

初学者向け協調プログラミング支援システムの開発

Development of a collaborative programming support system for novice

加藤 優哉*1, 酒井 三四郎*2, 松澤 芳昭*3
 Yuya KATO*1 Sanshiro SAKAI*2, Yoshiaki MATSUZAWA *3
 *1*2*3 静岡大学情報学部
 *1*2*3Faculty of Informatics, Shizuoka University
 Email: kato@sakailab.info

あらまし:本研究では,プログラミング入門教育の最終プロジェクトとして実施されるグループ課題(協調プログラミング)を支援するシステムの開発を行った. 開発されたシステム(CheCoPro)は,グループメンバーの 1)ソースコードを単純な操作での取り入れが可能である, 2)ファイルを直接編集するようなことが発生しない,という特徴を持つ. 入門教育の最終プロジェクト2グループによる使用実験を実施した結果,被験者は困難を感じることなく本システムを利用できることが確認された.

キーワード: 協調学習, プログラミング教育, 初学者

1. はじめに

プログラミング入門教育の場でグループを組んで互いに協力してプログラミングを行うこと(協調プログラミング)が行われている. 初学者は,協調プログラミングを行う際に困難を感じている. 構成管理ツールの利用ができないため,協調プログラミングの際にファイルの共有や役割分担に苦労していることが問題である.

そこで,本研究では初学者が直感的な操作で利用可能な協調プログラミング支援システムの開発を行った. 開発したシステム(CheCoPro)は,文系系大学生を対象としたプログラミング入門コースの最終プロジェクトで,2グループ(6名)を対象として使用実験を行った.

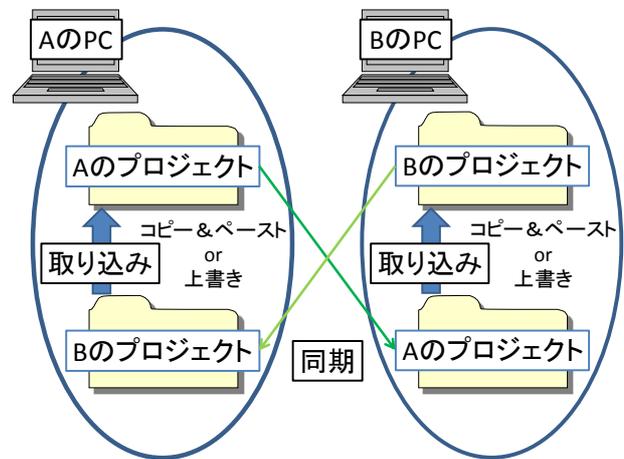


図 1: CheCoPro のモデル

2. 先行研究

黎らはグループでのプログラミング教育を支援するシステムを開発している⁽¹⁾. このシステムは,web上で電子教科書,仕様書,進捗状況,掲示板などを管理するシステムである. この支援システムで,システム設計の方法や仕様書の書き方などを自然に身につけることができると主張している. 問題点は,コーディング面が考慮されていない点である.

Eclipse上でソースコードの同期編集を可能とするSarosというプラグインをS.Salingerらが開発している⁽²⁾. エディタ上のテキストに加え,メンバーが開いているファイルやスクロールバーの位置などを示すことで意識共有を支援し,コードレビューや遠隔でのペア・パーティプログラミングの際に有用であると主張している. 問題点は,1つのプロジェクトをグループで共有するため,気軽にソースコードを編集できない点である.

3. 提案するシステム

本研究では,グループでプログラムを組むことに慣れておらず,バージョン管理システムを使いこな

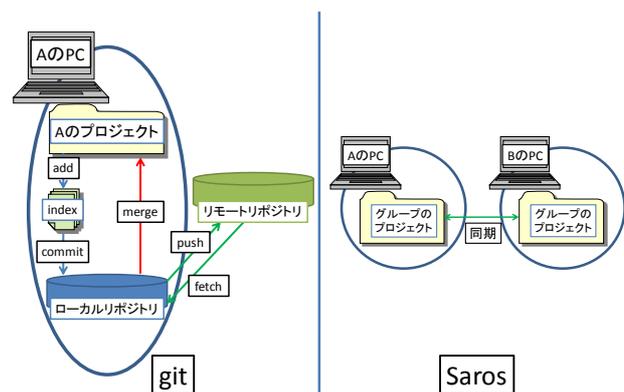


図 2: 既存システムのモデル

すことのできない学習者が利用可能なシステム「CheCoPro (Cheerfully Collaboration Programming)」を提案する. CheCoProのモデルを図1に,既存システムのモデルを図2に示す.

gitのモデルは,1)ローカルの変更をリモートに反映させるためにpushを行う必要がある,2)リモートの内容を取得するためにfetchを行う必要がある,3)mergeの際に競合が発生した場合に競合を解

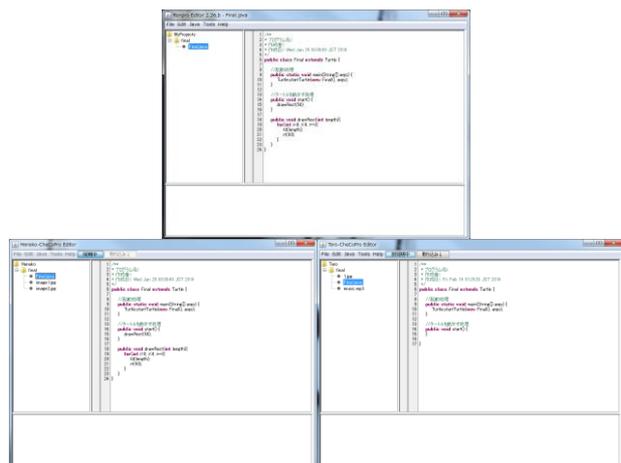


図 3：メンバーのソースコード閲覧の様子

決する必要がある、4) 1人で複数のバージョンの管理が可能である、という特徴を持つ。

Saros のモデルは、1) グループ全員で1つのバージョンを管理する、2) 同じファイルに複数人で同時に書き込み可能、3) 自分のウィンドウでメンバーが編集する様子を追従するモードが存在する、という特徴を持つ。

本モデルは、1) 1人1バージョンを管理する、2) ソースコードの取り込み方法はコピー&ペーストと上書き保存の2種類、3) メンバーの編集する様子を個別のウィンドウで確認しながら自分自身も別ウィンドウでコーディング可能、という特徴を持つ。

本モデルと既存システムのモデル (git, Saros) との違いを以下に示す。

- 取り込み方法
- 管理可能なバージョン数
- メンバーのファイルの書き換えの可否
- メンバーのソースコード編集の可視性

本システムのソースコードの取り込み方法はコピー&ペーストと上書き保存であるため、競合が発生することはない。管理するプロジェクトは自分のプロジェクトのみであり、自分以外のメンバーのファイルの書き換えができないことから単純な仕様である。図 3 のとおり、メンバーがソースコードを編集する様子を確認しながら自分自身もプログラムを組むことが可能である。以上のように、初学者が利用できるモデルを提案する。

4. 評価実験

4.1 実験方法

文科系大学生の1年生6名(3名2グループ)に、プログラミング入門コースで CheCoPro を利用し、1週間自由に1つの作品を制作するという課題を与えて観察した。その後アンケートを実施した。

4.2 実験結果

グループのソースコードや素材(画像・音声ファイル)の取り込みの流れを図 4 に示す。各グループのリーダーは A₁ と B₁ である。グループ A に関して、

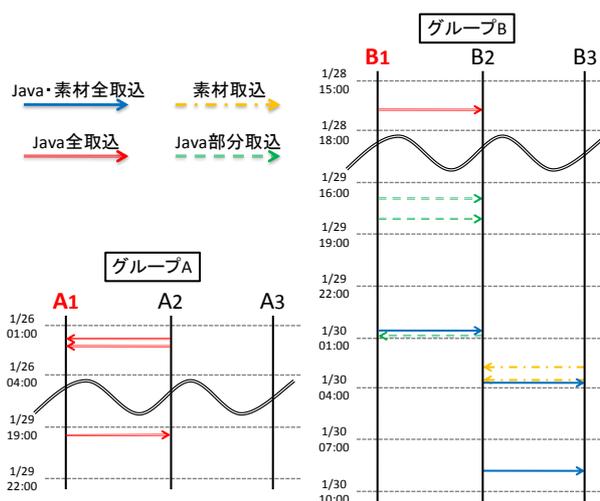


図 4：取り込みの流れ

A₁ が主体となってプログラムを組む様子が観察できた。グループ A は CheCoPro だけでなく、メッセージャーを用いてファイルを交換していた。A₂ のソースコードを A₁ は CheCoPro で2回取り入れている。このとき A₁ と A₂ は同じ内容のプロジェクトを持っていた。A₁ は新しいクラスを作成し、そこに A₂ のソースコードをコピーすることで、同じ内容のクラスを2つ所持した。予備のファイルを持つような利用方法が観察できた。

グループ B に関して、B₁ が主体となってプログラムを組む様子が観察できた。B₂ は B₁ のソースコードを取り入れ、取り入れたソースコードを基に変更を加え実行しているという操作を繰り返していた。B₃ はコーディング面では参加せず、素材を提供し、最終的に完成したプロジェクトを取り込み、テストを行っていた。

どちらのグループについても、システムの使い方が理解できないということはなく、特にグループ B に関しては取り込み方法が全パターン使われている。

5. おわりに

アンケートの結果はおおむね良好であり、被験者6名中5名に関して、ソースコードの共有・結合に苦勞を感じていなかった。単純な操作のみでソースコードの共有と取り込みが可能である CheCoPro のモデルの利点が、協調プログラミングを支援できていることがわかった。

参考文献

- (1) 黎翔寧, 宮寺庸造, 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”, 電子情報通信学会技術研究報告. ET, 教育工学, Vol.98, pp.81-88 (1998)
- (2) S. Salinger, C. Oezbek, K. Beecher, J. Schenk: “Saros: An Eclipse Plug-in for Distributed Party Programming”, Proceedings of the 2010 ICSE Workshop on Cooperative and Human Aspects of Software Engineering, pp.48-55 (2010)