様　式　Ｃ－１９、Ｆ－１９、Ｚ－１９　（共通）

科学研究費助成事業　　研究成果報告書

平成 ２７ 年　 ６ 月 ３０ 日現在

機関番号：２０１０３
研究種目：挑戦的萌芽研究
研究期間：2012～2014
課題番号：２４６５００１３

研究課題名（和文）本質的な構造と偶有的な構造の区別に基づく解析および操作の手法

研究課題名（英文）Identification, analysis and manipulation of essential structure from accidental ones

研究代表者
神谷　年洋（Kamiya, Toshihiro）

公立はこだて未来大学・システム情報科学部・准教授

研究者番号：７０４１５６６０

交付決定額（研究期間全体）：（直接経費）　　2,500,000円

研究成果の概要（和文）：大規模なコンピュータ・ソフトウェアを開発する際にプログラムにどうしても作りこまれてしまう偶有的な構造による複雑さは、開発者がプログラムを修正する際に問題となる。本研究では、偶有的な構造が含まれることによる問題に対処し、偶有的な構造の差異による見かけ上の違いにかかわらず、開発者がプログラムを分析する手法を提案し、ツールとして実装した。
具体的には、手続きの名前や値をキーワードとして与えると、それらの名前や値を含むなるべくコンパクトな実行系列を求め、該当するプログラムのソースコード部分を提示する検索手法、および、ソフトウェアのリファクタリングによる影響を取り除いた類似コード判定手法を提案した。

研究成果の概要（英文）：In developments of large computer software, accidental structures are usually included in their programs. Such accidental structures are additional complexity, problems for developers of such programs. This study aims to solve such problems of accidental structures, with proposed methods of program analysis, which consider (neglect) such surface differences in accidental structures: a keyword search method to extract a set of parts of source code which corresponds to an execution sequence including all of given keywords as name of procedure or values, and a method to detect similar code which tolerate changes in program structures by refactoring.

研究分野：ソフトウェア工学

キーワード：プログラム解析・理解　ソフトウェア可視化　デバッグ支援　コードクローン

1.研究開始当初の背景

コンピュータのソフトウェアを開発する技術が進歩し、粒度が大きくて複雑な部品を素早く組み合わせることで、多くの（あるいは大きな）機能をもつソフトウェアを開発することが可能になってきた。その結果として、開発されるソフトウェアに含まれる構造には、開発者が意図して作った構造と、開発者が意図せずに（再利用した部品にはじめから作りこまれていたり、あるいは再利用した部品を組み合わせるために要求されたりといった理由により）作ってしまった構造という　2種類の構造が含まれてしまう。

後者の、コンピュータのソフトウェアには、ソフトウェアを開発するためだけに必要とされ、ソフトウェアの機能からは必須ではない構造は、フレデリック・　P・ブルックス　Jr 氏によって「偶有的な構造」と名付けられて昔から知られていた。しかし、現在の開発技術の発達により、具有的な構造の問題がより増幅されるようになり、その解決のためのより強力な手法が必要となっている。

2.研究の目的

従来の開発技術、開発手法や分析技術の多くは、上述の「偶有的な構造」と本質的な構造を区別しない。（わずかに、リファクタリングと呼ばれる作業は「機能を修正せずに構造を綺麗にする」という言い方により、言外に偶有的な構造と本質的な構造を区別している。）そのような偶有的な構造の割合が増えるに従い、ソフトウェアの保守にかかる手間が増える。具体的には、 (P1) 理解 ： 偶有的な構造がソフトウェアに内在することにより、開発者はソフトウェアのの構造のうちで何が偶有的であり、何が本質的であるかを手作業で区別する必要がある。 (P2) 本質的な構造の修正 ： 本質的な構造を修正しようとしても、偶有的な構造の制約に引っかかり、偶有的な構造を合わせて修正することが必要となる。 (P3) 偶有的な構造の修正　：リファクタリングなどの手段で、具有的な構造を修正しようとして、本質的な構造を誤って壊してしまうことがある。本研究はこの　P1 から　P3 の課題を解決することを目的とする。

3.研究の方法

本研究は、従来のリバースエンジニアリング手法にデータマイニングの手法を取り入れ、開発技術が本質的な構造と偶有的な構造を区別できるようにし、さらに、本質的な構造を破壊的な影響を与えずに偶有的な構造を修正するための手法を確立する。これにより、上記 (P1) から (P3) の問題について、それぞれ以下のような解を与える。 (S1) 開発者が本質的な構造（すなわち、アプリケーションが本来解くべき課題）に集中する可能にする、 (S2) 偶有的な構造の制約により変更できないときに、別の（偶有的な）構造によって再実装することを補助する、 (S3) 本質的な構造を壊さないことを保証しながら偶有的な構造を修正することを可能にする。

4.研究成果

(1) プログラムの全実行パスを対象としたキーワード検索とその視覚化手法

多くの部品を再利用して開発しているプログラムによって実現されているソフトウェアを対象として、特定の処理や値の参照・生成が、そのプログラムのどのような実行系列によって行われているかを検索する手法を提案した。

本手法の特徴は、（1）対象となるプログラムの潜在的な全実行系列からの検索でであること（2） プログラムの特定の1箇所を見つけるのではなく、実行系列の部分列（特定の処理に関わる複数の箇所）を検索結果とすること、（3） なるべく小さな（したがって理解しやすい）実行系列を結果とするような視覚化手法を伴うこと、である。

潜在的な全実行系列からの検索を可能にするため、プログラムの一種の抽象実行（プログラムを実際に実行するのではなく、実行した時にどのような実行系列になるかを特定する）に

より、潜在的に可能なすべての実行系列を生成し、そのすべての実行系列上でキーワード検索を行う処理となっている。ただし、プログラムの全実行系列を実際に生成してしまうと天文学的な数になるため、ナイーブな手法で実際のプログラムには適用できない。これを実現するために And/Or/Call 木と名づけたデータ構造を提案した(図1参照)。

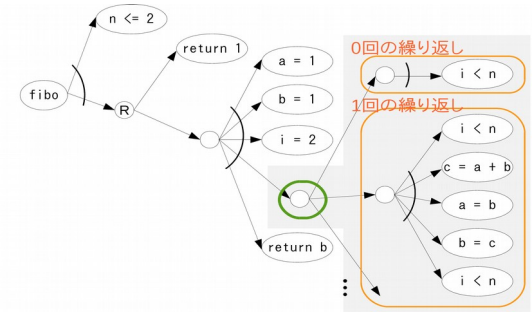

図 1 And/Or/Call 木の例

このデータ構造は、与えられたプログラムの全実行系列を表現し、かつ、その上でキーワード検索アルゴリズムを実行できるようなコンパクトなデータ構造である。 And/Or 木にプログラムの制御構造を割り当て、さらに、プログラム内の手続き呼び出しを Call 節点という節点で表現することで、1つのプログラムを1つの木として表現する。さらに、木の中のすべての Or 節点および Call 節点を「展開」することで、そのプログラムのすべての実行系列に相当する木を生成することが可能である。

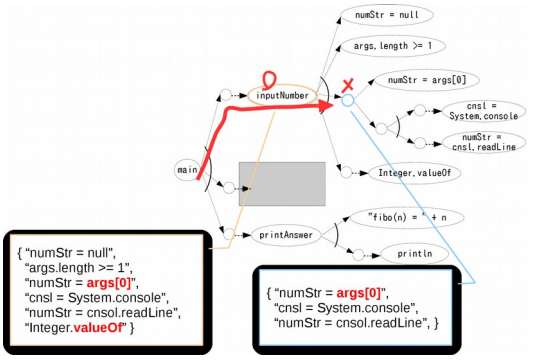

図2 And/Or/Call 木上でのキーワード検索アルゴリズムの実行例

さらに、この木の中で、検索キーワードに該当するラベルをもつ節点を残し、それ以外の節点を、連結性を損なわないなどの制約を満たしながら除去して得られた部分グラフを作ることにより、キーワード検索を行うことができる(図2参照)。この And/Or/Call 木による実行系列の表現とキーワード検索アルゴリズムを、静的型付けプログラミング言語である Javaと、動的型付けプログラミング言語である Python という2種類の性質が異なるプログラミング言語に対応するツールとして実装し、予備的な実験によって検出手法の精度や性能を確認した。

これらの成果を、 5. 〔学会発表〕における④、⑩、⑬で発表(うち1件は IEEE の国際会議)した。

(2) プログラムの振る舞いにの類似性に基づく類似コード検索手法

特にリファクタリングによってプログラムの構造が変更された場合を想定した、振る舞いが互いに類似したコード断片の集合を検出する手法を提案した。この手法により、例えば、あるソフトウェアの中に互いに類似したコード断片が3箇所あり、その2箇所についてはリファクタリング(振る舞いを変更することなく構造を変更)したときであっても、残る1箇所を、修正済みの他の2箇所と類似していることを、本手法により特定することが可能となる。

このように、本手法は、ソフトウェアの保守作業において、類似コード断片が含まれるようなプログラムにリファクタリングを行った際に結果として生じる「偶有的な構造」の差異による見かけ上の違いによる開発者のミス、および、その結果としての品質の低下を防ぐことが期待される。

本研究では、リファクタリング手法であるの pull up methodなどによるメソッドの粒度の変化の影響を取り除いた解析を行うため、プ

ログラムの手続き呼び出しを任意回数展開した上で比較するアルゴリズムを提案した。また、当初は手続き呼び出しの順序が同じもののみを類似していると判定する手法であったが、その後の発展により、手続き呼び出しの順序の入れ替えや他の処理が挟まっているような場合においても類似部分を抽出する手法を実現した。

本手法の特徴は、上述したように(1)リファクタリングによるプログラムの構造変化を前提として、そのような変化があってもなお類似した振る舞いとみなせる部分を検出する手法であること、(2)類似部分だけではなく、それらを呼び出す部分(コンテキストと呼ぶ)の分析も含むことで、プログラムを保守する開発者に対して修正の手がかりを提示する手法であること、の2点である。

本手法では、特定の手続き呼び出しが内包する実行系列(コンテンツ)とその手続き呼び出しを利用するコンテキストを、プログラム中の各手続き呼び出しに対して生成し、コンテンツの類似とコンテキストの類似の双方を特定する(図　3 参照)。

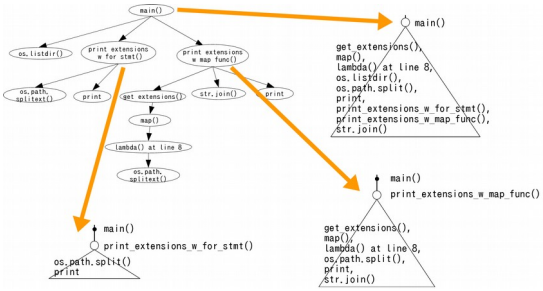

図3 プログラムの木による表現と各手続き呼び出しのコンテンツ／コンテキストを抽出した例

また、検出した類似コード断片についてそのコンテキストも1つの図として提示することで、開発者に対して修正の手がかりを与える。図　4 は提案手法を実際のソフトウェアに対して適用した例であり、互いに類似したコード断片の集合の間の差異に相当する部分が、また別の類似コード断片の集合になっていることを突きとめた例である。

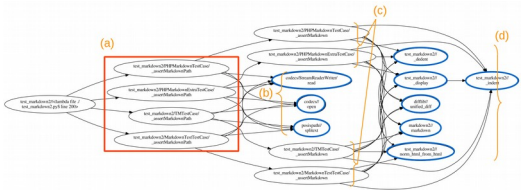

図4 提案した類似コード検出手法の適用例

提案した手法およびその実装による実験を、5. 〔学会発表〕における⑤、⑥、⑦、⑪、⑫で発表(うち2件は　IEEE の国際会議)した。

5.主な発表論文等

〔雑誌論文〕(計3件)

① Toshihiro Kamiya, "An Execution-Semantic and Content-and-Context-Based Code-Clone Detection and Analysis," Proceedings of the IEEE 9th International Workshop on Software Clones (IWSC 2015), pp. 1-7 (Mar. 6, 2015). 査読あり

② Toshihiro Kamiya, "An Algorithm for Keyword Search on an Execution Path," Proc. IEEE Conference on Software Maintenance, Reengineering and Reverse Engineering (CSMR-WCRE 2014), pp. 328-332 (Feb. 6, 2014) 査読あり

③ Toshihiro Kamiya, "Toward a Code-Clone Search through Entire Lifecycle of Software Product," Proc. 8th International Workshop on Software Clones (IWSC 2014), pp. 1-8 (Feb. 3, 2014) 査読あり

〔学会発表〕(計10件)

① 神谷年洋　, "任意粒度機能モデルに基づく動的型付けプログラミング言語向けソースコード検索手法の提案 ", 電子情報通信学会技術研究報告 Vol. 114, No. 415, pp. 121-126 (2015-01-27). 査読なし　. ブランナールみささ(鳥取県東伯郡三朝町)　.

② 神谷年洋 , "Webアプリケーションの UI 機能テストのための HTML構造パターンの抽出手法 ", 電子情報通信学会技術研究報告 Vol. 114, No. 127, pp. 81-85 (2014-07-10). 査読なし . 富良野文化会館(北海道富良野市弥生町) .

③ 及川 翔 , 神谷年洋 , "Webアプリケーションの UI 機能テストのための HTML構造パターンの提案 ", 電子情報通信学会 技術研究報告 Vol. 114, No. 23, pp. 37-42 (2014-05-09). 査読なし , いせ市民活動センター(三重県伊勢市岩渕) .

④ 神谷年洋 , "And/Or/Call グラフの提案とソースコード検索への応用 ", 電子情報通信学会技術研究報告書 , vol. 113, no. 269, pp. 173-178 (2013/10/24). 査読なし . IT ビジネスプラザ武蔵(石川県金沢市武蔵町) .

⑤ 神谷年洋 , "任意粒度機能モデルに基づくコードクローン検出手法の大規模プログラムへの適用に向けた改善 ", 電子情報通信学会技術研究報告書 , vol. 113, no. 159, pp. 133-137 (2013/07/26). 査読なし . 北海道立道民活動センター(北海道札幌市中央区) .

⑥ Toshihiro Kamiya, "Agec: An Execution-Semantic Clone Detection Tool," Proc. 21th IEEE International Conference on Program Comprehension (ICPC 2013), pp. 227-229 (May 21, 2013). 査読あり . サンフランシスコ(米国)

⑦ 神谷年洋 , "任意粒度機能モデルに基づくバイトコードからのコードクローン検出手法 ", 電子情報通信学会技術研究報告書 , vol. 113, no. 24, pp. 43-48 (2013/05/10). 査読なし. 香川大学(香川県高松市幸町).

⑧ 及川 翔 , 神谷 年洋 , "画面トレースの解析による webアプリケーションのユースケース再構築手法の提案 ", 電子情報通信学会技術研究報告書 , vol. 112, no. 275, SS2012-45, pp. 129-134 (2012/11/02) 査読無し. 広島市立大学(広島県広島市安佐南区).

⑨ Toshihiro Kamiya, "Conte∗t Clones or Re-thinking Clone on a Call Graph," Proc. 6th International Workshop on Software Clones (IWSC 2012), pp. 74-75 (June 4, 2012). ポジションペーパー. チューリッヒ(スイス).

⑩ 神谷年洋 , "コードの内容と文脈を用いた類似コード分析手法の提案 ", 電子情報通信学会技術研究報告書 , vol. 112, no. 23, pp. 19-24 (2012/05/10). 査読なし. 愛媛大学 総合情報メディアセンター (愛媛県松山市文京町).

6.研究組織

(1) 研究代表者

神谷 年洋(KAMIYA, Toshihiro)

公立はこだて未来大学・システム情報科学部・准教授

研究者番号: 70415660