

Call for Members for AY2023

小林研の研究内容の紹介

難しいこと面倒なことを
自動化することで
「ソフトウェア開発を
楽しみたい人」
を募集します



東京工業大学 小林研
Software Analysis Laboratory

1

小林研の研究領域

ソフトウェア工学

- ソフトウェア保守・進化支援 [2007~]
 - 「既にある(難解な)ソフトウェアの変更・修正」
 - プログラム理解支援, デバッグ技法, トレーサビリティ管理
- ソフトウェア開発支援環境 [2007~]
 - 「面倒なこと・難しいことの自動化」
 - プログラム解析ツール, デバッガ, DevOps, Infra. as Code
- ソフトウェア分析・設計技法 [1997~]
 - 「コードより抽象度の高い表現とは?」
 - 設計方法論, アーキテクチャ, デザインパターン

過去のデータに基づく帰納的アプローチ

- リポジトリマイニング, 機械学習に基づく手法

東京工業大学 小林研
Software Analysis Laboratory

2

小林研の大きな研究目標

ソフトウェア開発における属人性の軽減

- 「できる人」を増やす
- 熟練者の知識・経験を可能な限り「形式知」へ



東京工業大学 小林研
Software Analysis Laboratory

5

なぜ「知識や経験」に着目するのか?

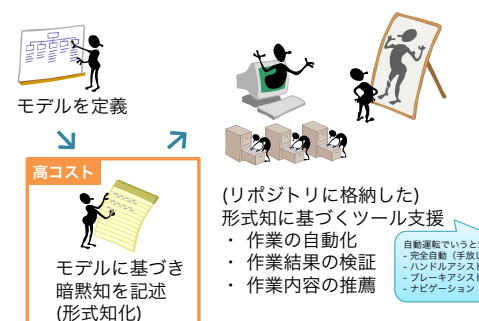
- ソフトウェア開発は高度な知的創造活動
 - 実現している機能 ⇄ ソースコードの変換作業は現在でも説明困難な活動
 - 開発者の理解・行動プロセスの補助が必要
- 2方向の支援アプローチ
 - 演繹的: プログラムの意味を解析し補助
 - 部分的なプログラムの意味を用いて理解・解析を支援
 - 命令単位での意味 → 関数単位での意味 → ...
 - 支援の確度は高いが, 適用範囲が限定的になりがち
 - 帰納的: 事例(プログラム・活動)を解析し補助
 - 多数の事例から意味/意図を導出: ML based AI的アプローチ
 - 適用範囲が広いが, 支援の確度は事例数と共通性に依存

こちらを採用

東京工業大学 小林研
Software Analysis Laboratory

6

研究アプローチ (基本的な考え方)



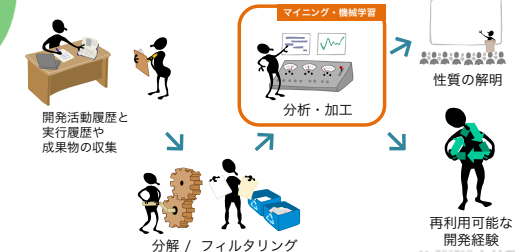
東京工業大学 小林研
Software Analysis Laboratory

7

研究アプローチ (帰納的アプローチ)

リポジトリマイニング・ML4SE

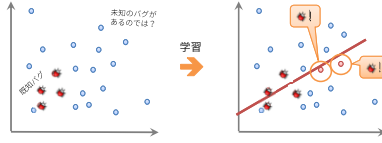
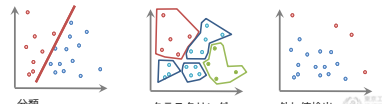
- 過去の成果物・履歴から再利用可能な経験を自動的に取り出し活用



東京工業大学 小林研
Software Analysis Laboratory

8

既存データからの経験抽出

- マイニング・機械学習・統計処理を応用
 
- 様々な技法を用いて経験を抽出
 

9

小林研が特に力を入れている領域

- ソフトウェア保守・進化
 - 保守＝既存のソフトのデバッグ・改修
 - バグの箇所を特定し、取り除くための支援
 - 進化＝既存のソフトへの機能追加・改善
 - 明示的・潜在的なバグを生み出さない変更を支援
 - 有益な情報のドキュメント化、情報提示



バグ防止



変更漏れ防止
開発者のバグ防止を支援し、バグ防止のためのドキュメント化、情報提示



デバッグ支援



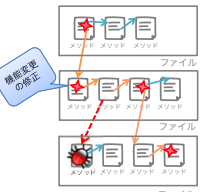
Just-in-Time 変更支援
開発者のバグ防止を支援し、バグ防止のためのドキュメント化、情報提示

11

バグ防止：変更漏れ発見

バグが混入する1つの要因 = 変更漏れ

- プログラムは複雑な依存関係をもつ
 - 変更波及：変更は様々な場所に伝播する
 - 変更波及を扱う手法＝波及解析 (Impact Analysis)



- 静的解析で波及先の特定は(ある程度)可能
 - コードが必須＝フレームワークなどを介すると分断される
 - ポインタ・動的束縛は「可能性」
- 全ての静的依存に波及するわけではない
 - 不要な関係が多いという報告

12

バグ防止：変更漏れ発見

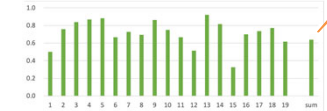
「変更漏れバグ」の現状分析 [修論2020]

- 実際に「変更漏れ」はどの程度・どんな規模で起こる？
 - 19のOSSから計2840件のバグ管理情報とリポジトリを調査
 - バグチケットから[Bug混入の編集, その修正]ペアを抽出

!! 平均 **44.7%** のバグ原因に変更漏れが含まれる

!! 90.7%で5ファイル以下, **53.2%は1ファイル**

- 改版履歴の分析でどの程度発見できる？ **63.9%**



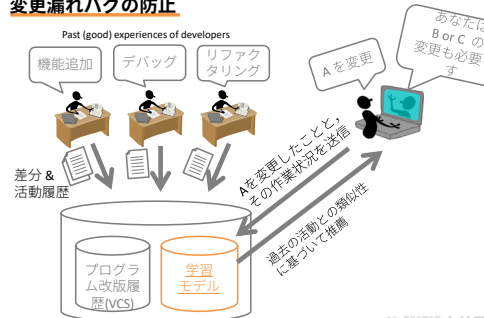
13

バグ防止：変更漏れ発見

活動の分析による経験(マイクロプロセス)の再利用

変更漏れバグの防止

Past (good) experiences of developers



14

バグ防止：変更漏れ発見

変更支援手法：ツールの出力例

Report for the commit 763a9fba93b91eb730e563eadb08de0444e80980

Changed files in the commit:

- java/org/apache/catalina/ha/session/LocalStrings.properties
- java/org/apache/catalina/core/LocalStrings.properties
- java/org/apache/tomcat/util/descriptor/web/LocalStrings.properties

[1] Top 1 candidate of overlooked change is:

because it was 66 % changed when the following your modified files were changed.

- java/org/apache/catalina/ha/session/LocalStrings.properties
- java/org/apache/catalina/core/LocalStrings.properties

It happened 2 times in past. For instances:

```

+-----+
|CommitID  : 84ead17|
|AuthorDate: Thu Mar 30 18:06:46 JST 2017|
|Author    : Mark Thomas<mark@apache.org>|
|Message   : Replace the use of (...) to delimit values in messages with...|
+-----+
|CommitID  : 8d46012|
|AuthorDate: Thu Mar 30 06:34:18 JST 2017|
|Author    : Mark Thomas<mark@apache.org>|
|Message   : Fix https://bz.apache.org/bugzilla/show_bug.cgi?id=60932 Corr...|
+-----+
  
```

[2] 2nd candidate of overlooked change is:

- java/org/apache/jasper/resources/LocalStrings_fr.properties

開発者の変更内容

変更忘れ候補

理由

条件付き確率 66%

過去の類似変更

15

東京工業大学 情報理工学系 小林研
Software Analytics Laboratory

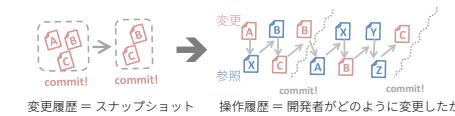


東京工業大学 小林研
情報理工学系
Software Analytics Laboratory

② 東京工業大学 情報理工学系 小林研
Kobayashi, Research Laboratory

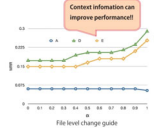


東京工業大学 情報理工学系 小林研
Rakuten Research Laboratory



バグ防止：変更漏れ発見
より詳細なマイクロプロセスの再利用実験

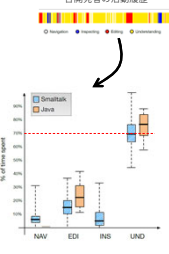
- 過去の開発経験を応用できるか？
 - 学生 17 名で「陽気な配管工が活躍するゲーム」を拡張
 - 例：Hammer Marioの導入，バックンフラワーが炎をはく
- 開発活動を記録して学習
 - どのファイル・関数を参照・変更したかを記録
 - 次の変更を予測できるか？を評価
- 現在までの結果：結構当たる！




23

バグ防止：Just-in-time 変更推薦
変更支援手法：国際共同研究への発展

- 開発者の活動分析 (2014)
 - スイス Lugano大 と実施
 - 操作ログを持ち寄り分析
 - Java: 15名, SmallTalk 7名
 - 発見：70+%を 理解に費やす
 - それまで定説では 35%程度
- 長期開発データの分析 (2017)
 - ノルウェー Oslo大 と実施
 - 4名の企業開発ログを分析
 - 先方は別の目的でログを取得



25

小林研が特に力を入れている領域 [再掲]

- ソフトウェア保守・進化
 - 既存のソフトへの機能追加
 - 明示的・潜在的なバグを生み出さない変更を支援
 - 既存のソフトのデバッグ
 - バグの箇所を特定し，取り除くための支援



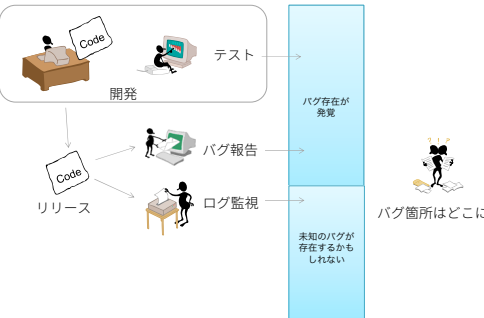
26

デバッグ支援の取り組み

- バグ発見・個所特定
 - 情報検索によるバグ箇所特定 (2020~)
 - 深層学習によるバグ含有予測 (2019~)
 - 動的解析を用いたバグ箇所特定 (2011~)
 - 大規模実行トレース解析による異常検知 (2019)
- 新しいデバッグ環境の開発
 - 仮想ファイルシステムを用いたデバッグ環境 (2017~)
 - Jupyterを用いたデバッグ環境 (2020)
- 振る舞い・機能に特化したプログラム理解支援
 - 実行履歴の抽象化・可視化(2009~)
 - 機能実現個所の特定 (2012~)
 - プログラム要約 (2017~)

27

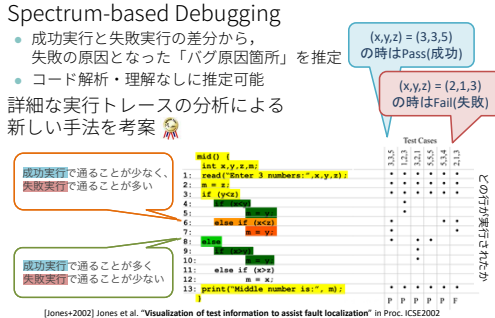
デバッグ支援：
バグ発見・個所特定



28

デバッグ支援：バグ箇所の特定
バグ箇所の自動特定

- Spectrum-based Debugging
 - 成功実行と失敗実行の差分から，失敗の原因となった「バグ原因箇所」を推定
 - コード解析・理解なしに推定可能
- 詳細な実行トレースの分析による新しい手法を考案



30

デバッグ支援：新しいデバッグ環境 文芸的デバッグ環境

- 新しいデバッグ支援環境 JISDLab を開発
 - 観測の影響を最小限に
 - 実行を停止せずに観測
 - 実行中に「観測したい値」を変更可能
 - Scriptable Debugging + Literate Program
 - デバッグ作業をスクリプト記述
 - 観測結果の加工・保存も可能
 - “実行できるバグレポート”の実現
 - 不具合の再現・状況確認・解析を可能に

Webブラウザから
Web App @Tomcat をデバッグ

Available at
github.com/tklab-group/JISDLab

32

デバッグ支援：プログラム振る舞い理解支援 実行履歴のシーケンス図による可視化

- 実行履歴の抽象化・可視化
 - 一般的なソフトウェアの実行 = 可視化困難
 - プログラムを解析しオブジェクトをグループ化
 - 全体像が把握しやすい「実行概要」を提供

生成したシーケンス図

提案手法で生成する実行概要シーケンス図

34

デバッグ支援：プログラム振る舞い理解支援 シーケンス図の活用

- SDExplore: 大規模SD表示ツールキット
 - Won Best Tool Demo Award @ IEEE/ACM ICPC 2018

Program Comprehension with Reversed SD

Interactive Loop Summarization

Interactive group folding/unfolding

Rich features for exploration

- Searching
- Filtering
- Zooming
- ... etc.

Operation Logging

Records interactions to support user activities analysis.

Browser-based tool

- Easy to use
- Platform-free
- Embed to Web page

Smooth exploration for massive-scale SD

< 1 sec responses time for SD with 3K objs, 2.5M mugs

35

これまでの取り組み：SE 4 IaC

- VM・コンテナによる環境(インフラ)構築
 - 例：レンタルサーバ, Amazon AWS, Dockerの利用
 - 手作業による間違い防止・効率化が必要
 - IaCスクリプト記述による自動化が主流に
- IaCスクリプトも「コード」
 - 多量のIaCスクリプト = やはり理解しづらい
 - ソースコードと同様にソフトウェア工学が必要
- SE 4 IaC
 - 設計支援：記述支援, 推薦・警告, 性能見積り
 - 保守支援：理解支援, バグ発見, リファクタリング
 - ...

19年度 卒論

21年度 卒論

41

これまでの取り組み：IaCの可視化

Infrastructure as Code

ソースコードと同様に
リバースエンジニアリング手法を適用

実際のサービス設定ファイルに適用

理解困難

IaC スクリプト記述

Kubernetes用可視化ツール (19年度 卒論)

42

2023年度メンバー

- 学生は10名前後
 - 教員：小林
 - 行動解析(変更推薦) x 2, IaC
 - M2：3名
 - 行動解析(変更推薦), バグ箇所検索, 未定
 - M1：3名
 - 未定 x 3
 - 学部：3名
 - 未定 x 3
 - 短期滞在・交換留学：若干名
- 連携研究室・大学・企業
 - 林研@大岡山とはいろいろ連携
 - 大学：京大, 奈良先端大, スイス Lugano大, 濠 Adelaide大
 - 企業：IT/SI企業 1社

修士は 3~4名/学年
他大学からの
入学も多数実績あり

44

