどのような間隔で行うか、各受講者の演習に対する貢献の評価をどのように行うかが挙げられる。また、貢献の評価を利用して各ペアの状況の判定や、演習問題の出来具合の評価をすることで、指導者の指導効率向上と負担軽減が考えられる。

参考文献

- [1] 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, 2006-CE-83(18), pp127-134, 2006.
- [2] 中島秀樹, 高橋直久, プログラミング学習のための QA サイクル受講者の習得度に応じた問題自動提示メカニズム, 電子情報通信学会論文誌, VOL.J88-D-I, NO.2, 2005
- [3] 中島秀樹, 高橋直久, 細川宜秀, 受講者の状況に応じた個別学習機能を有する多言語対応プログラミング学習システムの実現, 電気情報通信学会第15回データ工学ワークショップ(DEWS2004), 2004
- [4] ローリー ウィリアムズ, ロバート ケスラー, ペアプログラミング—エンジニアとしての指南書 PEARSON Education Japan,2003年
- [5] Daisuke YAMAMOTO, Tomoki MASUDA, Shigeki OHIRA, Katashi NAGAO. [Video Scene Annotation Based on Web Social Activities](#), IEEE Multimedia, Vol.15, No.3, pp.22-32, 2008.9
- [6] 青木一浩, 立岩佑一郎, 山本大介, 高橋直久, 分散型ペアプログラミング演習システムの実現, 信学技報, vol. 109, no. 335, ET2009-85, pp. 193-198, 2009年12月.
- [7] 未踏ソフトウェア創造事業 2007年度第II期未踏コース プログラミング教育を変える! Remote Pair Programming System 「Reppsy」 <http://www.ipa.go.jp/jinzai/esp/2007youth/kaihatuseika2.html>
- [8] 西川 穂高,酒井 三四郎,分散ペアプログラミングにおけるコミュニケーションログとコード変更箇所の対応付けによる理解支援,情報科学技術フォーラム一般講演論文集 4(1), 103-104, 2005-08
- [9] オープンソースフレームワーク "Adobe Flex", <http://www.adobe.com/jp/products/flex/>
- [10] Apache Software Foundation - Apache Tomcat <http://tomcat.apache.org/index.html>
- [11] Java テクノロジ サン・マイクロシステムズ <http://jp.sun.com/java/>
- [12] MySQL 世界でもっとも普及している、オープンソース データベース <http://www-jp.mysql.com/>