様　式　Ｃ－１９、Ｆ－１９、Ｚ－１９　（共通）

科学研究費助成事業　　研究成果報告書

平成 ２７ 年　 ６ 月　 １ 日現在

機関番号：１３３０２
研究種目：基盤研究(C)
研究期間：2012～2014
課題番号：２４５０００３５

研究課題名（和文）形式手法の統合によるシームレスなソフトウェア開発手法の提案

研究課題名（英文）Integration of Formal Methods for Seamless Software Developments

研究代表者
青木　利晃（Aoki, Toshiaki）

北陸先端科学技術大学院大学・情報科学研究科・准教授

研究者番号：２０３１３７０２

交付決定額（研究期間全体）：（直接経費）　　4,100,000円

研究成果の概要（和文）：本研究では，複数の形式手法を統合し，システム開発の上流工程から下流工程までをシームレスに接続する手法を提案した．また，実際の車載オペレーティングシステムの事例に適用し，提案手法の有効性を示すことができた．車載オペレーティングシステムなどの組込みシステムでは，開発工程の一部に形式手法を適用し，検証を行うことが主流であった．一方で，我々は，仕様記述から実装のテストまで，シームレスに接続し，全行程をカバーすることができた．これにより，産業界における形式手法の採用が加速され，ソフトウェアの信頼性，安全性が向上することを期待している．

研究成果の概要（英文）：In this research, we proposed a method to integrate multiple formal methods to cover the whole of system development phases consisting of formal specifications, designs and implementations. In addition, we succeeded in applying the proposed method to the verification of a practical automotive operating system and showing its effectiveness. Formal methods are usually used in a part of the development phases for embedded systems like automotive operating systems, however; in our approach, we suceeded in covering the whole of the phases. The automotive operating system that we verified is a practical one. By showing the fact that formal methods could be successfully applied to the practical system, we expect that adopting formal methods in industries is accelarated and reliability and safety of systems are improved more and more.

研究分野：ソフトウェア工学

キーワード：形式手法　モデル検査　形式仕様記述　テスト　車載ソフトウェア

１．研究開始当初の背景

近年，ソフトウェアの信頼性や安全性に関する問題が深刻になりつつあり，我々の日常生活にも影響が生じるケースが見受けられるようになってきた．この問題を解決するために形式手法が注目されている．形式手法では、数学や論理学を基礎とした言語やツールを用いて、対象となるソフトウェアを記述し，検証を行う．そのため，厳密にソフトウェア開発を進めることができ，ツールによる検証の自動化により，効率的な品質の向上が見込めるのである．しかしながら，現状では，形式手法は，ソフトウェア開発の限定された部分や工程にしか適用されておらず，上流工程（仕様分析）から下流工程(実装)まで適用することは困難である．そこで，本研究課題では，それぞれ適用工程が異なる形式仕様記述，設計検証，実証検証の３つの形式手法を統合してソフトウェア開発を行う手法を提案する．

２．研究の目的

本研究課題の目的は，それぞれ適用工程が異なる形式仕様記述，設計検証，実証検証の３つの形式手法を統合してソフトウェア開発を行う手法を提案することである．提案する手法は，大きく２つに分けられる．

(1)形式仕様記述と設計検証を統合する手法の提案．設計検証では，開発対象ソフトウェアの実現方法について検討を行い，その内部構造の形式的な記述と検証を行う．ここで，設計検証において検証すべき性質は，仕様に基づいて決められるものである．しかしながら，仕様が曖昧に記述されていると，正しく性質を記述することが困難であり，性質の記述漏れなどが生じる可能性が大きい．そこで，形式仕様記述により厳密に仕様を記述し，設計検証において検証すべき性質を導出する手法，および，それらの間の整合性を保証する手法を提案する．

(2)設計検証と実装検証を統合する手法の提案．検証した設計モデルに基づいて実装する際，設計検証で保証した性質は，実装後も成立していなければならない．よって，検証した結果をソフトウェア実装後も保証する仕組みが必要である．現実的なソフトウェア開発では，実装を手作業で行う場合が多い．そこで，本研究では，手作業で実装されたプログラムを対象に，設計モデルとの整合性を検証する手法を提案する．

以上の 2 つの統合の仕組みを提案することにより，仕様から実装まで，シームレスに形式手法を適用することが可能になる．そして，提案手法を車載オペレーティングシステムの開発に応用し，有効性の評価を行う．

３．研究の方法

本研究は，形式手法を実践的なものにする研究であるため，具体的な題材に基づいて進める必要がある．そこで，これまでの研究で，我々が検証実験などを行ってきた，OSEK/VDX と呼ばれる車載オペレーティングシステムを題材として用いる．それぞれの提案手法に関して，以下の方法により研究を行う．

(1)形式仕様記述と設計検証を統合する手法の提案．形式仕様記述言語としては Event-B を用いる．また，これまでの研究により，題材である車載オペレーティングシステムの設計モデルは構築済みであり，本研究でも，これを用いることにする．この設計モデルは Promela と呼ばれる記述言語で書かれており，モデル検査ツール Spin で検証することができる．そして，これらの題材を元に，形式仕様と設計モデルの関係について分析を行い，それらを統合して検証する手法を提案する．

(2)設計検証と実装検証を統合する手法の提案．設計モデルは(1)と同様，Promela で記述されたものを用いる．実装は，ルネサスエレクトロニクスで開発されている車載オペレーティングシステムを用いる．そして，テストケースを自動生成し，テストにより実装の検証を行う．

４．研究成果

本研究課題の研究期間において研究を実施し，以下の成果を獲得することができた．

(1)形式仕様を用いて設計モデルを検証する手法の提案．Event-B で作成された形式仕様を用いて Promela で作成された設計モデルを検証する手法を提案した．Promela で作成された設計モデルは，モデル検査ツール SPIN により検査される．この際，時相論理や表明などにより，性質を記述する．しかしながら，実践的には，時相論理や表明による仕様の表現は限定的にならざるをえない．そこで，Promela で作成された設計モデルが，Event-B で作成された形式仕様を満たしているかどうか検証する手法を以下の手順で提案した．

①状態遷移システム基づいた形式化：提案した手法では，Event-B で作成された形式仕様と Promela で作成された設計モデルの間に模倣関係が存在することを検証する．一方で，Event-B と Promela では，記述方式が異なる．そこで，状態遷移システムに基づいて，Event-B と Promela による記述を形式的に表現し，それらの間の模倣関係を定義した．

②有界性の定義：Event-B で作成された形式仕様は，潜在的に，無限の状態を持ち得る．一方で，Promela による記述は，有限状態に限られている．そこで，前者を有限状態に限定する境界(Bounds)を定義した．

③環境の生成：①と②の定義に基づいて，模倣関係を検査するための状態遷移モデルを定義した．この状態遷移モデルのことを環境と呼ぶ．また，Event-B で作成された形式仕様から Promela で記述された環境を生成するアルゴリズムを提案した．生成された環境と設計モデルを組み合わせることにより，モデ

ル検査ツール SPIN で模倣関係の検査を行うことができる．
④ツールの実装と評価．提案手法に基づいて設計モデルを検証するツールを実装した．実装したツールは，形式仕様記述言語 Event-B による記述と境界(Bounds)を入力とし，提案したアルゴリズムに基づいて，設計モデルを検査するための Promela 記述を自動的に出力する．出力された Promela 記述と同じく Promela により記述された設計モデルを組み合わせ，モデル検査ツール Spin で自動的に検査することにより，仕様と設計の整合性を自動的に検証することができる．また，このツールを，車載オペレーティングシステムの事例に適用し，評価を行った．これにより，提案手法とツールは，境界の設定により状態爆発問題を回避しつつ，十分な誤り検出能力を持つことを示すことができた．
(2)設計モデルを用いて実装を検証する手法の提案．Promela で記述された設計モデルを用いて，テストケースを自動生成し，実装の検証を行う手法を提案した．実装の正しさを確認するためには，実装自体を実際に実行して動作を確認することが望ましいと考えている．そこで，上記(1)で十分に検証された設計モデルに基づいて，実装を網羅的にテストする以下のような手法を提案した．
①モデル検査アルゴリズムを用いたテストケースの自動生成．設計モデルをオートマトンとみなし，従来から研究が行われている，オートマトンに基づいた整合テストの枠組みを用いることにした．この手法では，基本的には，オートマトンの状態を探索し，そのすべての状態に到達するテストケースを生成する．従来の整合テストでは，実装を完全にブラックボックスとして扱っているが，提案手法では，設計モデルの状態と実装の状態の比較を行うことにした．これは，設計モデルに詳細な計算が記述されているため，それを有効に活用するためである．また，SPIN を用いて，効率的に状態を探索し，テストケースを生成するツールも作成した．
②コンピュータクラスタを用いた大規模なテスト実施．①で提案した手法では，莫大な数のテストケースが生成される．そこで，コンピュータクラスタを用いて，並列にテストを実施する仕組みを提案した．それぞれのテストケースは独立に実行できるので，複数のコンピュータに分散して実行し，その結果を一ヶ所のストレージに格納するのである．コンピュータクラスタとしては，北陸 StarBED 技術センターのコンピュータクラスタを用いた．ここでは，SpringOS と呼ばれる，コンピュータクラスタを柔軟に管理する仕組みが提供されているので，それらを組み合わせて，大規模なテストを実施する環境を作成した．
③提案手法の評価．①,②で提案した手法を用いて車載オペレーティングシステムの事例に適用し，評価を行った．生成したテストケースは百万件を越えており，当初の見積りでは，一台の PC では十数年かかる見積りであった．そこで，2 のコンピュータクラスタを用いて，70 台の PC（1680 コア）を用いて並列にテストを実施した．その結果，約１１日でテストを完了することができた．テストを用いた網羅的な検証は時間的に不可能だと考えられている．もちろん，すべての場合を尽くすのは不可能ではあるが，絞り込んだ範囲ではあるが，何が起きるか自明ではない数のテストケースを用いて網羅的にテストすることは可能であることを示すことができた．

研究期間全体を通しては，研究は，おおむね順調に進んだと考えている．当初の予定どおり，複数の形式手法を統合し，ソフトウェア開発の上流工程から下流工程までをシームレスに接続した．さらには，実際の車載オペレーティングシステムの事例に適用し，提案手法の有効性を示すことができた．本研究の意義は，形式手法に基づいて，全行程をカバーすることができたことである．我々が注目している題材の車載オペレーティングシステムでは，開発工程の一部に形式手法を適用し，検証した事例のみであった．一方で，我々は，仕様記述から実装のテストまで，シームレスに接続し，全行程をカバーすることができた．これは，大きな意義のある成果であると言える．また，我々が検証している車載オペレーティングシステムは，実際に世の中で使われているものであり，それを対象として提案手法の有効性を示すことができた．これにより，産業界における形式手法の採用が加速され，ソフトウェアの信頼性，安全性が向上することを期待している．

５．主な発表論文等
〔雑誌論文〕（計５件）
1. Dieu-Huong Vu, Yuki Chiba, Kenro Yatake, and Toshiaki Aoki, A Framework for Verifying the Conformance of Design to Its Formal Specifications, IEICE Transactions, 査読有, Vol.E98-D,No.6, 2015, 採録決定.
2. Warawoot Pacharoen, Toshiaki Aoki, Pattarasinee Bhattarakosol, and Athasit Surarerks, Active Learning of Nondeterministic Finite State Machines, Mathematical Problems in Engineering, 査読有, vol. 2013, Article ID 373265, 2013, 11 pages(DOI:10.1155/2013/373265).
3. Hsin-Hung Lin, Toshiaki Aoki, and Takuya Katayama, Automated Adaptor Generation for Behavioral Mismatching Services Based on Pushdown Model Checking, IEICE Transactions, 査読有, Vol.E95-D, No.7, 2012, pp.1882-1893.
4. 矢竹健朗, 青木利晃, UML に基づく RTOS 設計検証のための環境自動生成法, 日本ソフトウェア科学会 学会誌 コンピュータソフトウェア, 査読有, Vo.29, No.3, 2012,

pp. 121-142.
5. Pham Ngoc Hung, Viet Ha Nguyen, Toshiaki Aoki, and Takuya Katayama, On Optimization of Minimized Assumption Generation Method for Component-Based Software Verification, IEICE Transactions, 査読有, Vol. 95-A, No. 9, 2012, pp. 1451-1460.

〔学会発表〕(計 21 件)
1. Toshiaki Aoki, Large scale testing using computer clusters, Static Analysis meets Runtime Verification, 2015 年 3 月 16 日～19 日, 湘南国際村センター(神奈川県・葉山町).
2. 青木利晃, 佐藤信, 谷充弘 , 矢竹健朗, 岸知二, 車載オペレーティングシステムを対象としたモデル検査とテストによる正しさの確信手法, 第 21 回ソフトウェア工学の基礎ワークショップ(ポスター発表), 2014 年 12 月 11 日～13 日, 霧島国際ホテル(鹿児島県・霧島市).
3. 青木利晃, 千葉勇輝, 松原正裕, 西昌能, 成沢文雄, ISO26262 における安全仕様のゴール木を用いた浅い形式化, 第 21 回ソフトウェア工学の基礎ワークショップ(ポスター発表), 2014 年 12 月 11 日～13 日, 霧島国際ホテル(鹿児島県・霧島市).
4. Toshiaki Aoki, Creating Confidence in the Correctness with formal methods and testing, Integration of Formal Method and Testing for Model-Based Systems Engineering, 2014 年 12 月 1 日～3 日, 湘南国際村センター(神奈川県・葉山町).
5. Haitao Zhang, Toshiaki Aoki, and Yuki Chiba, A Spin-based Approach for Checking OSEK/VDX Applications, Third International Workshop on Formal Techniques for Safety-Critical Systems, 2014 年 11 月 6 日～7 日, ルクセンブルグ(ルクセンブルグ).
6. Dieu-Huong Vu, Yuki Chiba, Kenro Yatake, and Toshiaki Aoki, Checking the Conformance of a Promela Design to Its Formal Specification in Event-B, Third International Workshop on Formal Techniques for Safety-Critical Systems, 2014 年 11 月 6 日～7 日, ルクセンブルグ(ルクセンブルグ).
7. Toshiaki Aoki, Practical Application of Formal Methods to Automotive Systems, Science and Practice of Engineering Trustworthy Cyber-Physical Systems, 2014 年 10 月 26 日～30 日, 湘南国際村センター(神奈川県・葉山町).
8. Hideto Ogawa, Makoto Ichii, Fumihiko Kumeno and Toshiaki Aoki, A Practical Study of Debugging using Model Checking, Industry Track, 20th Asia-Pacific Software Engineering Conference, 2013 年 12 月 2 日～5 日, バンコク(タイ).
9. Kriangkrai Traichaiyaporn and Toshiaki Aoki, Preserving correctness of requirements evolution through refinement in Event-B, 20th Asia-Pacific Software Engineering Conference, 2013 年 12 月 2 日～5 日, バンコク(タイ).
10. Haitao Zhang, Toshiaki Aoki, Hsin-Hung Lin, Min Zhang, Yuki Chiba and Kenro Yatake, SMT-based Bounded Model Checking for OSEK/VDX Applications, 20th Asia-Pacific Software Engineering Conference, 2013 年 12 月 2 日～5 日, バンコク(タイ).
11. 小川秀人, 市井誠, 粂野文洋, 青木利晃, POM/MC を用いた仮説ベースモデル検査デバッグ手法, 第 20 回ソフトウェア工学の基礎ワークショップ, 2013 年 11 月 28 日～30 日, ゆのくに天祥(石川県・加賀市).
12. Xiaoyun Guo, Hsin-Hung Lin, Kenro Yatake and Toshiaki Aoki, An UPPAAL Framework for Model Checking Automotive Systems with FlexRay Protocol, Second International Workshop on Formal Techniques for Safety-Critical Systems, 2013 年 10 月 29 日, クイーンズタウン(ニュージーランド).
13. Kriangkrai Traichaiyaporn and Toshiaki Aoki, Refinement Tree and Its Patterns: a Graphical Approach for Event-B Modeling, Second International Workshop on Formal Techniques for Safety-Critical Systems, 2013 年 10 月 29 日, クイーンズタウン(ニュージーランド).
14. Haitao Zhang, Toshiaki Aoki, Kenro Yatake, Min Zhang and Hsin-Hung Lin, An approach for checking OSEK/VDX applications, The 13th International Conference on Quality Software, 2013 年 7 月 29 日～30 日, 南京(中国).
15. Kenji Taguchi, Hideaki Nishihara, Toshiaki Aoki, Fumihiro Kumeno, Koji Hayamizu, Koichi Sinozaki, Building A Body of Knowledge on Model Checking for Software Development, Proceedings of Annual International Computers, Software & Applications Conference, pp. 784-789, 2013 年 7 月 22 日～26 日, (京都府・京都市).
16. Shinji Kikuchi and Toshiaki Aoki, Evaluation of Operational Vulnerability in Cloud Service Management using Model Checking, International Symposium on Service-Oriented System Engineering(SOSE), 2013 年 3 月 25 日～28 日, レッドウッドシティ(アメリカ).
17. 陳適, 青木利晃, モデル検査ツールにより出力された反例に基づく誤り特定手法, 第 19 回ソフトウェア工学の基礎ワークショップ, 2012 年 12 月 13 日～15 日, ゆふいん山水館(大分県・由布市).
18. 青木利晃, 佐藤信, 谷充弘, 矢竹健朗, モデル検査とテストによる車載オペレーテ

ィングシステムのシームレスな検証，組込みシステムシンポジウム，2012 年 10 月 16 日～19 日，国立オリンピック記念青少年センター(東京都・渋谷区).
19. Dieu-Huong Vu, Toshiaki Aoki, Faithfully Formalizing OSEK/VDX Operating System Specification, International Symposium on Information and Communication Technology, 2012 年 8 月 23 日～24 日，ハロンベイ(ベトナム).
20. Kenro Yatake, Toshiaki Aoki, SMT-based Enumeration of Object Graphs from UML class diagrams, International workshop UML and Formal Methods, 2012 年 8 月 27 日，パリ(フランス).
21. Hiroaki Tanizaki, Toshiaki Aoki and Takuya Katayama, A Variability Management Method for Software Configuration Files, The 24th International Conference on Software Engineering and Knowledge Engineering, 2012 年 7 月 1 日～3 日，レッドウッドシティ(アメリカ).

〔図書〕(計 1 件)
1. Toshiaki Aoki, Kenji Taguchi, Springer, Formal Methods and Software Engineering - 14th International Conference on Formal Engineering Methods, 2012 年, 528 ページ.

〔産業財産権〕
○出願状況（計 0 件）

名称：
発明者：
権利者：
種類：
番号：
出願年月日：
国内外の別：

○取得状況（計 0 件）

名称：
発明者：
権利者：
種類：
番号：
出願年月日：
取得年月日：
国内外の別：

〔その他〕
ホームページ等

６．研究組織)
(1)研究代表者
青木　利晃（AOKI TOSHIAKI）
北陸先端科学技術大学院大学・情報科学研究科・准教授
研究者番号：20313702