論 文 内 容 の 要 旨

氏 名 （ 佐々木 唯 ）	
論文題名	プログラム構造に着目したソースコード保守性の計測と改善に関する研究

論文内容の要旨

ソフトウェアのライフサイクル全体にかかるコストのうち保守コストの占める割合は高く，ソフトウェアの保守を効率化することが重要となっている．ソフトウェアの保守フェーズでは，ソフトウェアの保守性を計測または改善する活動が度々行われており，保守性に関連する指標の計測方法や，保守性に影響を与える要因について，様々な観点から研究が行われている．

本論文は，ソフトウェアの保守を効率化することを目指して遂行した，ソースコードの構造に着目して保守性を計測または改善する三つの研究について報告する．

一つ目は，ソフトウェアメトリクスを用いて理解性の低いモジュールを検出する場面において，効率的に検出するためのソースコード簡略化手法の提案である．サイクロマチック数などのソースコードの複雑さを表すメトリクスは，理解性の低いモジュールの特定に用いられることがある．しかし，ソースコード中に類似した構造の繰り返しが存在する場合，メトリクスの値が大きくても複雑であるとは限らない．そこで，メトリクス計測の前処理として，繰り返し構造を折りたたんでソースコード構造を簡略化する手法を提案する．評価実験の結果，提案手法を適用して計測したメトリクスの方が，理解性の低いモジュールの特定に有用であることが分かった．

二つ目は，ソースコードの可読性を向上させるための，プログラム文の並べ替え手法の提案である．ソースコードを読む作業は保守作業における開発者の行動の大半を占めるといわれており，特に変数が定義されてから参照されるまでの距離が離れていると，理解するためのコストが増大することが報告されている．従って，文の並びはソースコードの可読性に影響を及ぼすと考えられる．そこで，変数の定義と参照の間の距離に着目して，モジュール内の文を並べ替える手法を提案する．評価実験を行った結果，並べ替えの行われたモジュールは可読性が向上するという結果が得られた．

三つ目は，保守性の一つの観点としての，ソースコードに対する欠陥限局の適合性とその計測方法の提案である．欠陥限局とはソースコード中の欠陥箇所を推測する技術であり，近年はSpectrum-Based Fault Localization (SBFL) に関する研究が盛んに行われている．SBFLは，テストケースごとの成否と，どの文が実行されたかという情報を用いるため，同一機能を実装したソースコードであっても，記述や構造が異なればテストケースによって実行される文が変化し，SBFLの精度に違いが生じる可能性がある．そこで，ソースコードがSBFLにどの程度適しているかを，そのソースコードに対するSBFL適合性として提案し，更にその一つの評価指標としてSBFLスコアの計測手法を提案する．提案手法は，ミューテーションテスト技術を活用してソースコード中に意図的に発生させた欠陥を，SBFLによってどの程度正確に特定できたかを計測する．ソースコード構造の違いによるSBFLスコアの違いを分析した結果，SBFLスコアに影響を与えるソースコードの特徴を発見した．

様式７

論文審査の結果の要旨及び担当者

氏　名　（佐々木　唯　）		
	（職）	氏　名
論文審査担当者		
主 査	教授	楠本　真二
副 査	教授	井上　克郎
副 査	教授	増澤　利光

論文審査の結果の要旨

ソフトウェア保守は、既存のソフトウェアを改良、最適化していくとともに運用時に発見された不具合を修正するプロセスを意味する。ソフトウェアのライフサイクル全体に要するコストのうち保守コストが占める割合は90%以上であるという報告もあり、ソフトウェア保守作業の効率化はソフトウェア開発ベンダー、ユーザ双方にとって重要な課題となっている。本論文は、ソフトウェア保守支援の一つの方法として、ソースコードの構造に着目して保守性を計測、改善するために行った三つの研究成果をまとめたものである。

一つ目は、ソフトウェアの理解容易性の評価に関する成果である。ソフトウェア保守性を向上させるためにソースコードの理解容易性が低い箇所に対するリファクタリングがよく行われる。リファクタリングとは、ソフトウェアの外部的振る舞いを保ちつつ、理解や修正が簡単になるように、内部構造を改善する作業である。本論文では、ソフトウェアメトリクスを用いて理解性の低いモジュールを効率的に検出するためのソースコード簡略化手法を提案している。理解性の低いモジュールの特定には、ソースコードの複雑さを表すメトリクスが一般に用いられるが、自動生成されたコードや類似した構造の繰り返しが存在する場合、メトリクス値が大きくても複雑であるとは限らないことが指摘されている。その問題に対処するために、メトリクス計測の前処理として、繰り返し構造を折りたたんでソースコード構造を簡略化する手法を提案している。評価実験の結果、提案手法を適用して計測したメトリクスの方が、理解性の低いモジュールの特定に有用であることが確認されている。

二つ目は、ソフトウェアの理解容易性改善に関する成果である。ソフトウェア保守を担当する開発者には、ソースコードを読み、記述されている内容を理解する作業が求められる。本論文では、保守対象のソースコードの可読性を向上させるためのリファクタリング方法として、プログラム文の並べ替え手法を提案している。ソースコード上で変数が定義されている箇所から利用（参照）されるまでの距離が離れていると、理解するためのコストが増大することが知られていることから、変数の定義と参照の間の距離に着目して、機能的な変更をすることなくモジュール内の文を並べ替える手法を提案している。評価実験の結果、並べ替えの行われたモジュールは可読性が向上するという結果を得ている。

三つ目は、保守作業で実施される、運用時に発見された不具合修正の支援に関する成果である。近年、ソースコード中の欠陥箇所を自動的に推定する手法（欠陥限局手法）の一つであるSpectrum-Based Fault Localization（SBFL）に関する研究が活発に行われている。SBFL は、テストケース毎の成否と、文の実行情報を用いるため、同一機能を持つソースコードであっても、記述や構造が異なればテストケースによって異なる文が実行されることになり、欠陥限局の精度に違いが生じる可能性が指摘されている。本論文では、保守対象ソースコードがSBFL などの程度適しているかを、そのソースコードに対するSBFL 適合性として提案し、その一つの評価指標としてSBFL スコアの計測手法も提案している。提案手法は、ミューテーションテスト技術を活用してソースコード中に意図的に発生させた欠陥の場所を、SBFL によってどの程度正確に特定できたかを計測する。ソースコード構造の違いによるSBFL スコアの違いを分析した結果、SBFL スコアに影響を与えるソースコードの特徴を発見し、SBFL適合性の実際の保守現場での適用可能性を示している。

以上のように、本論文の成果は、ソフトウェア保守性の改善の実現において、技術面、並びに、実用面において高い貢献があると考えられ、博士（情報科学）の学位論文として価値のあるものと認める。