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　　６　月　１９　日現在

機関番号：　６２６１５
研究種目：　挑戦的萌芽研究
研究期間：　2016～2017
課題番号：　１６Ｋ１２４１４

研究課題名（和文）想定に漏れた環境変化に耐えるソフトウェアを実現する実行時モデル更新技術

研究課題名（英文）Models@run.time for software tolerant to unforseen changes in the environment

研究代表者
本位田　真一（HONIDEN, Shinichi）

国立情報学研究所・アーキテクチャ科学研究系・教授

研究者番号：７０３３２１５３

交付決定額（研究期間全体）：（直接経費）　　2,600,000円

研究成果の概要（和文）：近年のソフトウェアシステムは環境変化に対して柔軟に動作を継続する自己適応性が求められる．従来の開発手法では，開発時に想定された環境変化に対してのみ保証を伴った適応を可能にする．しかし実行時に直面する環境変化を開発時に想定し尽くすことは困難である．従来の開発手法では想定漏れのリスクは避けられない．そこで本研究では「開発時の想定に漏れた環境変化」に対しても耐えるソフトウェアを実現するための実行時モデル更新技術を提案した．環境・要求・動作仕様モデルをシステムが実行時に保持し，想定から漏れた変化を環境モデルに反映し，安全性を保証する動作仕様をシステム自身によって実用的な速度で導出する技術を構築した．

研究成果の概要（英文）：Modern software systems should be self-adaptive to continue their functionality in response to changes in the environment. Traditional software development assures the self-adaptive software under changes that can be foreseen at development time. However, it is inherently impossible to assume all possible changes at development time, thus this approach does not address unforeseen changes in the environment. This research aims to establish Models@run.time techniques to enable self-adaptive system tolerant to the unforeseen changes. In this approach, a system holds environment, requirement, and behavior specification models even at runtime, reflects changes found at runtime to the models, and makes decision about adaptation to ensure its safety by itself within reasonable computation time.

研究分野：　ソフトウェア工学

キーワード：　自己適応システム　実行時モデリング　環境モデル学習　制御器合成

１．研究開始当初の背景

Cyber-Physical Systems，Internet of Things に代表されるような，**実世界と密に連動することで新たな価値を生み出すソフトウェアシステム**の需要が高まっている．このようなソフトウェアは**動作の影響が実世界にまで及ぶ**ため，**ソフトウェア動作の安全性を保証**する高品質なソフトウェア開発が求められる．

開発の早期段階においてソフトウェア動作の安全性を保証する手法として**モデル検査等の形式的検証**がある．システムの実行環境を分析して環境の状態遷移モデルを構築し，環境モデルが満たすべき安全性を時相論理式等で記述する．加えてソフトウェアの動作仕様も状態遷移モデルとして表し，動作仕様による環境モデルへの影響を網羅的にチェックし，動作仕様の安全性充足を検証，保証する． 例えば，倉庫管理システムの場合，倉庫内の荷物，運搬ロボット，運搬通路等の状態やその状態遷移を環境モデルとして表し，それら環境内の要素を統合，制御するソフトウェア動作仕様モデルによる安全性の充足を検証する．しかしこの方式では，外的要因による運搬経路の封鎖，運搬ロボットの故障等の環境側の状況変化によって**環境モデルに合致しない状態遷移となった場合，安全性は保証されない**．このことは，特に**変化の激しい実世界と密に連動するソフトウェアにおいて顕著な問題**となる．

環境の状況変化を前提とし，柔軟に動作を変更することで安全性保証を継続する自己適応ソフトウェアの開発は，近年，ソフトウェア工学分野における重要な研究課題である．既存研究では，**起こりうる状況変化を開発時に想定し，想定される複数通りの環境モデル毎に要求充足を保証する動作仕様**を準備し，実行時にそれらを選択・切り替えることで想定された状況変化に対する安全性保証の継続を実現している．しかし，既存手法では**開発時の想定に漏れた変化には対応できない**という問題があった．開発時において，実行時に起こりうる変化を全て想定することは本質的に困難であり，想定し尽くそうとすると開発時の工数が飛躍的に増大する．

２．研究の目的

【目的】本研究提案では，**安全性を保証する動作仕様を実行時に，機械的に，実用的な速度で導出する技術を確立**することで，**「開発時の想定から漏れた変化」に対する適応を可能にする自己適応ソフトウェア開発の実現**を目指す．期間内の研究目的は下記である．

[目的 1]環境モデルの実行時更新技術の確立

従来，開発時に人手で行われていた「環境モデルの構築」を自動化し，実行時に取得した情報を過不足なく表すように，**環境モデル(状態遷移モデル)を自動で更新する技術**を確立する．

[目的 2]保証を伴う動作仕様の実行時自動導出技術の確立

従来，開発時に人手で行われていた「安全性を保証する動作仕様の導出」を自動化し，実行時に実用的な速度で実行可能とする技術を確立する．

提案者らは，これまでに，**開発時に，開発者によって行われる環境分析，動作仕様導出を支援する自動化技術**に関する研究を行ってきた．これらの成果を，**実行時に，システムによって，実用的な時間で実行可能にするよう拡張**し，従来研究の問題解決を試みる．

３．研究の方法

研究目的にて挙げた目的 1,2 それぞれの達成のため **[実施項目 1]環境モデル実行時更新技術の確立**，**[実施項目 2]保証を伴う動作仕様の実行時導出技術の確立**を行った．またその技術を統合した**[実施項目 3]評価実験**を行った．**平成 28 年度は各要素技術を構築**した．また，それらの技術を提案者らが以前に開発した自己適応フレームワークに組み込み，**スマートルーム制御を題材とした小規模な例題での評価**を行った．平成 29 年度では，**実用に耐えうる品質となるように各技術を拡張，発展**させた．前年度の評価結果を踏まえ，実行速度，精度等の観点から品質上の課題を洗い出し，各技術の洗練化を行った．また，洗練化された技術を**複数の例題で評価**し，**開発工数と，環境変化に対する頑健性の観点からの評価**を行った．

４．研究成果

実施項目 1: 環境モデルの実行時更新技術の確立

環境モデルを Labeled Transition System (LTS)形式で記述するものとする．環境に対して作用するシステム動作(アクション)毎に，アクションが引き起こしうる環境変化を全て遷移としてモデル化する．想定漏れが起きた場合，実行トレースが示す「実際に起きた環境変化」が環境モデルでは受理できなくなる．本実施項目では，この不整合を解消するため，**実行トレースによって判明した「実際に起きた環境変化」に対して過不足ない遷移をもつ LTS モデルとなるように実行時に更新する手法**を提案した．

環境モデルは，それまでに得られた実行トレースを受理可能でなければならず，また実際には起こらない不要な遷移を含んではならない．環境モデルの実行時更新を行う場合，**実行環境において十分な情報が得られない状況**において，**即応的に，正確な環境モデル構築**することが求められる．

本研究では，即応的に対応すべき「不足する遷移の追加」と，熟考して対応すべき「過度な遷移の削除」という，種類の異なる 2 つの環境モデル学習手法を組み合わせることで即応性と正確性を両立した環境モデル実行時更新手法を実現した．

実施項目 2: 保証を伴う動作仕様の実行時導出技術の確立

実行時に更新された環境モデルに対して，保証を伴う動作仕様を自動で導出する技術を確立する．本実施項目は，従来開発時に用いられていた **Discrete Controller Synthesis 技術**を実行時に応用することで実施する．Discrete Controller Synthesis は，LTS 形式で記述された環境モデルと，Fluent Linear Temporal Logic (FLTL)形式で記述された要求を入力とし，与えられた環境モデル下で要求充足が保証された LTS 形式のシステム動作仕様をゲーム理論に基づき自動で導出する技術である．この **Discrete Controller Synthesis 技術を，実行時の自己適応実現のために利用可能とするよう拡張，発展**させ，**動作仕様の実行時自動導出**を実現した．

環境変化に対して実行時に適応するためには，動作仕様の導出を高速に実行可能でなければならない．しかしながら，従来の Discrete Controller Synthesis は環境モデルの空間探索を一から行うため**実行時に用いる場合，実用に耐える実行速度ではない**．

本研究では，実行時に更新された**環境モデルの差分に着目して効率化を試みた**．まず，環境モデル内の差分を分析し，生じた変化が影響する箇所の部分的な再探索のみで動作仕様を導出することで効率的に導出する．また，典型的な**環境モデルの変化に対する動作仕様変更を**パターン化し，事前定義された**変更パターンを適用，組み合わせる**ことでさらなる効率化を試みた．加えて，システムが満たすべき安全性要求に対して，**各分析で行われた結果を再利用する手法**を提案した．これにより個別に分析する場合に起こる処理の重複を避けることが可能となる．

これらの手法により，従来の分析結果と比較し，分析の速度を 100 倍以上高速化することに成功した．

実施項目 3. 評価実験 (担当：本位田，鄭)

実施項目 1,2 で構築した技術を反映した自己適応ソフトウェアフレームワークを開発した．本位田が以前に開発した自己適応ソフトウェアフレームワークを拡張し，拡張した Discrete Controller Synthesis 技術を実装したソルバに加え，実行中に得られた実行トレースから LTS 形式で記述された環境モデルを更新する環境モデル更新器を開発し，実現した．

開発したフレームワークを用いて，自動倉庫管理システム，自動清掃システムという 2 種類のスマートシステムを例題とした評価実験を行った．その結果，環境モデル学習，動作仕様生成それぞれにおいて従来の手法と同程度の精度，安全性のモデルを 100 倍以上早く生成することが確認できた．これにより従来技術の単純な適用では数十分かかっていた実行時の処理を数秒という実用的な速度で実行することが可能となることを確認した．

５．主な発表論文等

〔雑誌論文〕（計 1 件）

① Ehsan Ullah Warriach, and Kenji Tei: A Comparative Analysis of Machine Learning Algorithms for Faults Detection in Wireless Sensor Networks, International Journal of Sensor Networks (IJSNet), Vol.24, No.1, pp.1-13, 2017 年 5 月，査読有

〔学会発表〕（計 11 件）

① Shinnosuke Saruwatari, Fuyuki Ishikawa, Tsutomu Kobayashi and Shinichi Honiden: Extracting Traceability between Predicates in Event-B Refinement, The 24th Asia-Pacific Software Engineering Conference (APSEC 2017), pp.61-70, 2017 年

② Daichi Morita, Fuyuki Ishikawa and Shinichi Honiden: Construction of Abstract State Graphs for Understanding Event-B Models, 3rd Symposium on Dependable Software Engineering: Theories, Tools and Applications 2017 (SETTA 2017), pp.250-265, 2017 年

③ Aurélien Vialon, Kenji Tei and Samir Aknine: Soft-Goal Approximation Context Awareness of Goal-driven Self-Adaptive Systems, The 2nd International Workshop on Models@run.time for Self-aware Computing Systems at ICAC 2017, pp.233-238, 2017 年

④ Moeka Tanabe, Kenji Tei, Yoshiaki Fukazawa, and Shinichi Honiden: Learning Environment Model at Runtime for Self-Adaptive Systems, the 32nd ACM SIGAPP Symposium On Applied Computing (SAC2017), pp.1198-1204, 2017 年

⑤ Shunichiro Suenaga, Kenji Tei and Shinichi Honiden: Applicability of Earned Value Management for Deadline Energy Constrained Applications, 2017 IEEE International Conference on Industrial Engineering and Engineering Management

(IEEM2017), pp.691-695, 2017 年

⑥ Tsutomu Kobayashi, Fuyuki Ishikawa, and Shinichi Honiden: Refactoring Refinement Structures of Event-B Machines, The 21st International Symposium on Formal Methods (FM 2016), pp.444-459, 2016 年

⑦ Tsutomu Kobayashi, Fuyuki Ishikawa, and Shinichi Honiden: Stepwise Refinement of Software Development Problem Analysis, The 35th International Conference on Conceptual Modeling (ER 2016), pp.488-495, 2016 年

⑧ Moeka Tanabe, Kenji Tei, Yoshiaki Fukazawa and Shinichi Honiden: **自己適応システムのための実行時環境モデル学習手法**, **合同エージェントワークショップ＆シンポジウム** 2016 (JAWS2016), pp.1-8, 2016 **年**

⑨ Masaki Katae, Kenji Tei, Yoshiaki Fukazawa and Shinichi Honiden: **階層離散制御器合成によるマルチロボットシステムの仕様生成手法**, **合同エージェントワークショップ＆シンポジウム** 2016 (JAWS2016), pp.1-8, 2016 **年**

⑩ Takaya Saeki, Fuyuki Ishikawa, Shinichi Honiden: Automatic Generation of Potentially Pathological Instances for Validating Alloy Models, International Conference on Formal Engineering Methods (ICFEM 2016), pp.41-56, 2016 年

⑪ Leandro Nahabedian, Victor Braberman, Nicolas D'Ippolito, Shinichi Honiden, Jeff Kramer, Kenji Tei and Sebastian Uchitel: Assured and Correct Dynamic Update of Controllers, 11th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS2016), pp.96-107, 2016 年

〔図書〕（計　0　件）

〔産業財産権〕

○出願状況（計　0　件）

○取得状況（計　0　件）

〔その他〕
ホームページ等

６．研究組織
(1)研究代表者
本位田 真一（HONIDEN, Shinichi）
国立情報学研究所 アーキテクチャ科学研究系，教授
研究者番号：70332153

(2)研究分担者
鄭 顕志（TEI, Kenji）
国立情報学研究所 アーキテクチャ科学研究系，准教授
研究者番号： 40434295

(3)連携研究者
なし

(4)研究協力者
なし