

博士論文審査結果の要旨

博士論文審査委員会

主 査 松浦 佐江子

審査委員 長谷川 浩志

審査委員 新津 善弘

審査委員 野田 夏子

審査委員 櫛山 淳雄

氏 名	青木 善貴
論文題目	ソフトウェアの品質向上のための開発現場における モデル検査技術を用いたソースコード検証の研究
〔論文審査の要旨〕 ソフトウェア開発の現場では不十分な要件定義やシステムの実装時のミス・誤解等により大小多くの不具合が発生し、その原因調査や修正に多くの工数を費やす。開発現場では、リリースの決定やシステムのマイグレーション時に、ソースコードに対して、テストだけでは発見しにくい問題を効率よく発見し、そのシステムが仕様を満たしていることを検証する方法が求められている。形式手法の 1 つであり、システムの振る舞いがある性質を満たすか否かを網羅的な探索により自動的に検証する方法であるモデル検査は有望な手段であるが、ソースコードをそのままモデル検査用のモデルに変換すると検査ができなくなる、検査式を定義することが困難になるという問題がある。これらの問題を解決するとともに、開発現場への導入の負荷を軽減することが必要である。本研究では、ソースコードをシステムの機能単位と対応付けた段階的変換方法により検査可能にするとともに、仕様を機能に対してその実行条件・状態を組み合わせたデシジョンテーブルにより表現して検査する方法を定義した。さらに、直接モデル検査ツールを操作せずに、検査を支援するツールを用いることにより、ソースコードがその仕様を満たすことを開発現場の技術者が検証できる方法を提案した。結論として、実際に開発現場で生じた不具合の原因の特定および、仕様から作成されたプログラムがその仕様を満たしているかといった、実際に仕様からプログラムを作成する際に作り込まれた不具合の発見と原因の特定を行うことができた。 審査は 1 月 29 日（木）13 時から行われ、1 時間のプレゼンテーション、1 時間の質疑応答を実施した。博士論文の構成に従い、研究の背景と目的、関連研究との比較による解決すべき問題点、提案手法、事例による評価について、発表が行われた。第一著者として、学術論文誌 1 件、査読付き国際会議件 6 件、査読付き国内会議 4 件と十分な研究成果が得られており、プレゼンテーションも、新規性のアピールが不足するものの、整理してわかりやすく発表された。今後は、開発現場で適用するための提案手法の前提、適用範囲等の条件をよりわかりやすく説明する必要があるとの指摘があった。投票を行い、委員全員一致で合格となった。	

<p>※報告番号</p>	<p>甲 第 174号</p>	<p>氏 名</p>	<p>青木 善貴</p>
<p>主論文題名</p> <p>ソフトウェアの品質向上のための開発現場におけるモデル検査技術を用いたソースコード検証の研究</p>			
<p>内容の要旨</p> <p>開発現場では不十分な要件定義やシステムの実装時のミス・誤解等により大小多くの不具合が発生する。これらの不具合を解消するために多くの工数が費やされ、プロジェクトの進行が妨げられる。解消するために特に多大な工数を要する不具合が発生する。これは要件定義段階のような上流工程で不具合の原因が発生する場合と実装段階のような下流工程で不具合の原因が発生する場合がある。仕様の誤解や考慮不足等を上流工程において早期に発見することも企業にとって重要である。不十分な要件定義に対処するために、上流工程においても、客観的指標による早期の検証を行うことが必要である。</p> <p>ある程度プロジェクトが進行した下流工程で発生する手戻りは、作業を再度やり直すことになるため工数の増大とスケジュールの圧迫を招く。大きな手戻りを発生させる不具合の原因は、システムで実現した機能と要求仕様に記述した機能との間に、小規模な修正では解消できない機能差が生じていることである。これは仕様に対するプログラムの整合性の検証が不十分な場合に発生すると考えられる。従ってこれを防ぐためには、要求仕様に対するプログラムの整合性を早期にもれなく検証する仕組みが必要である。</p> <p>また、不具合の解消にはまず原因を特定して再現確認することが前提であるが、原因の特定が困難な不具合は再現確認できないため調査工数がかかる。通常不具合の解消にかけられる時間には制約があるため、調査が不十分なまま修正を行ってしまうと、完全な修正ではないため不具合の再発を招き再度調査・修正の繰り返しになり多大な工数がかかることになる。不具合に対応する開発者が、不具合の現象からその原因を想定できない場合、このような原因特定が困難な状態になる。従ってこのような状態は単純なコーディングのミス等ではなく、複雑な条件の組合せや同期・タイミングの問題で発生することがほとんどであり、仕様に記述された業務機能の振舞いと実際に作成されたプログラムにより実現された業務機能の振舞いの間で整合性が取れないことを示している。このような不一致の発見は、通常は単体・結合テストにより行うがテストケースが不十分な場合は見逃されてしまうため、もれなく業務機能の振舞いの整合性を検証する仕組みが重要である。</p>			

<p>※ 報告番号</p>	<p>第 174号</p>	<p>氏 名</p>	<p>青木 善貴</p>
<p>これらの不具合は成果物の仕様についての妥当性・整合性の検査がもれなくできれば発見できると考える。近年、システム開発の上流工程においてシステムが取りうる様々な状態を満たすかを検査するモデル検査が注目されている。モデル検査は状態遷移系として定義されたシステム（検査モデル）に対し、要求する性質を論理式で記述し、状態遷移系がこの論理式を満たすことを検査する手法であり、システムがもたらす機能の振舞いを検査することに適した手法である。</p> <p>本研究は下流工程においてソースコードを検証することを主な目的としており、Java のソースコードに対して開発するシステムがもたらす業務機能の振舞いをモデル検査を用いて「システムがある条件下で正当でない状態に陥ることは決して起こらない」もしくは「特定の状態へ到達することができる」といった性質を検査できるソースコード検証手法を提案する。</p> <p>提案手法では仕様形式化する手法としてデシジョンテーブルのように開発現場の開発者でも適切かつ容易に定義できる形式を利用して、システムを取りうるべき状態の検査モデル(本研究ではこれを仕様モデルと呼ぶ)に変換し、それをソースコードの制御フローに基づき作成した検査モデル（本研究ではこれをソースコードモデルと呼ぶ）と結合して検査する。この検査により仕様とソースコード間の業務機能の振舞いの不一致が発見され、妥当性・整合性が確認できる。この作業を検査者が適切かつ容易に実施できるように検査には支援ツールを用いる。</p> <p>またモデル検査は元々上流工程で仕様検証することを目的としたものである。提案手法をソースコードの代わりに UML モデルへ適用すれば、要件定義の段階において仕様として記述されているシステムがもたらす業務機能の振舞いが想定外にならないことの検証ができると考えられるため、UML モデルへの適用についても検討する。これを UML モデル検証手法として提案する。</p> <p>本研究の成果としては、モデル検査の専門的な知識を持たない開発者でも容易にモデル検査技術を用いて、不具合の原因の特定を安定した工数で行えるようにすることによりシステム開発の開発現場で広くプロジェクトの品質や作業効率の向上が望める。また、経験を重視する傾向が強い開発現場においてソフトウェア工学の考え方を導入することにより経験的なアプローチだけでなく、工学的なアプローチが行われる土壌が養われることが期待される。</p>			