様　式　Ｃ－１９、Ｆ－１９－１、Ｚ－１９　（共通）

科学研究費助成事業　　研究成果報告書

平成　３０　年　　５　月　１５　日現在

機関番号：　１３３０２
研究種目：　基盤研究(C)（一般）
研究期間：　2015～2017
課題番号：　１５Ｋ００９４

研究課題名（和文）次世代車載オペレーティングシステムにおける先進機能の形式検証に関する研究

研究課題名（英文）Formal verification of advanced functionalities in next-generation automotive operating systems

研究代表者

青木　利晃（Toshiaki, Aoki）

北陸先端科学技術大学院大学・先端科学技術研究科・教授

研究者番号：２０３１３７０２

交付決定額（研究期間全体）：（直接経費）　　3,500,000円

研究成果の概要（和文）：本研究では，AUTOSAR OSの先進機能を形式検証する手法を提案した．AUTOSAR OSでは，次世代の自動車を見据えて，保護機能とマルチコア機能が提供されている．そこで，これらの機能を実践的に形式検証する手法を提案した．保護機能の形式検証では，AUTOSAR OSの仕様書に基づいて形式仕様を作成し，定理証明による検証過程において，仕様の矛盾を発見することに成功した．マルチコア機能の検証では，複数のメモリモデルに基づいてプログラムを自動定理証明により自動検証する手法を提案し，Linux，TOPPERS/FMP化ネールなどで用いられているspinlockプログラムの形式検証に成功した．

研究成果の概要（英文）：We proposed methods to formally verify advanced functions of AUTOSAR operating system. Protection function and multicore function are provided for next generation cars in AUTOSAR operating system. Thus, we proposed practical methods to formally verify those functions. In the formal verification of the protection function, its formal specification was developed based on the specification of AUTOSAR operating system and we succeeded in finding the inconsistency of the AUTOSAR operating system specification during proving the consistency of the specification by theorem proving. In the formal verification of the multicore function, we proposed a method to automatically verify programs by automated theorem proving based on multiple memory models, and successfully verified spinlock programs of real operating systems such as Linux and TOPPERS/FMP.

研究分野：　ソフトウェア工学

キーワード：　形式手法　形式検証　車載システム　形式仕様　定理証明

１．研究開始当初の背景

車載ソフトウェアの安全性や信頼性に関する問題は，社会において非常に大きな関心となりつつある．自動車は，従来は機械的に制御されてきたが，近年，コンピュータ制御技術の発展と利便性や性能の追求により，多くの部品の電子化が進んでいる．これにより，車載ソフトウェアの規模の急速な増大と複雑化がもたらされ，電子制御部分の安全性と信頼性に関する問題が取り上げられつつある．世界標準においては，機能安全に関する標準が一般の電子システムだけでなく，車載ソフトウェアに特化されたものが策定されている．また，実社会においては，自動車のリコールが多発しており，最も注目されたのは，2010 年に発生したトヨタ車の急加速問題である．この問題では，電子スロットル制御システムの検証が NHTSA と NASA により実施された．我々は，このような車載ソフトウェアの安全性や信頼性の問題を背景に，車載オペレーティングシステム(以下，オペレーティングシステムを OS と略す)の検証手法の研究と実践を行っている．

OS は車載ソフトウェアの基盤であり，安全性の評価の際，非常に重要な位置づけとなる．我々は，これまでに，OSEK/VDX と呼ばれる国際標準に基づいた車載 OS の検証を行ってきた．現在，OSEK/VDX の活動は AUTOSAR に引き継がれ，近年，新たな車載 OS の国際標準 AUTOSAR OS が策定されている．AUTOSAR OS では，タスクの管理機能は OSEK/VDX のものを採用している．新たに追加された機能は，主に，保護機能とマルチコア向け機能である．保護機能は，複数のアプリケーションを同一の ECU(Electronic Control Unit)で動作させる際，重要なアプリケーションとそうでないものを分離させるものである．これは，ECU の性能の向上により，導入された機能である．また，マルチコア向け機能は，ECU のマルチコア化を見据え，導入されている．

一方で，AUTOSAR OS の仕様書は多くが自然言語で記述されており，記述の統一性がなく煩雑で不明瞭である．また，非常に抽象的に記述されているため，OS の実装をイメージすることが難しい．さらに，OSEK/VDX のプロセス管理を前提として，その差分のみを規定しているため，機能の一貫性の保証が困難である．このように，AUTOSAR OS は，様々な問題を抱えているが，次世代の車載ソフトウェアの基本ソフトウェアであることを考えると，その安全性と信頼性を担保する方式を提案することは，非常に重要である．そこで，本研究課題では，形式手法/検証により，AUTOSAR OS の正しさを保証する手法を提案する．形式手法/検証では，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．これにより，高い安全性と信頼性を達成することができると期待されている．

２．研究の目的

本研究課題では，次世代車載オペレーティングシステム AUTOSAR OS の先進機能を形式検証する方式を提案する．前述したが，AUTOSAR OS の仕様書は非形式的に記述されており，様々な問題を抱えている．そこで，まず，仕様書の形式化を行う必要がある(以下の 1)．そして，その仕様書に基づいて，AUTOSAR OS の実装を検証する手法を提案する(以下の 2)．

(1) AUTOSAR OS 仕様の形式化

AUTOSAR OS における先進機能である保護機能の仕様を形式仕様記述言語で記述し，形式化する．形式仕様記言語としては Event-B を用いる．Event-B は，集合と関数に基づいて仕様を記述する．保護機能は，タスクや資源などの OS オブジェクトの集合間の制約として表現する．Event-B では，RODIN と呼ばれるツールが提供されており，ツールにより機械的な証明(定理証明)も可能である．これにより，AUTOSAR OS 仕様の曖昧性と不明瞭性を排除することができ，仕様書の品質を向上させることができると考えている．

(2) 実装の形式検証手法の提案

AUTOSAR OS の実装が正しいことを検証する手法を提案する．適用する手法は，形式検証の代表的な技法であるモデル検査と定理証明である．これらの技法は，互いに利点と欠点を持っており，相補的に使うのが望ましい．そこで，実装や仕様の特徴に合わせて，それらを使い分け，組み合わせる手法を提案する．対象は，AUTOSAR OS に基づいたマルチコア上への実装である TOPPERS/FMP である．TOPPERS/FMP は名古屋大学を中心に開発されている AUTOSAR 準拠の OS である．

３．研究の方法

本研究課題は，AUTOSAR OS 仕様の検証とマルチコアプロセッサ向け OS 実装の検証により構成される．それぞれについて，以下の方法により研究を行う．

(1) AUTOSAR OS の仕様の検証

①AUTOSAR OS の仕様書は多くが自然言語で記述されており，記述の統一性がなく煩雑で不明瞭である．そこで，本研究課題では，形式仕様記述言語 Event-B を用いて，その仕様を記述する．しかしながら，現状の仕様書は，Event-B を用いて直接記述できるくらい整理はされてはいない．よって，形式的に記述する前に，仕様の分析を行い統一された形式で仕様を記述し直し整理する．

②①で整理した内容に基づいて， 形式仕様記述言語 Event-B を用いて，AUTOSAR OS の保護機能の仕様を記述し，形式仕様を作成する．Event-B は欧州で開発されている形式仕様記述言語であり，比較的新しく，メンテナンスも行き届いている．RODIN と呼ばれるツールが提供されており，機械的な証明も行うことができる．

③②で作成した形式仕様が正しいことを定

理証明により検証する．作成した形式仕様が正しいと確信するために，重要な性質を不変表明として記述し，それが成立することを証明する．さらに，形式仕様と現状の AUTOSAROS の仕様書を比較し，不備があれば，それを指摘し，改善案を示す．
(2) マルチコア向け OS 実装の検証
①マルチコアプロセッサ上のプログラムは高度な並行性を持っており，その実行は，ハードウェアの挙動の影響を受ける．そのため，プログラムにかかれている順番で命令が実行されるとは限らず，ハードウェアの挙動を含めた検証が必要となる．そこで，まず，マルチコアプロセッサの挙動を明らかにする．
②①で明らかにした挙動を取り扱う方法について検討する．研究開始後，メモリモデルが重要であることが明らかになった．メモリモデルは，挙動の種類によって，SC，TSO，PSO，WO などの分類がされており，さらに，プロセッサによっては，それらとは少し異なる挙動もある．これらの挙動を取り扱う方法について明らかにする．
③②で明らかにした方法を実現する方式を検討する．並行性を取り扱うには，モデル検査が適しているが，マルチコアプロセッサ向け実装の検証では，従来のモデル検査とは異なる探索をしなければならない．そこで，まず，モデル検査による検証法について検討する．また，モデル検査では，有限の範囲でしか検証が実施できない．よって，定理証明による検証についても検討する．そして，最終的には，TOPPERS/FMP の検証を実施する．

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) AUTOSAR OS 仕様の形式化と検証手法の提案．
AUTOSAR OS 仕様の保護機能の形式仕様を作成し，その正しさを保証する検証を実施した．これにより，AUTOSAR OS 仕様を形式化し，検証する方式を明らかにすることができた．この成果は以下に分類される．
①AUTOSAR OS 仕様の保護機能の形式仕様の獲得．
AUTOSAR OS のオリジナルの文書では，保護機能の仕様が自然言語(英語)で記述されている．その記述には曖昧な表現が多く，一貫して十分なメモリ保護を規定しているか確信を持つことができない．そこで，形式仕様記述言語 Event-B を用いて，オリジナル文書に基づいた，保護機能の形式仕様を作成した．オリジナルの文書では，read/write を行う元の OS コンポーネントと先の OS コンポーネントの関係に基づいて，それを許可するか，禁止するか規定している．また，OS コンポーネントは，所有関係や階層化された構造を持っている．そこで，集合と関係に基づいて OS のコンポーネントの構造を形式化し，その構造に基づいた条件として許可するか禁止するか記述した．これにより，保護機能の記述を明確にすることができた．さらに，この形式化の過程で，オリジナルの仕様における曖昧な箇所を多数指摘することもできた．
②AUTOSAR OS 仕様の形式仕様の定理証明による検証と矛盾の発見．
Event-B では，不変表明に基づいて証明責務と呼ばれる条件が生成され，これらを証明することにより，不変表明が成立することを保証する．よって，保護機能の無矛盾性を保証する不変表明を記述し，証明責務の証明を行った．無矛盾性とは，OS コンポーネント間に異なるアクセス，すなわち，禁止と許可の両方が規定されていることは無いということである．OS コンポーネントは所有関係や階層を持っており，さらに，オリジナルの仕様では，断片的にアクセスを規定しているため，矛盾が存在する可能性を否定できない．そこで，生成された証明責務を定理証明により証明しようとしたところ，証明できないものが存在した．詳細を分析したところ，オリジナルの仕様に矛盾があることを発見した．また，矛盾を修正し，再度，証明責務の証明を試みたところ，すべて証明することができた．これにより，無矛盾で曖昧性が排除された保護機能の形式仕様を獲得することができた．
(2) マルチコア向けプログラムの自動検証手法の提案．
OS の実装では，ハードウェアレベルの挙動は，アセンブリ言語で取り扱われることが多い．また，マルチコアプロセッサの挙動は，採用されているメモリモデルに大きく影響を受ける．そこで，メモリモデルを考慮した，アセンブリプログラムを対象とした以下の自動検証手法を提案した．
①メモリモデルを考慮した自動有界検証手法の提案．
アセンブリプログラムは，構造化されていないが，ジャンプ命令などによる繰り返しを含む．そこで，有界モデル検査と同様のアプローチで，有限ステップに限定して，検証を実施する．アセンブリプログラムの有限ステップの実行は，記号実行により，記号的に獲得する．メモリモデルは，Gharachorloo Framework[1]と Herding Cats Framework[2]に基づいた．そして，自動定理証明(SMT)を用いて，メモリモデルに基づいて可能性のある実行列を獲得し，それらの実行列において，与えられた性質が成立するかどうか自動的に判定する．
②メモリモデルを考慮した自動演繹的検証手法の提案．
①の手法では，有限ステップに限定されているため，動作し続けるアセンブリプログラムでは，部分的な正しさしか保証できない．そこで，構造化されたアセンブリプログラムを対象として，演繹的に正しさを保証する手法を提案した．対象のアセンブリプログラムは，任意の形をしたものではなく，事前に繰り返し部分が構造化されているものとする．そし

て，繰り返し内の命令に基づいて，不変表明を獲得し，その妥当性，および，与えられた性質の正しさを，自動定理証明(SMT)で自動証明する．

また，①②の手法を Linux，TOPPERS/FMP 化ネールなどで用いられている spinlock プログラムに適用し，それらの自動検証に成功した．

以上のように，本研究では，最終的に，実際の AUTOSAR OS の仕様，および，AUTOSAR OS の実装を検証することに成功した．さらに，仕様の検証では，矛盾を指摘することができた．これらのことから，車載 OS における先進機能の実践的な形式検証手法を提案できたと言える．

参考文献

[1] Kourosh Gharachorloo. Memory consistency models for shared-memory multiprocessors. Technical report, Stanford University, Stanford, CA, USA, 1995.

[2] Jade Alglave, Luc Maranget, Michael Tautschnig. Herding Cats: Modelling, Simulation, Testing, and Data Mining for Weak Memory, J ACM Trans. Program. Lang. Syst., pp.0164-0925, Vol. 36, No. 2, 2014.

５．主な発表論文等

（研究代表者、研究分担者及び連携研究者には下線）

〔雑誌論文〕（計 3 件）

1. Min Zhang, Toshiaki Aoki and Yueying He, A spiral process of formalization and verification - A case study on verification of the scheduling mechanism of OSEK/VDX, Journal of Information Security and Applications, 査読有, Vol.31, 2016, pp.41-53.

2. Haitao Zhang, Toshiaki Aoki and Yuki Chiba: Verifying OSEK/VDX Applications, A Sequentialization-Based Model Checking Approach, IEICE Transactions, 査読有, Volume 98-D, No.10, 2015, pp.1765-1776.

3. 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, Volume E98-D No.6, 2015, pp.1137-1149.

〔学会発表〕（計 13 件）

1. Nhat-Hoa Tran, Yuki Chiba and Toshiaki Aoki, Domain-Specific Language Facilitates Scheduling in Model Checking, 24th Asia-Pacific Software Engineering Conference, 2017 年 12 月 4 日～8 日, Nanjing(China)

2. Xiaoyun Guo, Hsin-Hung Lin, Toshiaki Aoki and Yuki Chiba, A Reusable Framework for Modeling and Verifying In-vehicle Networking System in the Presence of CAN and FlexRay, 24th Asia-Pacific Software Engineering Conference, 2017 年 12 月 4 日～8 日, Nanjing(China)

3. Takashi Tomita, Daisuke Ishii, Toru Murakami, Shigeki Takeuchi and Toshiaki Aoki, Template-Based Monte-Carlo Test Generation for Simulink Models, Seventh Workshop on Design, Modeling and Evaluation of Cyber Physical Systems, 2017 年 10 月 19 日, Seoul(Korea)

4. Pattaravut Maleehuan, Yuki Chiba and Toshiaki Aoki, Assembly Program Verification for Multiprocessors with Relaxed Memory Model using SMT Solver, 11th International Symposium on Theoretical Aspects of Software Engineering, 2017 年 9 月 13 日～15 日, Nice(France)

5. Masahiro Matsubara, Fumio Narisawa, Atsuhiro Ohno, Toshiaki Aoki and Yuki Chiba, Dissolution of the Gap between Safety Requirements Written in a Natural Language and Formal Notations, Technical Session of SAE 2016 World Congress and Exhibition, 2016 年 4 月 12 日～14 日, Detroit(USA).

6. Dieu Huong Vu, Yuki Chiba, Kenro Yatake and Toshiaki Aoki, Verifying OSEK/VDX OS Design using Its Formal Specification International Symposium on Theoretical Aspects of Software Engineering, 2016 年 7 月 17～19 日, Shanghai(China).

7. 太田十字光，田辺良則，青木利晃，Java Pathfinder における弱公平性条件の実装，日本ソフトウェア科学会第 33 回全国大会，2016 年 9 月 6 日～9 日，東北大学片平キャンパス(宮城県仙台市)

8. 冨田尭，石井大輔，青木利晃，Simulink モデルに対するテストスイート自動生成，第 14 回ディペンダブルシステムワークショップ，2016 年 12 月 14 日～15 日，花びしホテル(北海道函館市)

9. Haitao Zhang, Toshiaki Aoki and Yuki Chiba, Yes! You Can Use Your Model Checker to Verify OSEK/VDX Applications, International Conference on Software Testing, Verification and Validation, 2015 年 4 月 13 日～17 日, Graz(Austria).

10. Hideto Ogawa, Makoto Ichii, Fumihiro Kumeno and Toshiaki Aoki, Experimental Fault Analysis Process Implemented Using Model Extraction and Model Checking, COMPSAC, 2015 年 7 月 1 日～5 日, Taichung(Taiwan)

11. Toshiaki Aoki, Kriangkrai Traichaiyaporn, Yuki Chiba, Masahiro

Matsubara, Masataka Nishi and Fumio Narisawa, Modeling Safety Requirements of ISO 26262 using Goal Trees and Patterns, International Workshop on Formal Techniques for Safety-Critical Systems, 2015年11月6日～7日, Paris(France)
12. 青木利晃, トライチャイヤポーンクリアンクライ, 千葉勇輝, 松原正裕, 西昌能, 成沢文雄, ゴール木とパターンを用いたISO26262における安全要求のモデル化, 組込みシステムシンポジウム, 2015年10月21日～23日, 早稲田大学グリーン・コンピューティング・システム研究開発センター(東京都新宿区)
13. 青木利晃, 千葉勇輝, 松原正裕, 成沢文雄, ISO 26262 のための安全要求記述言語と追跡可能性検証手法の提案, プログラミングおよびプログラミング言語ワークショップ, 2016年3月7日～9日, ダイヤモンド瀬戸内マリンホテル(岡山県たまの市)

〔図書〕(計1件)
1. Toshiaki Aoki, Makoto Satoh, Mitsuhiro Tani, Kenro Yatake, Tomoji Kishi, Springer, Cyber-Physical System Design from an Architecture Analysis Viewpoint, 2017, 159(109-132).

〔産業財産権〕

○出願状況(計0件)

○取得状況(計0件)

〔その他〕

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・先端科学技術研究科・教授
研究者番号：20313702