

コード管理ツール Git を活用した 分散 PBL におけるチーム活動過程の可視化の試み

松原 克弥, 伊藤 恵, 木塚 あゆみ

公立ほこだて未来大学 システム情報科学部

matsu@fun.ac.jp

An Approach of Visualizing Team Activities in Distributed PBL with the Git Code Management Tool

Katsuya Matsubara, Kei Ito, Ayumi Kizuka

School of Systems Information Science, Future University Hakodate

概要

実践的な教育の取り組みとして、高等教育機関における PBL(Project-Based Learning) を用いた授業の導入が進んでいる。その有効性が認知されるにしたがって、近年、複数の機関が遠隔から参加する分散 PBL が実施されるようになった。少人数のチームを構成したグループ学習を行う PBL の学習評価では、学生個々の学びや最終成果物だけでなく、チーム活動におけるプロジェクトの遂行過程を評価することが重要となる。しかし、分散 PBL では、チーム構成メンバが地理的に分散していることにより、PBL 実施教員によるチーム活動の把握が困難となる。本稿では、開発支援ツールとして Git を用いた分散 PBL を実施し、その Git レポジトリの変化を可視化することにより、チーム活動過程を把握する試みについて述べる。

1 はじめに

大学等の高等教育機関における実践的な教育の取り組みとして、情報教育の分野を中心に、PBL(Project-Based Learning) を用いた授業の導入が進んでいる [1]。また、2002 年度に 15 の大学が連携して開始された実践的情報教育ネットワーク enPiT[3] では、PBL 主催大学の学生に加えて、異なる複数の機関の学生との混成チームを構成して、テレビ会議システム等を通して遠隔から参加する PBL(以降、分散 PBL) が実施されている。

新たな授業形式である PBL では、個々の学生の学びを評価するこれまでの学習評価に加えて、チーム毎のプロジェクトの最終成果物や成果発表、さらには、プロジェクトの運営や管理といったチーム活動過程を評価することが重要となる。しかし、分散 PBL では、チーム構成メンバが地理的に分散していることにより、PBL 実施教員がメンバ全員の活動状況を目視等で確認することが容易ではないため、チームの活動過程に対する評価が難しいという課題がある。

本研究では、ソフトウェア開発を主体とした分散 PBL における開発支援ツールの利用に注目し、ツールの利用履歴からチームの活動状況を把握することを試

みる。チームメンバへのアンケートやヒアリング、報告書や議事録等のドキュメントを介した状況把握と比べて、ツールの利用履歴を用いる本アプローチでは、説明能力やドキュメント記述能力の影響を受けないこと、数字という客観性のあるデータに基づいた評価が期待できる。解析対象とするツールは、近年、分散ソフトウェア開発を支援するツールとして導入が進んでいる、Git コード管理ツールを用いる。Git 利用履歴の可視化により、分散したメンバで活動しているチーム状況の把握を容易にすることを目指す。

以降、本稿では、チーム活動の把握に適した Git の特徴について説明する。その特徴を可視化するための方法を示した後、Git 利用履歴からのチーム活動を推定について論じる。さらに、実際の分散 PBL における試行とその結果について示し、その結果からチーム活動推測の試みについて述べる。

2 Git コード管理ツールの特徴

Git は、ソースコード・バージョン管理システム(以下、VCS)に分類されるツールである。Subversion のような、1つのレポジトリを共有する集中型 VCS と異なり、Git は、各開発者がソースコードのレポジトリを所有し、レポジトリへの変更内容のみをやりとり

するという分散型の VCS となっている [2]。各開発者が独立したレポジトリを用いて開発を進められることや、ソースコードの変更の責任所在とその履歴管理が容易であるという特徴から、多くのオープンソース開発プロジェクトで Git が利用されている。

Git では、ソースコードに対する変更を、コミットと呼ばれる作業でレポジトリに登録する。コミットで登録される情報には、ソースコードへの変更差分に加えて、コミットを行ったユーザの名前、コミット日時、コミット内容を示す文章が含まれる。各開発者に登録されたコミットは、プロジェクトの中心レポジトリへコミットを組み入れること (以降、マージ) により、分散開発されたコードがひとつのソフトウェアとして統合される。レポジトリに登録されたコミットは、その登録順に整列したリスト*1として管理される。一連のコミット列の途中から異なるコミット列を適用するバージョンを作ることができる。この分岐操作をブランチと呼ぶ。レポジトリ作成時に存在するブランチをマスターブランチと呼び、マスターブランチ以外のブランチは、開発中の一時的な用途として作成され、最終的にマスターブランチへマージする運用が一般的である。

現在では、クラウドベースで Git レポジトリを提供するサービスが登場しており、GitHub*2は、Git レポジトリのホスティングサービスとして広く採用されている。GitHub を用いた分散開発では、各開発者が主レポジトリを複製 (クローン) したレポジトリ上で開発を行い、その後、主レポジトリのマスターブランチに対する変更履歴の統合 (マージ) を要求する*3というプロセスが用いられることが多いこのプロセスに沿った分散開発では、主レポジトリのマスターブランチに統合されたコードが最終成果物となる。

3 Git 利用履歴を用いたチーム活動過程の推定

本研究では、分散 PBL におけるチーム活動過程を可視化する方法として、成果物を管理する Git レポジトリに対するコミット履歴に注目する。プロジェクトの開発過程で行われるコミットの数や頻度、コミットの作成者によって、チーム活動の進捗や役割分担などの状態を推測することを試みる。

*1 コードの変更は、スタックのように、登録の古いものから順番に上に重ねて適用される。

*2 <https://github.com/>

*3 プルリクエストと呼ぶ

3.1 Git 利用履歴の可視化

本試行では、Git レポジトリのコミット履歴の可視化のためのツールとして、GitStats*4を用いた。GitStats は、対象レポジトリのマスターブランチに登録されているファイルの数やライン数、コミット数、コミット日時や登録者 (以降、著者) の情報を集計し、図やグラフなどを用いて可視化する。集計結果は HTML ページとして出力するため、Web ブラウザにより容易に確認することができる。

3.2 チーム活動過程の推定

本節では、GitStats による可視化結果からチーム活動過程を推定する試みについて述べる。

3.2.1 開発進捗と情報共有に関するチーム活動の推定

チーム活動では、各メンバの状況の共有が重要となる。各メンバがレポジトリを所有する分散開発では、各メンバの開発成果を頻繁にマスターブランチへマージするほうが、他メンバの状況把握につながる。また、マスターブランチへのマージ頻度が低いと、マージの際にメンバ間の変更が衝突する可能性が高まり、マージのオーバーヘッドが増加する。

3.2.2 スケジュール管理に関するチーム活動の推定

PBL におけるチーム活動では、プロジェクト遂行能力として、スケジュールの管理が重要である。特に、開講期間の限られている PBL では、授業時間外での活動状況が成果に影響を与える。コミットが行われた時間帯を集計し、休日や深夜にコミット数が多いプロジェクトは、スケジュール管理に問題があるという評価を下すことができる。

3.2.3 開発分担に関するチーム活動の推定

PBL によるチーム開発では、全メンバへ役割を割り当て、学習機会を等しく与えることが重要となる。特定のメンバのみで開発の負担が偏っていないことは、PBL におけるチーム活動評価の重要な指標となりうる。コミットを著者別にカウントし、メンバ間で偏りがいないかを見ることで、適切な役割分担と平等な貢献度合いを評価することができる。

4 分散 PBL における試行

2016 年 8 月に実施された分散 PBL で、本可視化を試行した結果について述べる。対象とした分散 PBL は、地理的に分散した 4 つの大学の大学院学生が参加し、共通の課題に対してシステム提案とプロトタイプ実装を行う内容で、5 日間 (月曜日から金曜日) の集

*4 <http://gitstats.sourceforge.net/>

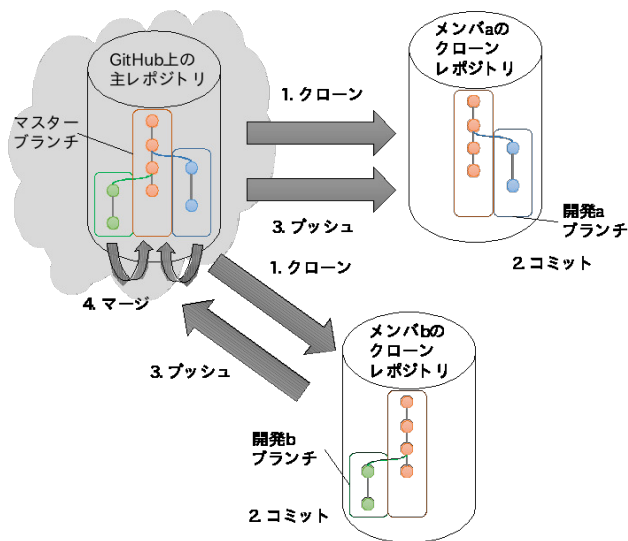


図1 実施した分散PBLにおけるGitレポジトリの運用プロセス

中形式で行われた。取り組む課題は、実在する保育園に対するヒアリングにより設定し、現在手作業で行っている卒園児向け情報発信のIT化を支援するシステムの開発とした。複数の大学の学生が必ず混成するようにチームを構成し、各チーム活動におけるコミュニケーションは、開発成果の管理と共有を行うためのGitレポジトリのクラウドサービスGitHub、文字によるチャットやファイル交換のためのコミュニケーションアプリSlack^{*5}、Polycomテレビ会議システムを用いた。

実装に際して、GitHub上にチーム毎に1つのレポジトリを作成し、各自のPC上のストレージへGitHubに作成したレポジトリのクローンを作成する(図1)。各メンバは、開発したプログラム等の成果物をPCのレポジトリに作成したブランチ上でコミットとして登録していき、適宜、GitHub上のレポジトリの対応ブランチへアップロード(プッシュ)する。プッシュされたコミットは、GitHub上でマスターブランチへマージすることで、他メンバのクローンレポジトリへの反映を行う。

本分散PBLでは、6つのチームを構成してそれぞれ開発を進めたが、以降、そのうちの特徴的な3つのチーム(Aチーム、Cチーム、Eチーム)について、可視化結果と前章で述べた方式に基づいて推定したチーム活動について述べる。

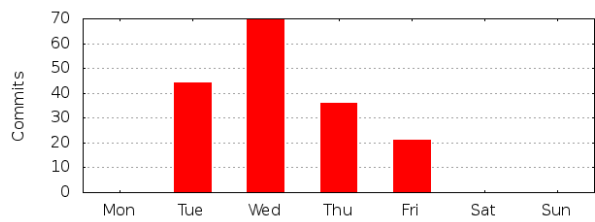


図2 Aチームのマスターブランチにおける日別コミット数

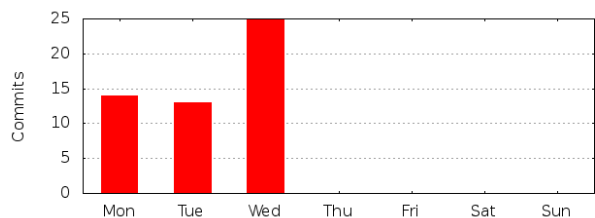


図3 Cチームのマスターブランチにおける日別コミット数

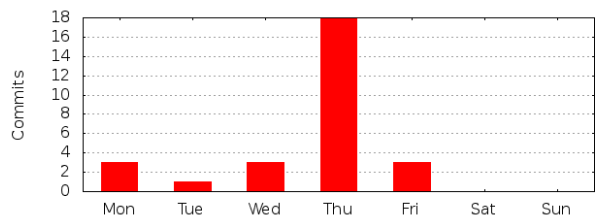


図4 Eチームのマスターブランチにおける日別コミット数

4.1 マスターブランチのコミット数推移

チーム別のマスターブランチのコミット数の推移を示した図2、図3、図4に示す。各グラフの横軸はPBLを実施した期間の各曜日、縦軸が各日におけるコミットの数である。

図2のチームAの結果では、初日にコミットはないものの、2日目から40を超えるコミット数があり、最終日の午前まで積極的なコミットが行われていることが読み取れる。チームAのマスターブランチにおける総コミット数は171で、全チームのなかで最も多い。前節の時間帯別のコミット数を可視化した結果からは、授業時間外のコミットがないことが判明したが、授業時間内でチーム別最多のコミットを登録していることから、効率的に開発が進められていたことが推測できる。

図3のチームCは、前節の結果からの推測どおり、初日からコミットがあり、3日目に最大数のコミットが行われていることから、3日目で開発を完了させようとしている意図が読み取れる。

*5 <https://slack.com>

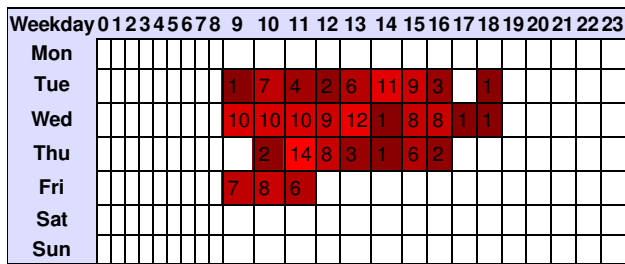


図5 Aチームの時間帯別コミット数

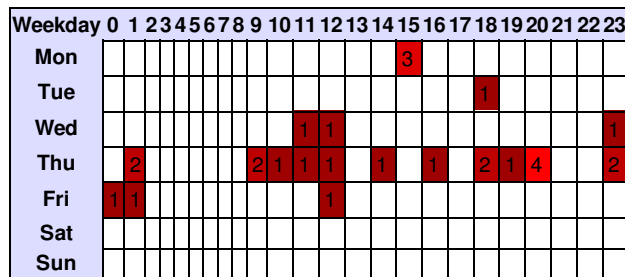


図7 Eチームの時間帯別コミット数

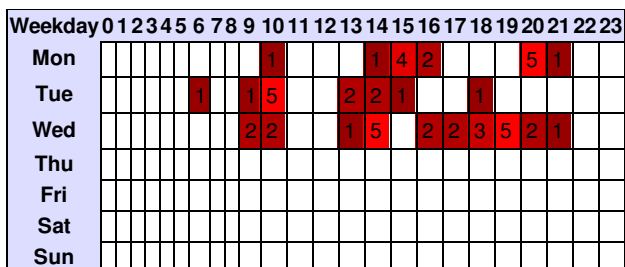


図6 Cチームの時間帯別コミット数

図4のチームEは、4日目が最もコミット数が多く、主に期間後半で開発を進めたことが推測できる。

4.2 時間帯別のコミット数

図5、図6、図7は、マスターブランチに登録されたコミットをその登録時間帯に沿って集計した結果である。各マトリックスにおける行はPBL実施期間の曜日、列がコミットを登録した時間帯を示している。コミットの曜日および時間帯は、各メンバのレポジトリに登録したときのもので、主レポジトリのマスターブランチへマージする際も変更されない。マス目の数字は、対応する行の曜日と列の時間帯に登録されたコミット数である。空欄のマス目は、コミットの登録がひとつもなかったことを示している。

図5の結果から、Aチームのコミット時間が2日目(火曜日)から最終日(金曜日)午前までに渡って平均的に分布していることがわかる。また、すべてのコミットが9時から18時の間に行われており、与えられた授業時間内のみの活動で計画的に開発を進めていることが推測できる。

それに対して、図6のCチームでは、初日から積極的なコミットを行い、4日目以降にコミットがないことから、3日目までで開発を完了させていることがわかる。

図7のEチームの結果からは、コミットが主に深夜に行われており、授業時間の範囲内では開発が完了していないことが読み取れる。

4.3 著者別コミット数の推移

図8、図9、図10は、マスターブランチに登録されたコミットをその著者別に集計し、その推移を可視化したグラフである。各グラフの横軸がPBL実施期間の各日、縦軸が累積コミット数を示している。各ラインがコミットの著者を示しており、GitStatsが出力する結果に含まれる著者名の記載を削除している。また、図8において、ラインの総数がチームメンバ数よりも多いが、これは、開発開始当初にgitの著者情報設定が誤っていた際に登録したコミットを指している。

図8のAチームでは、メンバ全員のコミット数が毎日平均的に増えており、均一な開発分担により、チームメンバ全員が開発に貢献できていることが読み取れる。また、コミット数が増加するポイントが毎日存在し、各メンバのコミットがマスターブランチに統合され、メンバ間での共有が行われていることが推測できる。

図9のCチームでは、期間前半のコミット数が特定のメンバに対してのみ増加しており、開発当初を主導しているメンバが存在していることが推測できる。期間後半は、他のメンバのコミットが増えていっているが、コミット数が増加しないメンバも存在することから、開発に関する分担に均一性がないことが示されている。

図10のEチームでは、特定のメンバだけがコミットを行っている状況がグラフから読み取れる。また、全くコミットを行っていないメンバもいることから、特定のメンバのみで開発成果を作り出していることがわかる。

5 おわりに

本研究では、分散PBLにおけるチーム活動状況の把握を目的として、Gitレポジトリに対するコミット履歴の可視化を提案した。Gitは、ソフトウェアの分散開発で広く用いられているコード管理ツールであり、その関連ツールも数多く存在している。本稿では、

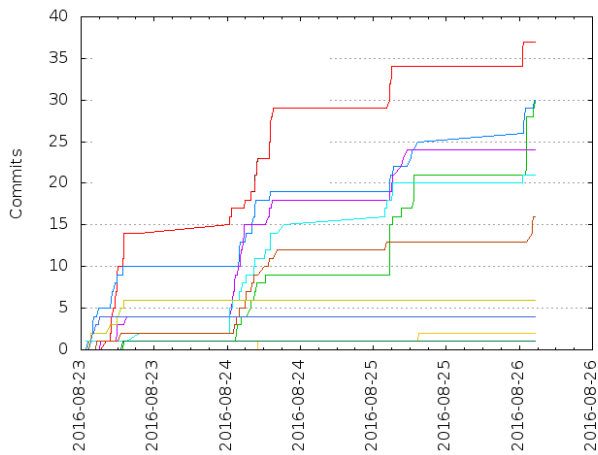


図8 Aチームの著者別コミット数の推移

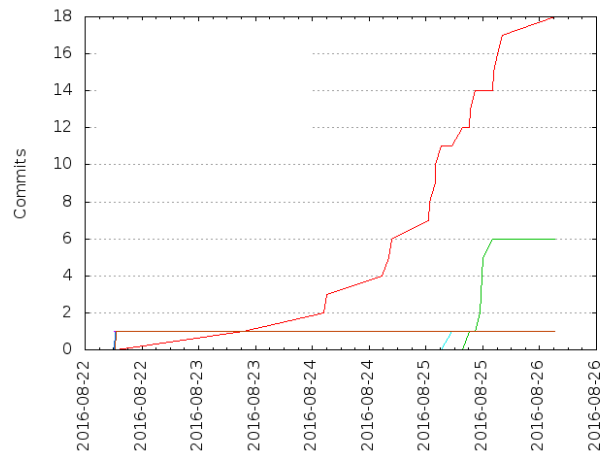


図10 Eチームの著者別コミット数の推移

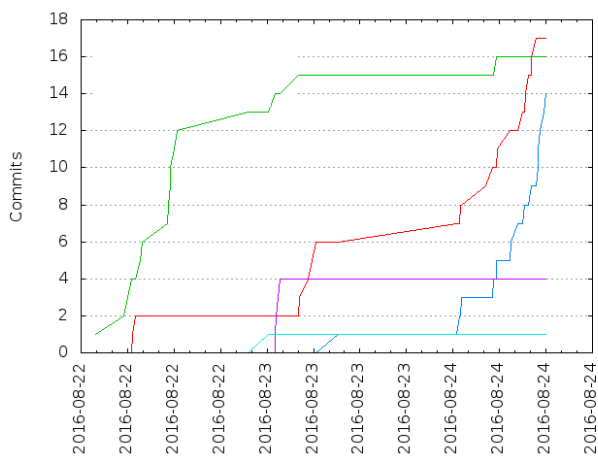


図9 Cチームの著者別コミット数の推移

GitStats というオープンソースの Git レポジトリ状況可視化ツールを用いて、コミットの登録日時や数、コミットを行ったメンバに関する統計情報を収集可視化した。その可視化結果から、PBL におけるチーム活動の状況や経過の把握を試みる方法について提案した。実際の分散 PBL において、Git レポジトリへのコミットを本アプローチを用いて可視化し、その特徴からチーム活動過程を推定する試みについて述べた。

今回の試行では、チーム構成メンバの Git 習得度の違いから発生する影響について考慮できていない。PBL を開始してから Git の使い方を学習したメンバが存在する可能性があることから、事前に講習会や自主学習を課すことで、Git 習得度の影響を小さくした状態で PBL を実施する必要がある。また、今回の試みでは、チーム活動過程が最終成果物へ与える影響について評価できていない。円滑なチーム活動がプロジェクトの最終成果に対してよい影響を与えることが期待されるが、本アプローチにより把握したチーム活動過

程と成果物との関係性について評価を行いたい。本提案が解析対象とした Git レポジトリのコミット履歴では、開発以外のチーム活動過程を把握することが難しい。今後、Slack や GitHub の issue 機能などの履歴を組み合わせ、設計や最終方向などの開発以外の活動についての評価方法の検討を進める予定である。

参考文献

- [1] 美馬のゆり. 大学における新しい学習観に基づいたプロジェクト学習のデザイン. 工学教育, Vol. 57, No. 1, pp. 45–50, 2009.
- [2] 伊藤恵, 木塚あゆみ, 奥野拓. 過去の PBL の開発履歴を活用した PBL 運用支援. 日本ソフトウェア科学会大会論文集, Vol. 31, pp. 249–260, 2014.
- [3] 文部科学省. 情報技術人材育成のための実践教育ネットワーク形成事業: 分野・地域を越えた実践的情報教育ネットワーク, 2012. <http://www.enpit.jp>.