

DebugConcierge: クラウド知識に基づいたデバッグ支援

廣瀬 賢幸^{†1} 鷓林 尚靖^{†1} 細合 晋太郎^{†1}
亀井 靖高^{†1} 渡邊 卓也^{†2}

現在、様々な自動バグ修正技術が提案されているが、必ずしも期待する効果が得られていない。依然として手動でのバグ修正が主流である。しかしながら、手動でのバグの修正には多くの手間と時間がかかる。本稿では、このような問題を解決するため、デバッグのための情報推薦システム DebugConcierge を提案する。DebugConcierge は、クラウド知識に基づいたコード修正案の推薦、コードの自動修正、修正レベル（修正、暫定修正、応急処置）の管理、の 3 つの側面からデバッグを支援する。

1. はじめに

ソフトウェアからバグを取り除くことは、多くの場合、手作業である。自動バグ修正の研究もされているが、適切な修正をするのは、まだ現実的でない²⁾。一方、手動でのバグ修正には多くの手間と時間がかかる。

例えば、プログラマが抱える悩みとして、以下の 3 つが考えられる。

- (1) 修正方法がわからないので、ヒントが欲しい。
- (2) 修正のためのコーディングを省きたい。
- (3) 修正方法は分かるが、修正する時間が取れない。

(1) については、修正の情報を推薦することで支援することができる。(2) については、コードを部分的に自動修正することで、支援することができる。(3) については、時間のかからない一時的な修正を行い、その修正を管理することで支援できる。

本稿では、これらの悩みを解決するため、情報推薦システム DebugConcierge を提案する。

2. DebugConcierge の機能と利用シナリオ

本稿で提案する情報推薦システム DebugConcierge は、Eclipse のプラグインとして実装する。対象とするプログラミング言語は Java である。DebugConcierge は、コード修正案の推薦、コードの自動修正、修正レベルの管理の 3 つの機能を提供する。

2.1 コード修正案の推薦

DebugConcierge では、2 種類の修正情報を推薦する。1 つは、GenProg^{*1} などの自動バグ修正ツールからの修正候補である。もう 1 つは、StackOverflow^{*2} (Q&A サイト) などのクラウド知識による修正ヒン

ト¹⁾ である。これらの修正情報の特徴を考えてみる。まず、自動バグ修正ツールのものは、バグ除去の確度が高いが、情報量が少ない。一方で、クラウド知識ベースのものは、情報量が多いが、バグ除去の確度は少ない。したがって、組み合わせることで、どちらか一方だけでは修正できなかったバグを修正できる。

開発者は、キーワードか、ソースコードの一部をもとにして、クラウド知識ベースからバグ除去の参考となる情報（修正例や API ドキュメントなど）を検索することができる。この情報検索機能には、プログラミング支援ツール CodeConcierge³⁾ の機能を利用する。

自動バグ修正ツールにはいくつかのタイプがある。たとえば、GenProg では遺伝的アルゴリズムを用いて、コードの順番を変更したり、コードブロックを削除したり、return 文を挿入したりして、バグの修正を試みる。Qi ra²⁾ の研究によると、これらの修正案は本質的なバグの除去にはほとんどの場合役立たないが、一時的にテストケースをパスさせるには有効である。

DebugConcierge では、バグの修正を、本質的なバグ修正、暫定的バグ修正の 2 種類に分けて考える。クラウド知識ベースから提示される修正案は本質または暫定的なバグ除去に、自動バグ修正ツールからの修正案はバグの応急処置に役立つ。

2.2 コードの自動修正

プログラマが選択した修正情報を実際にコードに反映させていく。修正候補と修正ヒントでは、コードの自動修正時の挙動が異なる。図 1 にその概略を示す。

図中の修正候補は、自動修正ツールからの修正案で、該当箇所にその修正を施す。図中では、return 文を挿入する案が推薦され、その案を採用すると、該当箇所に return とコードが挿入される。

一方、図中の修正ヒントは、クラウド知識からの情報で、修正情報のコードを該当ソースコードに対応するように変更を加える。例えば、変数や引数の名前

†1 九州大学

†2 エヂリウム株式会社

*1 <http://dijkstra.cs.virginia.edu/genprog/>.

*2 <http://stackoverflow.com/>

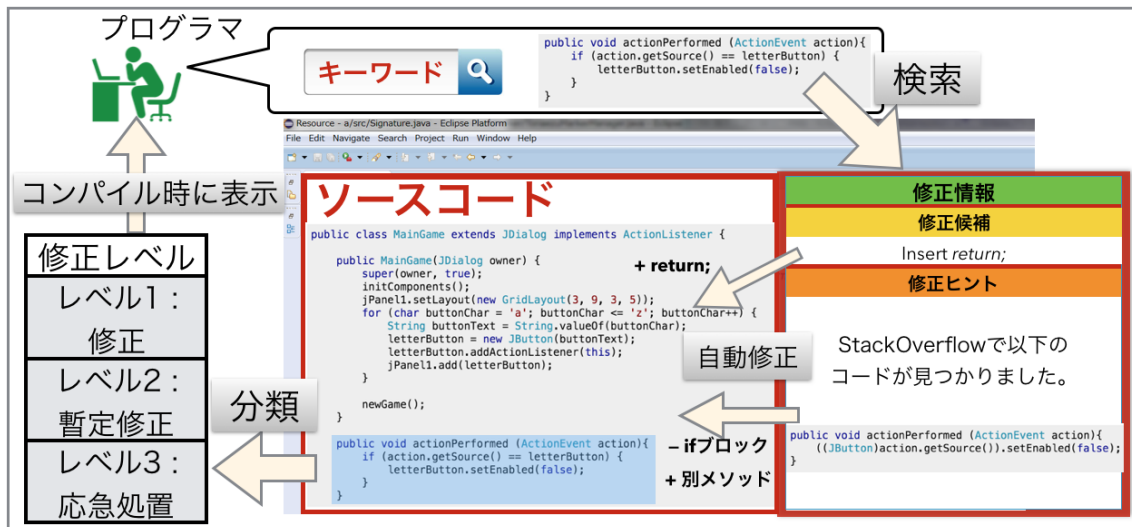


図 1 DebugConcierge の概要

を対象コードに合わせて変更する。図中では、ソースコード中の `actionPerformed` メソッドの実装を推薦された修正情報の実装に変更する。

2.3 修正レベルの管理

修正によるコードの変更を、プログラマが後から追跡できるよう、行われた修正のレベルを管理する。修正コードは本質的なバグ修正と暫定的なバグ修正の 2 種類に分けられる。さらに、暫定的な修正は、プログラマによりさらに細分化されこともある。図 1 では、次の 3 つの修正レベルに分けている。レベル 1 は、完全にバグを修正するコードである。レベル 2 は、将来的に修正方法を変えるような、暫定的な修正コードである。レベル 3 は、とりえずプログラムを動かしたいとき埋め込むような応急処置の修正コードである。本質的なバグ修正はレベル 1 で、暫定的なバグ修正はレベル 2 とレベル 3 に細分化されている。

修正コードのうち、暫定的な修正は、将来的にバグの温床になりうる。なぜなら、ソフトウェアが発展するにつれて、プログラマが暫定的な修正の存在を忘れてしまうからである。そのため、プログラマに暫定的な修正の存在を知らせる必要がある。

DebugConcierge では、コンパイル時にプログラマに暫定的な修正箇所を知らせ、本質的な修正を促す。また、git などのコード変更履歴管理システムにコードの変更の情報が記録される際に、修正レベルも共に記録する。図 1 にその概略を示す。修正コードを修正レベルに分類した後、コンパイル時にプログラマに修正コードの位置などの情報を知らせる。

3. 今後の課題

直近の課題として、本稿で述べた機能を実装することが挙げられる。本研究では、今後、コード修正案の精度向上、情報推薦支援の高度化に取り組みたい。具体的には、今のところプログラマが明示的に検索を行わなければ、修正情報は表示されないが、ファイル編集履歴など、プログラマの行動履歴に基づいて、プログラマの意図を予測し、プログラマが求めているような情報を予め推薦するような機能を作成したい。

謝辞: 本研究は、文部科学省科学研究補助費 基盤研究 (A) (課題番号 26240007) による助成を受けた。

参考文献

- 1) Fuxiang Chen, Sunghun Kim, Crowd Debugging, In *Proceedings of the 10th Joint Meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering (ESEC/FSE '15)*, pp.320-332, 2015.
- 2) Zichao Qi, Fan Long, Sara Achour, and Martin Rinard, An Analysis of Patch Plausibility and Correctness for Generate-And-Validate Patch Generation Systems, In *Proceedings of the 2015 International Symposium on Software Testing and Analysis (ISSTA '15)*, pp.24-36, 2015.
- 3) 高橋裕太, 鷗林尚靖, 細合晋太郎, 亀井靖高, 渡邊卓也, CodeConcierge: クラウド知識に基づいたプログラミング支援, ウィンターワークショップ 2016・イン・逗子, 2016.