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 １７ 日現在

機関番号：１２６０８
研究種目：基盤研究(C)
研究期間：2012～2014
課題番号：２４５０００３３

研究課題名（和文）自己反映的ソフトウェアのための実行時検証とそのための仕様記述方式

研究課題名（英文）Specification Methods for Runtime Verification of Reflective Software

研究代表者
渡部　卓雄（Watanabe, Takuo）

東京工業大学・情報理工学（系）研究科・准教授

研究者番号：２０２２２４０８

交付決定額（研究期間全体）：（直接経費）　　3,600,000円

研究成果の概要（和文）：アクターモデルを基盤とした広域自己反映計算モデルの形式化方式およびその検証手法の提案を行った．広域自己反映計算とは，並行計算系の総体的な構造・振舞を対象とする自己反映計算方式である．本研究では，個々のアクターのメタレベルを並列合成することでこの方式を実現し，並行文脈指向プログラミング等への応用を通してその有効性を示した．また，アクターモデルを含む言語や計算モデルについて，その性質記述および証明のモジュール化を可能にする条件を明らかにし，それにもとづく検証方式および証明の再利用方式を提案した．

研究成果の概要（英文）：The main contributions of this work include (1) a novel construction method of group-wide reflective architectures and (2) extensible and modularized verification methods for concurrent language execution models. The key idea of the former is to apply parallel composition to actors that constitute the meta-level of the actor groups. This results in a uniform construction method for various types of meta-level actors. The proposed method can be used to implement concurrent context-oriented systems. For the latter, we proposed a method to support the extension of verified programs by interactively modifying their definitions and proofs and a modularized description/verification method for reflective concurrent systems.

研究分野：計算機ソフトウェア

キーワード：自己反映計算　広域自己反映計算　アクターモデル　文脈指向プログラミング　形式手法　実行時検証

1.　研究開始当初の背景

一般にプログラムの構造や実行過程に関する計算を，そのプログラムに関するメタレベル計算と呼ぶ．そしてプログラム自身の記述や構造および実行過程に関する計算，つまり自分自身に関するメタレベル計算のことを**自己反映計算 (リフレクション)** という．

一般に自己反映計算の能力（フレキシビリティ）とプログラムの安全性はトレードオフの関係にあり，多くの実用システムでは自己反映計算の機能を大幅に制限することで安全性を担保している．例えば Java には実行時にクラスを定義できるような自己反映計算機構が備わっているが，使い方が限定される上にセキュリティポリシーによっては完全に無効にされてしまう．また，型システムを用いた安全性の保証は言語拡張や多段プログラミング（実行時コード生成の一種）等では成果を挙げているが，動的な自己反映計算については現時点ではごく限定された結果しか得られていない．

一方，プログラムに要求される様々な性質を実行時に検査する技術は**実行時検証**（実行時検査，実行監視ともいう）と呼ばれており，専門の国際会議 (Runtime Verification) も開催されて活発に研究が行われている．静的検証と比べて検証できる性質が網羅的ではなく，また実行時オーバーヘッドという避けられない問題もあるが，動的適応システム等，一般に静的検証が難しい対象には適した手法といえる．また自己反映計算機構を用いて実行時検証を実現する手法もいくつか提案されているが，自己反映計算操作（自身に対するメタレベル計算操作）そのものが検証対象となるケースはほとんどない．

2.　研究の目的

本研究の目的は，実行時検証を用いて動的な自己反映計算（実行時の情報に依存する自己反映計算）を安全に行うための手法の提案である．対象となる系はアクターモデルにもとづく並行計算系とする．メタレベル計算動作を状態遷移系として抽象化する手法により，自己反映計算を含むシステム総体の実行時検証を可能にすることを特色とする．そしてこの手法はメタレベルに関する性質記述のモジュール化と，モデル検査法の併用による検査精度と実行時検証効率の向上を可能にする．本研究の成果は，文脈指向計算系を含む動的適応系や耐故障システム等，実行時情報に依存して動作するシステムの信頼性や安全性の向上に寄与することが期待できる．

3.　研究の方法

実行時検証に適した自己反映系の構成方式　主に基盤となる技術である実行時検証に適したメタレベル計算の表現手法を含む自己反映的プログラムの構成方式の研究を行う．

- 基本的な自己反映計算の枠組みの確立：アクターモデルを基盤として自己反映計算の枠組みについて検討し，その性質を明らかにする．本研究では，研究代表者が以前提案した広域自己反映計算 (Group-Wide Reflection, GWA) の実現方式を，アクターの並列合成という手法を用いて再構成する．これにより，従来から自己反映計算において難しいとされていたメタレベル計算の記述を容易にできる．

- プロトタイプの実装と評価：提案した GWA の計算モデルを，プログラミング言語 Erlang によって実装し，例題を通してその評価を行う．提案手法を並行文脈指向計算系の実現に応用し，その有効性を示す．

安全性に関する性質記述と検証手法　上記と並行して，自己反映計算システムの安全性に関する性質の記述形式およびその実行時検証手法についての研究を行う．

- 並行自己反映計算系の形式化とそのモジュール化：本研究で扱う並行自己反映計算系では，ベースレベルおよびメタレベルがアクターとして表現され，メッセージを用いて相互作用を行う．こ

のような系は状態遷移系として形式化でき，その性質は時相論理式などで自然に与えることができるが，ベースレベルとメタレベルの振る舞い記述が混在することでモデルの記述量が増大し，かつモデル記述のモジュール性が低下することが問題になる．研究代表者は Java における表明記述にアスペクト指向プログラミング (AOP) の手法を適用することで記述量を大幅に減少させることに成功しており，ここでもその手法を適用する．

- 検証手法とその評価：上記の形式化手法に加え，自己反映的な性質を含む並行計算系を対象とした検証方式の提案を行う．具体的には，拡張性，故障や誤りなどの非機能的要件に属する動作を含む系について，(実行時）検証を可能にする条件を明らかにする．それにもとづく検証方式の有用性を具体例を通して示す．

4. 研究成果

広域自己反映計算の形式化とその応用　アクターモデルを基盤とした広域自己反映計算モデルの形式化方式およびその検証手法の提案を行った（学会発表 8,9）．広域自己反映計算とは，複数の計算主体（ここではアクター）からなる系の総体的な振る舞いに対するメタレベルを介した自己反映計算方式である．この手法は研究代表者が過去に提案していたものであるが，メタレベルの構成が実装方式に依存したアドホックなものであった．本研究では，個々のアクターのメタレベルを並列合成することによりアクターのグループのメタレベルを構成する手法を提案した．

本手法の応用として，並行計算系における文脈指向プログラミング (COP) の一方式である並行文脈指向プログラミング (CCOP) を提案し，その実現方式の提案と評価を行った (学会発表 3,5)．具体的には実行時文脈の変化をアクターからなるグループ内において非同期的に伝搬する手法を広域自己反映計算手法を用いて実現している．加えてプログラミング言語 Erlang によるプロトタイプを元に提案手法の評価を行い，その有効性を確認した．

性質記述・証明のモジュール化と実行時検証　定理証明支援系 Coq で記述された証明済みの言語定義について，既存の証明を再利用しつつ拡張できるための条件を明らかにし，具体的な拡張手法を提案した（発表論文 1，学会発表 11）．加えて，複数の拡張を合成する手法を提案した（学会発表 7）．これはアクターモデルを含む言語や計算モデルについて，その性質記述および証明のモジュール化を可能にする方式である．これに関連して，アクターモデルのひとつの定式化である Aπ 計算の Coq による形式化を行った（学会発表 4）．

また，非同期メッセージ通信を伴う並行計算系を対象として，故障動作などを含む仕様記述のモジュール化方式を提案した（学会発表 1,2）．これにより，限定的ではあるが非機能的要件を自己反映計算として実現したアクター系の検証が可能になる．提案手法にもとづき，形式仕様記述言語 Sandal の設計および検証系の実装を行い，その有効性を明らかにした．

以上の考え方の応用として，人為的な誤りを含む（人間が実行する）タスクの形式化を行い，それにもとづくタスクの実行時検証の有用性を明らかにした（学会発表 6,10）．これは，人間が実行するタスクを並行計算系として定式化し，上で提案した方式と同様に人的誤り動作が混在するタスクを（非機能的な動作を含む並行計算系と同様に）モジュール化して記述することを可能にするものであり，これによってタスクの誤り耐性などの評価が容易になる．

5. 主な発表論文等

[雑誌論文] (計 1 件)

(1) 森口草介, 渡部卓雄, 定理証明支援系 Coq への対話的修正機構の導入, 情報処理学会論文誌 プログラミング (PRO), Vol. 5, No. 4, pp. 27–38, Sep., 2012. (査読あり)

[学会発表] (計 11 件)

(1) Masaya Suzuki & Takuo Watanabe, Sandal: A Modeling Language Supporting Exhaustive Fault-Injection, Workshop on Computation: Theory and Practice (WCTP 2014), Oct. 6–7, 2014, De La Salle University (Manila, Philippines), Theory and Practice of Computation (WCTP 2014), World Scientific, 2015 (印刷中). (査読あり)

(2) Masaya Suzuki & Takuo Watanabe, A Language Support for Exhaustive Fault-Injection in Message-Passing System Models, 1st Workshop on Logics and Model-Checking for Self-* Systems (MOD* 2014), Sep. 12, 2014, Centro Residenziale Universitario di Bertinoro (Bertinoro, Italy), EPTCS, Vol. 68, pp. 45-58, 2014, DOI:10.4204/EPTCS.168.4. (査読あり)

(3) 竹野創平・渡部卓雄, アクターモデルに基づく並行文脈指向プログラミング機構の実装と評価, 日本ソフトウェア科学会第 31 回大会, Sep. 7–10, 2014, 名古屋大学（愛知県名古屋市）. (査読なし)

(4) 安武祥平・渡部卓雄, Aπ 計算の Coq による形式化, 日本ソフトウェア科学会第 31 回大会, Sep. 7–10, 2014, 名古屋大学（愛知県名古屋市）. (査読なし)

(5) Takuo Watanabe & Souhei Takeno, A Reflective Approach to Actor-Based Concurrent Context-Oriented Systems, 6th International Workshop on Context-Oriented Programming (COP 2014), pp. 3:1–3:6, Jul. 29, 2014, Uppsala University (Uppsala, Sweden), DOI:10.1145/2637066.2637069. (査読あり)

(6) Naoyuki Nagatou & Takuo Watanabe, A Model-Checking Based Approach to Robustness Analysis of Procedures under Human-Made Faults, 2nd Asia Pacific Conference on Business Process Management (APBPM 2014), Jul. 3–4, 2014, Queensland University of Technology (Brisbane, Australia), Lecture Notes in Business Information Processing, Vol. 181, pp. 117–131, Springer-Verlag, 2014, DOI:10.1007/978-3-319-08222-6_9. (査読あり)

(7) 森口草介, 渡部卓雄, 検証付きプログラムに対する対話的修正の合成, 日本ソフトウェア科学会第 20 回ソフトウェア工学の基礎ワークショップ (FOSE2013), Nov. 28–30, 2013, 山代温泉（石川県加賀市）, ソフトウェア工学の基礎 XX, レクチャーノート/ソフトウェア学, Vol. 39, 近代科学社, pp. 131-136, 2013. (査読あり)

(8) Takuo Watanabe, Towards a Compositional Reflective Architecture for Actor-Based Systems, 3rd International Workshop on Programming based on Actors, Agents, and Decentralized Control (AGERE!@SPLASH 2013), pp. 19–24, Oct. 27–28, 2013, Hyatt Regency Indianapolis (Indianapolis USA), DOI:10.1145/2541329.2541341,
(査読あり)

(9) Takuo Watanabe, Compositional Construction of Group-Wide Meta-Level Architectures, Workshop on Computation: Theory and Practice (WCTP 2013), Sep. 30 – Oct. 1, 2013, University of the Philippines Diliman (Quezon City, Philippines), Theory and Practice of Computation, World Scientific, pp. 95–107, 2014, DOI:10.1142/9789814612883_0007. (査読あり)

(10) Naoyuki Nagatou & Takuo Watanabe, Robustness Analysis on Human-made Faults in Procedural Manuals, Workshop on Computation: Theory and Practice (WCTP 2013), Sep. 30 – Oct. 1, 2013, University of the Philippines Diliman (Quezon City, Philip-

pines), Theory and Practice of Computation, World Scientific, pp. 79–94, 2014, DOI:10.1142/9789814612883_0006. (査読あり)

(11) Sosuke Moriguchi & Takuo Watanabe, An Interactive Extension Mechanism for Reusing Verified Programs, 28th ACM Symposium On Applied Computing (SAC 2013), pp. 1236–1243, Mar. 18–22, 2013, Instituto Superior de Engenharia de Coimbra (Coimbra, Portugal), DOI:10.1145/2480362.2480594. (査読あり)

[図書] (計 0 件)

[産業財産権] (計 0 件)

[その他]

6. 研究組織

(1) 研究代表者 渡部 卓雄
(東京工業大学・大学院情報理工学研究科・准教授, 研究者番号: 20222408)

(2) 研究分担者 なし

(3) 連携研究者 なし