

ソフトウェア開発における知識還元のためのプロジェクト再現ツール

大蔵君治† 後藤慶多† 川口真司† 花川典子‡ 飯田元†

ソフトウェア開発プロジェクトにおいて、開発者はしばしば同じ失敗を繰り返す。その理由の一つとして、開発者が過去の経験から知識をうまく獲得できていないことが挙げられる。そこで、我々は過去のプロジェクトを再現することで知識の獲得を支援するツール、プロジェクトリプレイヤの開発を行った。プロジェクトリプレイヤは過去のプロジェクトを動的な振る舞いも含めて画面上で再現するツールで、我々の提案する知識還元モデル Knowledge Feedback Cycle (KFC) を実現するためのツールの一つである。我々はプロジェクトリプレイヤの有効性を検証するために評価実験を行い、プロジェクトリプレイヤが開発者と研究者の両方にとって有用であること、また研究者がプロジェクトリプレイヤを用いてプロジェクトのより深い分析が可能であること確認した。

A Tool to Replay Software Project for Knowledge Feedback Cycle

Kimiharu Ohkura,† Keita Goto,† Shinji Kawaguchi,† Noriko Hanakawa,‡ and Hajimu Iida†

In a software development project, developers often repeat same mistakes. One of the reasons is that developers were not able to grasp knowledge very well from experience of past projects. We developed Project Replayer, which replays past projects in dynamic way to support developers to grasp knowledge from past projects. The Project Replayer is a tool for establishing Knowledge Feedback Cycle (KFC) we proposed. We confirmed that the Project Replayer is useful for both developer and researcher through an experiment. We also confirmed that researcher can analyze project more deeply using the Project Replayer.

1. はじめに

近年、社会におけるソフトウェア開発の需要増加に伴い、開発プロジェクトにおいて繰り返される失敗が深刻な問題となってきている。同じ失敗が繰り返される理由の一つとして、過去の経験から知識をうまく得ていないことが挙げられる。ソフトウェア開発において共有すべき知識を取り出すために、過去、多くの分析手法 [1][4] が提案されてきた。しかしながら、大量のデータから手動で有益な情報を見つけ出すことは大変困難である。更に、プロジェクトには開発に関するドキュメントや形式的なレポートからは知ることのできない暗黙的な背景事情が存在する。我々はそのような背景事情を不可視文脈 (Invisible Contexts) と呼ぶ。報告されない細かなトラブルや、形式張らない議論の存在が不可

視文脈の例として挙げられる。不可視文脈はプロジェクトの分析において重要な要素であるが、事後分析においてそのような文脈を発見することは難しい。

以上のような問題を解決すべく、我々は過去の経験から得た知識を開発者に効率よく還元するための概念として Knowledge Feedback Cycle (KFC) を提案し、KFC を構成するツールの一つであるプロジェクトリプレイヤ [2] を開発した。プロジェクトリプレイヤはソフトウェア開発において自動生成された様々な開発ログをもとに、ソフトウェア上で実際のプロジェクトを再現するツールである。更に、メールアーカイブを解析することによりプロジェクトの不可視文脈を明らかにする。本稿では主に、プロジェクトリプレイヤおよび KFC の有効性について詳説する。

2. Knowledge Feedback Cycle (KFC)

2.1. 目的

ソフトウェア開発の現場において、開発者はその開発業務を通して新しい知識を獲得する。その知識を将

†奈良先端科学技術大学院大学 情報科学研究科
Graduate School of Information Science, NAIST

‡阪南大学 経営情報学部

Faculty of Management Information, Hannan University

来のソフトウェア開発プロジェクトへ容易に伝えることができれば、過去の経験を有効に活用することができる。我々は、その手段として過去のプロジェクトから将来のプロジェクトへ知識を循環させる知識還元型開発環境、Knowledge Feedback Cycle (KFC)を提案した(図1参照)。本研究の目的はKFCを構築し、過去のプロジェクトから将来のプロジェクトへ有用な知識を循環させることである。

2.2. 概要

知識の循環を実現するために KFC は Empirical Project Monitor (EPM) [3], プロジェクトリプレイヤ, プロジェクトシミュレータと呼ばれる 3 つのツールを用いる。開発者はプロジェクトシミュレータを通して過去のプロジェクトに含まれる知識を得るために KFC を利用する。一方で、研究者は EPM によって収集されたデータやプロジェクトリプレイヤによって可視化されたデータを利用してプロジェクトの分析を行う。KFC が想定している標準的な作業の流れを次に述べる。

1. EPM がソフトウェア開発プロジェクトの様々なデータ (バージョン管理ツールやバグトラッキングツール, メール等のログデータ)を自動的に収集する。
2. 研究者が集められたデータを利用してプロジェクトの分析を行う。
3. 開発者がプロジェクトリプレイヤを通して過去プロジェクトのレビューを行なう。またその際、EPM で収集されなかったデータの追加を行なう。
4. 研究者は、1と3で集められた情報をもとに更に分析を行い、シミュレーションモデルを構築する。
5. 開発者は 4 で構築されたシミュレーションモデルを組み込んだプロジェクトシミュレータを利用して、過去プロジェクトに含まれる暗黙的知識を学習する。また、プロジェクトシミュレータを利用して開始前プロジェクトの計画支援を行う。このようにして計画されたプロジェクトは、更に次のサイクルでのステップ 1 適用対象となる。

以上のステップを繰り返すことで開発者へ有用な知識の還元が可能となる。

3. プロジェクトリプレイヤ

3.1. 概要

プロジェクトリプレイヤは EPM によって自動的に収集されたデータをもとに、ソフトウェア上で実際のプロジェクトを再現するツールであり、KFC における知識還元を支援することを目的に開発された(図2参照)。開発者

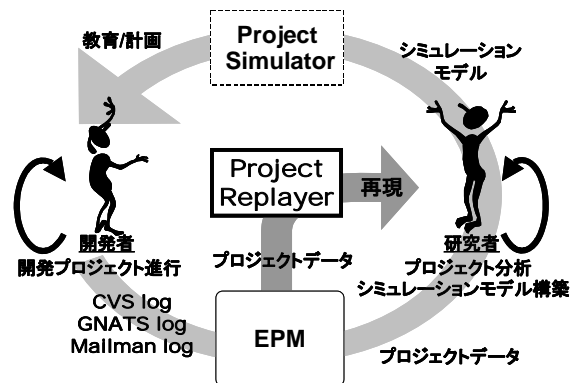


図1 Knowledge Feedback Cycle (KFC)

Fig.1 Knowledge Feedback Cycle (KFC)

はプロジェクトリプレイヤを用いて容易に自身の行なったプロジェクトのレビューを行い、一方で研究者はプロジェクトリプレイヤを用いて分析対象のプロジェクトをより詳細に理解することができる。

現在のバージョンでは4つの基本モジュールと1つの拡張モジュールから構成される。各モジュールの機能を、次節以降で述べる。

3.2. 機能概要

まず、基本モジュールとして組み込まれている4つのビュー、及び拡張モジュールとして組み込まれたメールビュー、及びタイムバーについてその機能を述べる(図3参照)。

イベントリストビューはEPMによって収集されたデータを時系列順に表形式で並べ、網羅的に表示する。ユーザは、イベントリストビューによってその日に発生した開発に関するイベントを一覧し、概要を掴むことが可能となる。

グラフビューは再現するプロジェクトのソースコード行数(LoC)の推移や累積ソースコード行数(CLoC)の推移をグラフ表示する。グラフビューを確認することで、プロジェクト全体の進捗を把握することができる。

メンバービューはプロジェクトに参加した全てのメンバーの担当作業と、その日の行動を表示する。メンバービューを確認することで、人的管理の観点からプロジェクトを認識することができる。

ファイルビューは、バージョン管理ツールで管理されている全てのファイルを表示する。各ファイルの項目はファイル名と進捗バーで構成される。進捗バーは当該ファイルの完成度(再現終了日を完成とする)を示す。ファイルビューを確認することでプロジェクトを成果物の観点から認識することができる。

メールビューは、メールアーカイブの解析結果を表示するプロジェクトリプレイヤの拡張モジュールである。プロジェクトリプレイヤは、分析表示の前処理としてメールアーカイブのクラスタリングを行い、アーカイブに含まれる種々の話題を自動分類する。メールビューは他のビューと連動してプロジェクトリプレイヤが現在再生している日時に存在する全ての話題を表示し、プロジェクトの不可視文脈を明らかにするための支援を行う。現在の実装では、メールビューはグラフィックビューに統合表示されている。

タイムバーはプロジェクトリプレイヤがプロジェクト中の再現すべき日付を示している。スライダを移動させることで特定の日時へ動的に移動して再現することができる。

4. 実験

4.1. 概要

我々は過去の実験によって、プロジェクトリプレイヤが実際のプロジェクトを再現でき、研究者がそのプロジェクトの流れを把握できること、シミュレーションモデル構築のための支援に役立つこと、また開発者が容易に過去プロジェクトを振り返ることができることを確認した[2]。本実験では、研究者がプロジェクトリプレイヤを用いて対象プロジェクトの不可視文脈、すなわち形式的な開発記録やレポートからは発見できない暗黙的な背景事情を見つけ出せるかどうかの検証を行う。実験対象は、大学院生によるオープンキャンパスのためのタイピングゲーム開発プロジェクトである。プロジェクトの詳細は表1の通りである。

4.2. 実験結果

実験を行った結果、研究者は実際のプロジェクトの振る舞いに加え、次に挙げる3つの不可視文脈を明らかにすることができた。

文脈1. 開発者同士のミーティングに関する情報
対象プロジェクトでは開発に関するミーティン

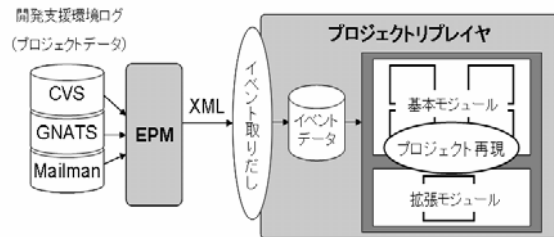


図2 プロジェクトリプレイヤの概要

Fig.2 Overview of Project Replayer



図3 プロジェクトリプレイヤの実行画面

Fig.3 Screenshot of Project Replayer

グが3回行われている。ミーティングはいずれも小規模なものであったため、記録は残されていないかった。

文脈2. 開発中断期間の存在

対象プロジェクトは5月27日から6月2日までの間、開発が中断されている。この期間はCVSコミットや新規バグ登録も1件も発生していない。これらの現象は、研究者に対して異常が発生していることを印象づけることとなった。しかし、中断の理由はわからなかった。

文脈3. 開発におけるメールの役割

表1 実験対象プロジェクト

Table.1 Target project of the experiments

オープンキャンパス タイピングゲーム開発プロジェクト	
メールアーカイブのサイズ	176KB (48 件)
開発期間	2005 年 5 月 21 日から 6 月 16 日 (26 日間)
コード行数の合計*	9,578 行
ファイル数*	105 個

* コード行数及びファイル数は EPM から取得したデータである

メールの大半は開発スケジュールの調整や、コード修正のアナウンス、リソースのアップロードといった種々の通知に用いられていた。詳細な打ち合わせの殆どは実際のミーティングで行われていたと研究者は予測できた。また、その予測の内容は正しいものであった。

これらの文脈は2時間程度の精査によって明らかになった。以上の結果から、研究者はプロジェクトリプレイヤを用いたプロジェクトの精査によって、そのプロジェクトの流れのみならず、背景に存在する様々な事象を効率よく発見できることを確認できた。しかし、文脈2において開発中断の理由を発見できないなど、クラスタリング精度に起因すると思われる問題も見受けられた。

5. 考察

我々は本実験、及び過去の実験結果から、プロジェクトリプレイヤの有効性を次の3点について確認することができた。

- 有効性 1.** 開発者が過去のプロジェクトを容易に振り返ることができる
- 有効性 2.** 研究者が対象プロジェクトの流れを容易に掴むことができる
- 有効性 3.** 研究者が対象プロジェクトの不可視文脈を明らかにすることができる

先述したように、我々は過去の経験を開発者に還元するためのモデルとしてKFCを提案した。有効性1によってKFCの利点の一つである、開発者自身による過去プロジェクトのレビュー効率化は実現されたと言える。また、有効性2はプロジェクトに参加していない研究者がプロジェクトの流れを掴むことができ、シミュレーションモデル作成の一助となりえること、また、本実験によって確認した有効性3により、原因のわからないグラフの変化やプロジェクトの文脈が明らかになり、モデルの検証を支援することも可能であると言える。しかしながら、分析精度はまだ十分とは言えず、この点についてはアルゴリズムの改良等によって精度の向上を行っていく必要がある。

6. まとめ

我々は、開発者への知識還元を実現するための環境としてKnowledge Feedback Cycle (KFC)を提案し、KFCを構成するツールの一つとしてプロジェクトリプレイヤを開発した。また、実験によって、研究者がプロジェクトリプレイヤを用いて対象プロジェクトの流れ、及びプロジェクトの不可視文脈を抽出可能であることを確認

した。しかしながら、この結果は研究者の精査能力に大きく依存しており、またメールビューの操作性やメールのクラスタリング精度も十分とは言えないものであった。これらの点については、今後ユーザビリティの向上、クラスタリングアルゴリズムの検討を行い改善していく必要があるだろう。

今後の展望として、我々はKFCの実現に向けてプロジェクトリプレイヤの洗練及びプロジェクトシミュレータの開発を行っていくと考えている。

謝辞

本研究において、分析に関する様々な助言をいただいたSRA 先端科学研究所の阪井誠様に深い感謝の意を表します。

本研究の一部は、文部科学省 科学研究費 基盤研究(C)17500024の補助を受けた。また、文部科学省「eSociety 基盤ソフトウェアの総合開発」の委託に基づいて行われた。

参考文献

- [1] Conklin, M. S., Howison, J. and Crowston, K.: Collaboration using OSSmole: a repository of FLOSS data and analyses" In Proceedings of MSR2005, St Louis, Missouri, (2005).
- [2] Goto, K., Hanakawa, N. and Iida, H. :Project Replayer An investigation tool to revisit processes of past project, In Proceedings of SPW/Prosim2006, pp. 72 -79 (2006).
- [3] <http://www.empirical.jp/>
- [4] Scacchi, W., Jensen, C., Noll, J. and Elliott, M.: Multi-modal modeling, analysis, and validation of open source software development processes, In Proceedings of the First International Conference on Open Source Systems, (2005).