

審査の結果の要旨

氏 名 丸山 一貴

本論文は「プログラム実行点の概念に基づくデバッグパターンの抽出と自動化」と題し、これまでほぼ完全に手動で行われてきたプログラムのデバッグ作業について分析を行い、その一部を抽出して自動化するための研究をまとめたもので、以下の9章からなる。

第1章「背景」では、研究の背景、研究全体の概要と論文の構成について述べている。

第2章「デバッグパターン」では、プログラマが行うデバッグ作業についてその思考プロセスを分析して新しい枠組を提示し、作業の一部に再利用の可能性と自動化の余地が存在することを述べ、デバッグパターンの定義を与えている。デバッグパターンの抽出と自動化によって、熟練プログラマの経験に基づいた高度なデバッグ手法がプログラマ間で共有できると、作業時間の短縮が実現されることを述べている。また、従来の研究成果のいくつかがその枠組で説明できることを示している。

第3章「関数単位疑似逆実行」では、研究全体の中で予備実験としての色彩を持つ関数単位疑似逆実行の実装に関する知見について述べている。逆実行は従来広く研究されてきた分野であるが、そのオーバーヘッドの大きさから実用に供されてこなかった。本論文ではこれを、典型的な実行制御であるデバッグパターンの1つと位置付けている。本章では、プログラムの再現性を仮定し、実行制御における制御の移動先を関数呼び出しに限定した先行研究について紹介した上で、対象プログラムの変換を利用することで、実用レベルのオーバーヘッドで同機能が提供されることを示している。また、その結果をもって実行制御の自動化の有効性を述べ、関数呼び出しに限らない、より一般的な実行制御の基盤が必要であると論じている。

第4章「プログラム実行点」では、前章の論点を受けて、一般的な実行制御の基盤として実行点という概念の利用を提案している。この概念はデバッグ中のプログラムの多くが、プログラムの実行進捗度を把握するために用いているイメージである。本章では実行点の具体的な表現方法として4種類を挙げ、その中で行番号とタイムスタンプを用いた表現が適切であるとし、その手法について述べている。

第5章「C及びJavaへの実装」では、前章で提案した実行点の具体的な表現をポータブルに実現するため、第3章で用いたプログラム変換による手法を提案し、実装している。Cプログラムに対してはコンパイラの間言レベルでの変換を利用し、Javaプログラムに対してはバイトコードレベルでの変換を用いている。また、それぞれの実装について実行速度とファイルサイズのオーバーヘッドを計測し、実用的なレベルであることを示している。

第6章「実行点の応用：デバッグパターンの例」では、デバッグ作業における典型的な3種類の処理を示し、利用例とともに実行点を用いた実装方法について述べている。

第7章「マルチスレッドプログラムへの応用」では、再現性のないプログラムの例としてJavaのマルチスレッドプログラムを挙げ、スレッド切替のタイミングを記録、再生することで前章のデバッグパターンが適用可能であると論じている。また、Javaの持つデバッガインタフェースを用いて記録と再生が制限つきながらポータブルに実現できることを示し、デバッグパターンの応用範囲が十分広いことを述べている。

第8章「各種の議論」では、第4章で示した実行点の表現方法について、他の3種類の表現方法との相違を論じ、行番号とタイムスタンプによる表現が適切であることを示している。また、自動デバッグやプログラム変換に関する先行研究との比較を行い、本研究の位置付けを述べている。

第9章「結論」では、本研究の成果をまとめ、その意義と今後の展望を述べている。

以上これを要するに、本論文ではソフトウェアの信頼性向上のために重要となるデバッグについて、これまでのようなプログラマ個人の能力のみに依存した方法ではなく、プログラマ間でその知識と経験を共有し、より効率的なデバッグを行うための方法を提案し、実用的なオーバーヘッドでの実装を通して有効性を証明しており、その成果は情報工学上貢献するところが少なくない。

よって本論文は博士（情報理工学）の学位請求論文として合格と認められる。