

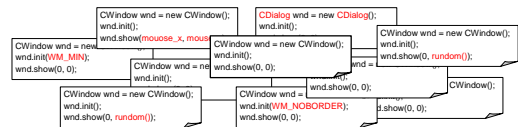
コードクローン履歴閲覧環境を用いたクローン評価の試み

川口真司* 松下誠**
井上克郎** 飯田元*

* 奈良先端科学技術大学院大学
** 大阪大学

コードクローンとは

- プログラム中に何度も類似した箇所が現れるコード片
- 主に、安易なコピー & ペーストによって生まれる
- 大規模ソフトウェアの保守において問題となっている
 - ある一箇所のクローンにバグがあった場合、すべての類似箇所について同じ修正が必要かどうか、検討が必要
 - 若干の修正が加わっていることが多く、網羅的に把握することが困難



クローン抽出システム

- 文字列を利用するもの
 - ソースコードの字句そのものから類似部を抽出
 - CCFinder* など
- 依存関係グラフを利用するもの
 - 各種単位間の依存関係グラフの形から
 - Baxter らの手法** など
- メトリクスを利用するもの
 - 関数、クラスなどの単位ごとにメトリクスを定義し、その値が類似したものを抽出
 - Balazinska らの手法*** など
- これらの手法により、大規模なソフトウェアからクローンを自動的に検出することが可能になった

* T. Kamiya, S. Kusumoto, and K. Inoue, CCFinder: A multilingual token-based code clone detection system for large scale source code. IEEE Transactions on Software Engineering, 28(7):654-670, Jul 2002.
** I. Baxter, A. Yahn, L. Moura, M. Anna, and L. Ber, Clone detection using abstract syntax trees. In Proc. International Conference on Software Maintenance, pp. 368-377, Mar 1998.
*** M. Balazinska, E. Merlo, M. Dagenais, B. Lague, and K. Korrogannis, Advanced clone analysis to support object-oriented system refactoring. In Proc. 7th Working Conf. on Reverse Engineering (WCORE 2000), pp. 98-107, Brisbane, Queensland, Australia, Nov 2000.

問題点

1. 検出されるクローンの数が膨大
 - 大規模ソフトウェアの場合に顕著
 - 全体的な傾向からの考察はできても、改善行動に結びつけるのが困難
2. すべてのクローンが一律に除去対象とは限らない
 - 削除すべきクローン
 - 抽象化の手間を惜しんでコピー & ペーストしたコード片
 - 役割分担の関係上、変更が困難なコード片
 - 削除すべきでない、削除しなくてもよいクローン
 - 将来の変更を見越して、意図的に重複箇所としている
 - デザインパターンやフレームワークを適用している箇所

削除すべきクローンとそうでないものとの判別が必要

クローン評価に向けて

- クローンの中身を解析
 - クローンを含む関数、クラスについて解析
 - メトリクスの算出
 - 依存関係の解析
 - 開発者の意図までを把握するのは困難
- クローンの発展過程、履歴を解析
 - クローンの過去を知ること、関係した開発者やその意図を推し量ることはできないだろうか
 - すぐ消されたクローンと、発生以来さまざまな部分にコピーされているクローンとでは明確な違いはあるのではないか
 - クローンを作成した開発者によって、違いがあるのではないか

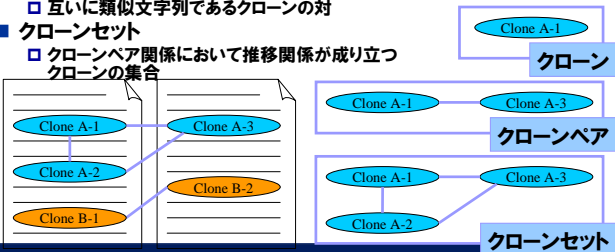
クローン履歴・関係者の解析

- クローンの発展過程、関係した開発者を解析する手法の提案
 - クローン履歴手法の紹介*
 - クローン関係者解析手法
 - 関係者: 作成者, 編集者, 削除者すべてを含んだ総称
- 得られた解析結果と、クローン評価との関連性に関する予備実験
 - 変更種別(作成, 編集, 削除)回数がクローン関係者ごとに異なるかどうかの検証
 - 単一コミットに含まれるクローンセット数と、クローン評価との関連の分析

* 川口真司, 松下誠, 井上克郎, 飯田元: 開発履歴システムを用いたコードクローン履歴分析手法の提案. 電子情報通信学会論文誌 D, Vol. J89-D, No. 10, pp. 2279-2287, 2006.

クローンに関する諸定義

- クローン
 - 類似文字列が存在するコード片
 - クローンの位置は (ファイル名, 開始行番号, 終了行番号) で表される
- クローンペア
 - 互いに類似文字列であるクローンの対
- クローンセット
 - クローンペア関係において推移関係が成り立つクローンの集合



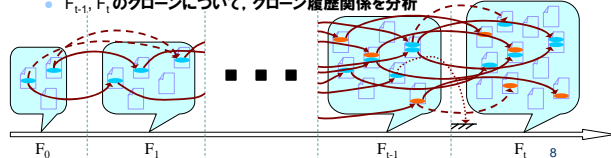
第154回SIGSE

Copyright © 2006 Nara Institute of Science and Technology



クローン履歴抽出手法

- 指定された期間 $[0, t]$, 間隔 Δt について期間 $[0, t]$ を Δt ごとに分割, それぞれの時間におけるファイル群を F_0, F_1, \dots, F_t と表す
過去のプログラムの取得には版管理システム (ex. cvs, subversion, ...) を用いる
- とおりあるバージョン間について分析
 - F_0 をリポジトリから取得, クローンを抽出
 - F_1 をリポジトリから取得, クローン抽出
 - F_0, F_1 のクローンについて, クローン履歴関係を分析
- ...
- F_t をリポジトリから取得, クローン抽出
- F_{t-1}, F_t のクローンについて, クローン履歴関係を分析



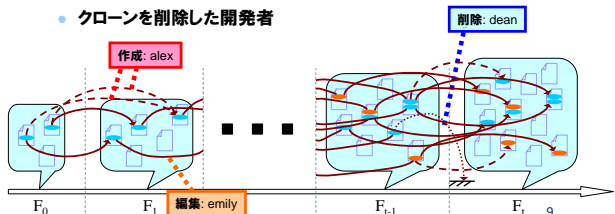
第154回SIGSE

Copyright © 2006 Nara Institute of Science and Technology



クローン関係者

- クローンの変更は何らかの形で関わった開発者
- 以下の3つの和集合
 - クローンを作成した開発者
 - クローンを編集した開発者
 - クローンを削除した開発者



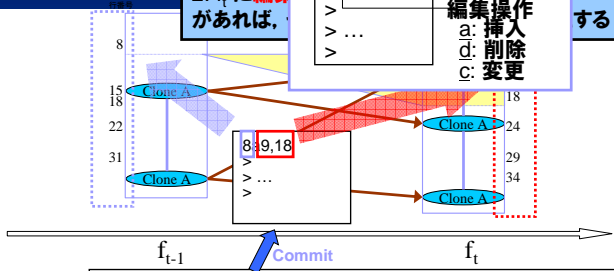
第154回SIGSE

Copyright © 2006 Nara Institute of Science and Technology



関係者抽出

- 版管理システム
 - f_{t-1} に編集元領域
 - f_t に編集先領域
 があれば、編集操作



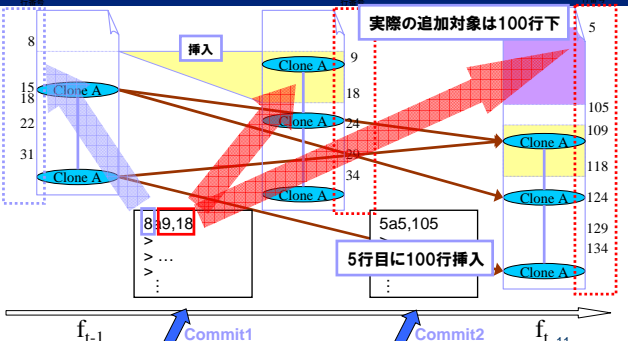
revision 1.82
date: 2003/07/20 21:56:32; author: tgl;
state: Exp; lines: +51 -29
Another round of error message editing, covering backend/commands/

第154回SIGSE

Copyright © 2006 Nara Institute of Science and Technology



複数回のコミットがあった場合

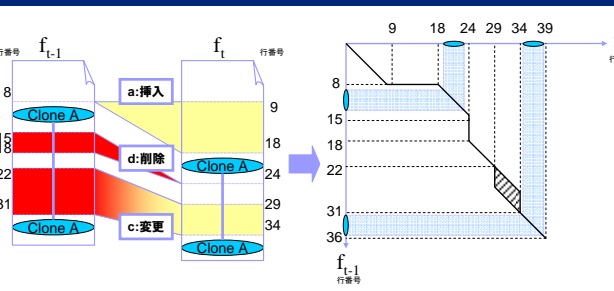


第154回SIGSE

Copyright © 2006 Nara Institute of Science and Technology



行番号の調整



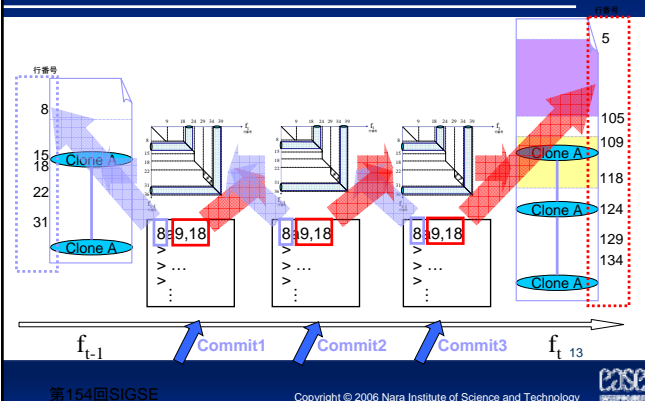
衝突時には対応行が一意でない → 最小、最大の推定値を考慮

第154回SIGSE

Copyright © 2006 Nara Institute of Science and Technology



複数のコミットを考慮した解析手順



クローン履歴の評価への応用

- 過去の履歴がクローン評価に利用可能かどうかを以下の2つの視点から検証する
 - 開発者によって、クローンとの関わり方に差があるかどうか
 - ある一度のコミット時に編集したクローン数と、そこに含まれるクローンの品質との関連
- 分析対象
 - PostgreSQL のサブモジュール (src/backend/commands 以下のソースコード)
 - 2003/01/01 ~ 2004/01/01 までを30日間隔で解析

14

第154回SIGSE

Copyright © 2006 Nara Institute of Science and Technology



実験方法

- 実験1: 開発者とクローン変更回数
 - 開発者ごとのクローン変更回数を集計
 - ある時刻に、1つのクローンセットを変更していたら1回。
 - ある時刻に、クローンセットを5個変更していたら、5回とカウント。
 - 開発者によって、変更作業(作成、編集、削除)に差があるかどうかを分析
 - クローンを追加してばかりの開発者がいないか
 - 逆に、クローンの削除を多く行っている開発者はいないか
- 実験2: コミットに含まれるクローンセット数
 - 同時に単一の開発者によって行われたコミットについて、そのとき編集したクローンセット数の分布を集計 (本実験では同時 = 1分以内とする)
 - 含まれるクローンの傾向について分析
 - 少数のクローンセットのみを編集しているコミット
 - 大量のクローンセットを一度に編集しているコミット

15

第154回SIGSE

Copyright © 2006 Nara Institute of Science and Technology



実験結果1 – 開発者ごとの特性

- 対象サブモジュールに関連した開発者ごとの特性
- 解析期間中にクローンの変更に関わった開発者ごとの特性

全コミット数に対して、クローン追加が多い

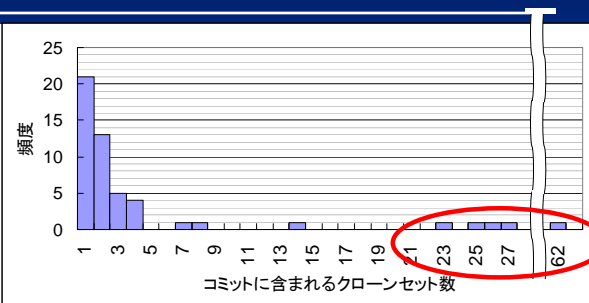
	momjian	petere	tgl
クローン追加	6	10	32
クローン削除	2	3	17
クローン編集	35	27	135
(小計)	43	40	184
総コミット数	1317	143	1428

16

第154回SIGSE Copyright © 2006 Nara Institute of Science and Technology



実験結果2 – コミットから見た特性



- 大多数のコミットでは、そのとき編集されたクローンセットは一つ
- ただし、いくつかのコミットでは多数のクローンセットを一齐に編集している

17

第154回SIGSE

Copyright © 2006 Nara Institute of Science and Technology



大量のクローンを編集したコミットの例

```
< eelog(WARNING, "DefineAggregate: attribute %s not recognized",
< defel->defname);
...
> ereport(WARNING,
> (errcode(ERRCODE_SYNTAX_ERROR),
> errmsg("aggregate attribute %s not recognized",
> defel->defname)));
```

"Another round of error message editing, covering backend/commands/."

- 機械的な変換作業
 - エラーメッセージ出力部の一括変換
- クローンの発見が困難でなかったと考えられる
 - grep 等を利用した文字列検索可能なクローン
- このようなクローンは対処優先度が低い
 - 機械的に処理をすべき対象を同定可能
 - 保守工程において障害とならない

18

第154回SIGSE

Copyright © 2006 Nara Institute of Science and Technology



実験結果の考察

- 開発者に着目した分析
 - 全体のコミット回数と比較して、クローン作成の多い開発者が見つかった
 - 彼が作成したクローンのいくつかは他の開発者によって削除されていた
 - 開発者によってクローンへのかかわり方は異なってくる
- コミットに着目した分析
 - コミット時に編集したクローンセット数の寡多から、着目しなくてもよいクローンセットが見つかった
 - 膨大なクローン情報のスクリーニングに有用

19

まとめと今後の課題

- クローン履歴分析手法の紹介, およびクローン関係者抽出手法の提案
- クローン評価のためのクローン履歴, 関係者の分析
- 今後の課題
 - 分析手法について
 - 適正に開発者が漏れなく取得できているかどうかの評価
 - ブランチを考慮した分析
 - 分析作業について
 - 広範なデータ分析
 - PostgreSQL の他モジュールも含めた解析
 - 他のソフトウェアを対象とした解析
 - 実験結果のより精細な検証
 - その他, クローン評価に有用な仮説立案・検証

本研究の一部は、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

20