

変数名の命名に着目したリファクタリング支援ツール MCC の試作

田上諭 片山徹郎
宮崎大学大学院 工学研究科 工学専攻

1 はじめに

近年, IT の進化とともに, 社会における情報システムの担う役割は段々と大きくなっている. そのため, システム障害やソフトウェアの不具合がもたらす経済的, 社会的影響は, 計り知れないものとなっている. このような背景から, 高品質なシステムが重要視されるようになった.

システムの品質を維持や改善する方法としてリファクタリングがある [1]. リファクタリングは, 外部的振る舞いを保ったまま, ソースコードが分かりやすくなるように改善することである. 意味不明な変数名があると, どのような処理をするのか理解しづらくなる [2]. ソースコードを修正する際に, 必要な処理を削除してしまったり, 意図された使用用途とは違うように変数や関数を使ってしまったりして, バグを混入させることが起こりうる.

そこで本研究では, コードの品質改善の支援を目的として, リファクタリング支援ツール MCC(Make Clean Coder) を試作した. MCC は, C 言語で書かれたソースコードに対して静的解析を行い, 変数名の命名に着目したクリーンコードの作成支援を行うツールである. MCC は, 命名が適切でない変数名を指摘することができる. なお本研究において, 命名が適切でない変数名とは, 変数名が 1 文字である場合や, 変数名が辞書にないものである. 辞書データには, 実験用に公開されている辞書であるデ辞蔵 Web サービス [3] を利用する. MCC を利用することにより, ソースコードを理解するための時間が減少し, コーディング時間の短縮, バグ混入の可能性の減少, 機能追加に必要な時間の減少が図れる.

2 リファクタリング支援ツール MCC

図 1 に, 試作した MCC の概観と実行結果を示す. MCC は, 次に示すメニューバー, 編集エリア, 表示エリアからなる. ユーザーは, メニューバーにある各メニューアイテムを選択することによって, MCC の機能を使用できる. MCC の機能としては, 「ファイルを開く」「ファイルを保存」「コードを解析」「編集エリアをクリア」「表示エリアをクリア」の 5 つの機能がある.

MCC の機能の 1 つである「コードを解析」では, コード内の変数宣言において, 適切でない変数名の場合に, 「行番号: 変数名~から役割が推測できません. 変数名を変更してください」と警告を表示エリアに表示する. スネークケースやキャメルケースの場合は, 複合語の元になっている単語それぞれを辞書 [3] で検索する. 適切でない変数名の場合は, 辞書に登録されていない単語をそれぞれ表示し, 「行番号: 変数名~の~から役割が推測できません. 変数名を変更してください」と警告を表示エリアに表示する.

3 適用例

MCC が, 「コードを解析」ができることを検証するために, C 言語で書かれたソースコードの例に適用した (図 1 参照). 編集エリアの 41 行目は, 変数名が `r_num[50]` となっている. これは, `r` は辞書には登録されているが, 1 文字であり, `num` は辞書に登録されていないので適切でない変数名であり, 「41: 変数名 `r.num[50]` の `r`, `num` から役割が推測できません. 変数名を変更してください」と警告を表示エリアに表示している. 同様に編集エリアの 42, 45, 46, 47, 48, 49, 50, 57, 58 行目も適切でない変数名があるので, 「変数名~から役割が推測できません. 変数名を変更してください」と警告を表示エリアに表示している. よって, 正しくコードを解析できていることが確認できる.

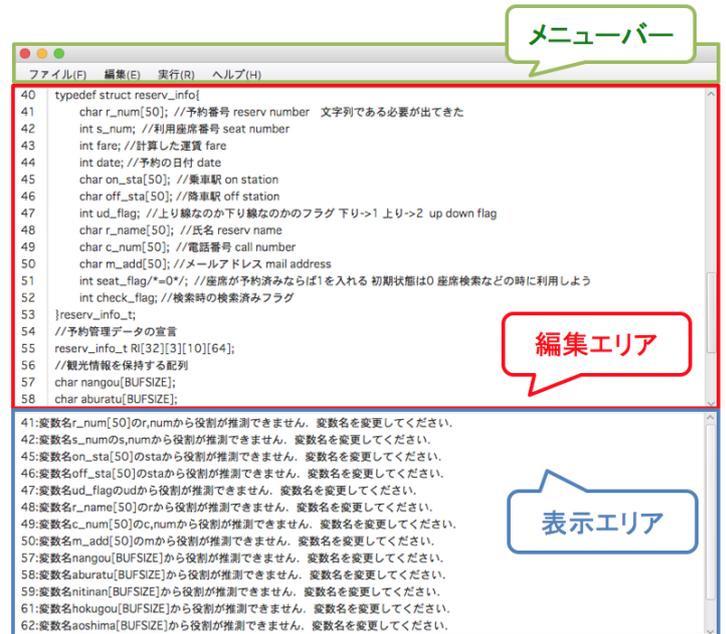


図 1: MCC の概観と実行結果

4 おわりに

本研究では, コードの品質改善の支援を目的として, リファクタリング支援ツール MCC を試作した. MCC は, C 言語で書かれたソースコードに対して静的解析を行い, クリーンコードの作成支援を行うツールであり, 変数名の命名に着目する.

本研究で試作した MCC は, C 言語で書かれたソースコードを読み込み, 静的解析を行う. 静的解析では, 変数名が適切に命名されているか辞書を用いて判断する. 変数名が適切でない場合は, その変数名のある行番号と変数名を表示し警告する. 加えて, MCC はソースコードの編集機能を備えている.

MCC に C 言語で書かれたソースコードの例を適用することによって, MCC が正しく動作することを確認した. MCC は, 命名が適切でない変数名を指摘できる. これにより, ソースコードを理解するための時間が減少し, コーディング時間の短縮, バグ混入の可能性の減少, 機能追加に必要な時間の減少が図れる.

以上のことから, 本論文で試作した MCC を利用することにより, コードの品質改善の支援ができると考えられる. 以下に, MCC の今後の課題を示す.

- 対応していない変数名への対応
- 関数名への対応

参考文献

- [1] Martin Fowler 著, 児玉公信 訳: “新装版 リファクタリング 既存のコードを安全に改善する”, オーム社, 2014.
- [2] Dustin Boswell, Trevor Foucher 著, 角征典 訳: “リーダブルコード”, オライリー・ジャパン, 2012.
- [3] デ辞蔵 Web サービス: “デ辞蔵 オンラインでもオフラインでも使える電子辞書”, <http://dejizo.jp/dev/index.html>