

機関番号：14401

研究種目：若手研究（B）

研究期間：2009～2010

課題番号：21700030

研究課題名（和文）

ソフトウェア部品間のデータ授受関係解析に基づく部品接続誤りの自動検出

研究課題名（英文）

Automatic Fault Detection based on Data-Flow Analysis for Software Components

研究代表者

石尾 隆（ISHIO TAKASHI）

大阪大学・大学院情報科学研究科・助教

研究者番号：60452413

研究成果の概要（和文）：本研究では、ソフトウェア部品の間でやり取りされているデータの流れを抽出する新しいアルゴリズムを構築した。既存手法は計算に長い時間がかかることが知られていたが、本研究では、ソフトウェアの動作を一部近似して計算する方式によって、高速な計算を実現した。構築した手法を用いて、複数のソフトウェアに対する適用実験を行った結果、ソフトウェア部品の典型的な接続情報を抽出できること、プログラムの調査に有効であることを確認した。

研究成果の概要（英文）：In this research, we have developed an algorithm of data-flow analysis for software components. While an existing data-flow analysis takes much time to analyze software, our research achieved an efficient analysis based on an approximation of the behavior of software. We have conducted case studies that show our algorithm extracts typical data-flow among components and our algorithm supports program understanding.

交付決定額

（金額単位：円）

	直接経費	間接経費	合計
2009年度	1,700,000	510,000	2,210,000
2010年度	1,300,000	390,000	1,690,000
年度			
年度			
年度			
総計	3,000,000	900,000	3,900,000

研究分野：ソフトウェア工学

科研費の分科・細目：情報学・ソフトウェア

キーワード：プログラム解析，データフロー解析

## 1. 研究開始当初の背景

社会基盤となる大規模ソフトウェアは、その品質を向上するために、次のような手順で開発されることが多い。

- (1) ソフトウェアを多数の部品（コンポーネント）の集合として設計する。
- (2) 個別に十分なテストを実施された部品を、適切な形で接続する。

しかし、実際には、ソフトウェア部品に対して、誤ったデータを入力してしまうことが、

しばしばソフトウェアの欠陥につながる。この原因は、各部品間でやり取りされるデータに関するルール、すなわちインタフェースを厳密に表現することが困難なためである。開発者によるインタフェースの確認を容易にする技術が必要とされている。

## 2. 研究の目的

本研究の目的は、既存ソフトウェアから、

ソフトウェア部品間のデータ授受関係を抽出し、現在開発中のソフトウェアと比較することで、インタフェースの使用誤りの可能性を開発者に提示する手法を構築することで、これにより、ソフトウェア部品を適切に組み合わせた効果的なソフトウェア開発を可能とする。

### 3. 研究の方法

研究目的を達成するために必要な要素技術として、2つの研究課題に取り組んだ。各研究は、それぞれ、さらに2つずつの作業に分けて実施した。

(1) 既存ソフトウェアから、部品間でのデータ授受関係を抽出する手法の構築 (2009年度)。

- ①ソフトウェアのソースコードが入力されたとき、そこからデータ授受関係を抽出する技術の構築。
- ②ソフトウェアの実行時に、ソフトウェア部品間で実際に生じているデータ授受関係の分析。

(2) ソフトウェアに含まれたデータ授受関係を可視化し、開発者による調査を可能とする技術の構築 (2010年度)。

- ① (1)①で構築したデータフロー解析アルゴリズムの統合開発環境 Eclipse への組み込み、およびその評価実験。
- ②ソフトウェアの欠陥の調査方法に関する企業の開発者へのインタビューの実施と、欠陥調査専用のツールの試作。

なお、解析技術をツールとして実装するにあたって、解析対象のプログラミング言語には、企業の情報システムで数多く使われている Java を採用した。

### 4. 研究成果

研究方法で述べた各取り組みに対応して、以下のような成果を得た。

(1) 新しいデータフロー解析手法

- ①Java のソースコード集合から、従来手法に比べて高速にデータフロー情報を抽出する技術を実現した。従来の手法では、データフローとはソフトウェアが実行する命令の順序から求められるもので

あったが、本研究では、命令の実行順序を無視した近似計算を行うことで、実用上許容できる正確性の低下と引き換えに、大幅な速度の向上を実現した。具体的には、10 万行程度のソフトウェアで、従来手法では 30 分程度かかっていた処理を、新手法では 5 分程度で実行することができる。開発者が実施する日々の作業は、30 分から 1 時間程度と、比較的短い時間区切りで行われることが多いという統計もあり、既存手法に比べ、より実際の開発環境で適用しやすいという利点を得られた。

複数のオープンソースソフトウェアを対象に、ソフトウェア部品間のデータ授受関係を抽出し、データフローにソフトウェア部品の典型的な使用方法が反映されていることを確認した。図 1 は、あるソフトウェア集合における、ソフトウェア部品間のデータ授受関係の分布を計算したものである。ソフトウェアは、それぞれに固有のデータ授受関係を数多く含んでいる一方で、頻繁に再利用される部品に対する典型的なデータ授受関係も数多く抽出することができることを示している。

なお、本研究では、あくまでデータ授受関係の調査を主目的とした実験を行った。しかし、データフロー情報は、たとえばソフトウェアの複雑さを数値として評価するための方法など、複数のプログラム解析技術に適用されている。英国 King's College London の Mark Harman 教授らの研究グループと、本研究で構築した手法の応用について、意見を交換している。

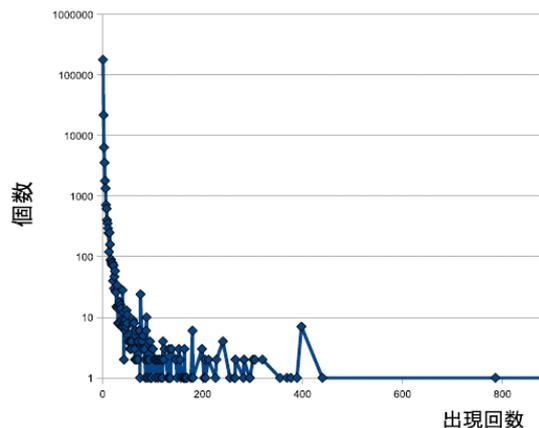


図 1 データ授受関係の分析例

- ②ソフトウェアの実行中に、その内部で使われている部品間での実際のデータの流れを抽出、グラフとして可視化するソ

ソフトウェア ROBIN を試作した。図2は、ROBIN を用いて、ある予定管理プログラムの中で作られたデータの流れを可視化したものである。この図において、1つの四角形が1つのソフトウェア部品に対応しており、矢印は部品間のデータの流れを表現している。このツールを用いて、複数のソフトウェアを分析した。

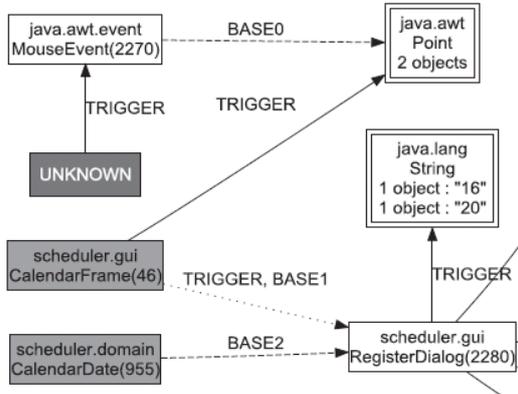


図2 プログラム実行中のデータ授受関係

(2) データフロー分析環境の構築

①データ授受関係の詳細を開発者に提示するための環境として、統合開発環境 Eclipse に(1)①で開発したアルゴリズムの実装を組み込み、開発者が選択したプログラム要素に関するデータ授受関係をグラフとして可視化するツールを構築した。このツールによって実際に可視化されるグラフの例を図3に示す。開発者がソースコードからプログラム要素を選ぶと、その要素に対応する頂点が画面上で黄色く強調表示され、その周辺のプログラム要素とのデータ授受関係が出力される。グラフ上の楕円および長方形で表現されている頂点をマウスで選ぶと、開発者は、それに対応したソースコードを閲覧することができる。

開発者にとってデータ授受関係の理解が容易であることを評価するために、企業の開発者4名と大学院生12名に対して、ツールを使用して特定の作業を実施してもらい実験を行い、ツールの有効性を統計的手法によって確認した。また、この実験では、企業の開発者が、統合開発環境を使ってどのように作業をしているか、その手順も分析した。その結果、開発者がソフトウェアのソースコードを読解して集めている情報が、ツールが画面上に提示する情報と同一であり、ツールの実現方法が妥当であることを確認した。

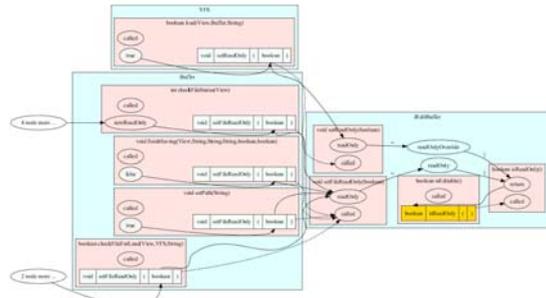


図3 Eclipse 上で表示されるグラフ

②ソフトウェアに欠陥がないかどうかをソースコードから確認するインスペクション作業のやり方について、ある企業の開発者にインタビューを行った。この結果を踏まえて、開発者が情報を知りたいと考えているソフトウェア部品をキーワード検索によって抽出しておき、各ソフトウェア部品の記述に共通して出現するキーワードやデータ授受関係の情報を表形式で提示するツール Cage を試作した。このソフトウェアは複数の企業の開発者に提供しており、試用結果に基づくコメントを得ている。今後改良すべき点は指摘されているが、それらが改良されれば一部業務への適用も可能であるとのコメントも得ており、今後、企業の中で長期的な評価や、ソフトウェア開発への活用につながることを期待される。

開発したツール Cage のスクリーンショットを図4に示す。縦軸にソフトウェア部品の一覧を提示し、横軸には2つ以上のソフトウェア部品に共通した特性の一覧を並べている。各ソフトウェア部品がその特性を持つことを青色の四角形のマス目で表現しており、利用者は、直感的に、複数の部品に共通した特性を把握することができる。

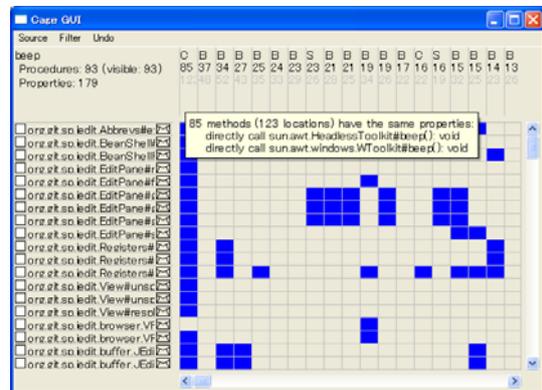


図4 複数の部品に共通した特性の表示

以上の成果を整理すると、本研究では、新しいデータフロー解析技術を構築し、その結果を企業の開発者にとって利用可能なソフトウェアの形式で試作した。試作したソフトウェアについては、企業の開発者に配布を行っている。

#### 5. 主な発表論文等

(研究代表者、研究分担者及び連携研究者には下線)

[雑誌論文] (計2件)

①渡邊 結, 石尾 隆, 井上 克郎, 協調動作するオブジェクト群の変化に基づく実行履歴の自動分割. 情報処理学会論文誌, 査読有, Vol. 51, No. 12 (2010), pp. 2273-2286

②市井 誠, 石尾 隆, 井上克郎, 複数のソフトウェアを横断した利用関係に基づくユーティリティクラスの自動検出. 情報処理学会論文誌, 査読有, Vol. 50, No. 10 (2009), pp. 2514-2519.

[学会発表] (計10件)

①悦田翔梧, 石尾 隆, 井上 克郎, 変数間データフローグラフを用いたソースコード間の移動支援. 第171回ソフトウェア工学研究発表会, 2011. 3. 14, 化学会館

②Takashi Ishio, A lightweight dataflow analysis to support source code reading. 12th CREST Open Workshop, 2011. 1. 24, King's College London (イギリス)

③中野 佑紀, 伊達 浩典, 渡邊 結, 石尾 隆, 井上 克郎, オブジェクト生成関係抽出ツール ROBIN, 第17回ソフトウェア工学の基礎ワークショップ, 2010. 11. 18, 新潟県南魚沼郡

④石尾 隆, 井上 克郎, コード片に共通した特性を自動抽出するソースコード閲覧ツールの試作. 第170回ソフトウェア工学研究発表会, 2010. 11. 12, 大阪大学

⑤柳 慶吾, 石尾 隆, 井上 克郎, ソフトウェア部品利用例抽出のためのデータフロー解析手法の提案と評価. 第167回ソフトウェア工学研究発表会, 2010. 3. 19, 国立情報学研究所

⑥宗像 聡, 石尾 隆, 井上 克郎, クラス動作シナリオ可視化手法のプログラム理解作業に対する有効性評価, 第167回ソフトウェア工学研究発表会, 2010. 3. 18, 国立情報学研究所. 情報処理学会ソフトウェア工学研究

会CS領域章受賞.

⑦伊達 浩典, 石尾 隆, 井上 克郎, オープンソースソフトウェアに対するコーディングパターン分析の適用, ソフトウェアエンジニアリングシンポジウム 2009, 2009. 9. 9, 東京女子大学

#### 6. 研究組織

##### (1) 研究代表者

石尾 隆 (ISHIO TAKASHI)

大阪大学・大学院情報科学研究科・助教

研究者番号: 60452413

##### (2) 研究分担者

無し

##### (3) 連携研究者

無し