

平成 30 年 6 月 19 日現在

機関番号：12608

研究種目：基盤研究(C) (一般)

研究期間：2015～2017

課題番号：15K00089

研究課題名(和文) 広域自己反映計算にもとづく適応的並列計算系の実行時検証とそのため仕様の記述方式

研究課題名(英文) Specification and Runtime Verification Methods for Adaptive Parallel Systems based on Group-Wide Reflection

研究代表者

渡部 卓雄 (Watanabe, Takuo)

東京工業大学・情報理工学院・教授

研究者番号：20222408

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)：本研究課題では、アクターモデルを基盤とした自己反映的システムのための仕様記述および実行時検証のための仕組みを提供することを目的とする。そのためのアプローチとして、広域自己反映計算(GWR)のための新しい手法を提案した。それにもとづき、プログラミング言語Erlangによる広域自己反映計算機構を実装し、並行文脈指向プログラミングシステムへの応用を通してその有効性を確認した。また、アクターモデルのための形式検証フレームワークおよびAkkaのための実行時検証ライブラリを開発した。特に後者については分散型Webアプリケーションおよびデスクトップアプリケーションを対象にその有用性を示すことができた。

研究成果の概要(英文)：The goal of this project is to provide a framework for specification and run-time verification of actor-based reflective systems. To achieve the goal, we took a new approach to the actor-based group-wide reflection (GWR), which allows each actor to have capabilities of reasoning about/acting upon the collective behavior of the group it belongs to. We applied the proposed method to implement concurrent context-oriented programming (COP) systems. To realize COP in concurrent systems based on asynchronous communication, we must take care to control synchronizations among context changes and other computations. We adopt an instance of GWR to solve the synchronization problem regarding messages that cross two contexts. We also developed a static verification framework for actor-based systems in Coq, as well as a run-time verification library for Akka-based applications.

研究分野：プログラミング言語

キーワード：広域自己反映計算 自己反映計算 アクターモデル 関数リアクティブプログラミング

1. 研究開始当初の背景

携帯端末等の組み込みシステムやセンサネットワーク、web サービス等のオープンなシステムは、位置情報やハードウェア状態の変化、故障や機能変更といった実行時文脈の変化に柔軟に適応できることが望ましい。このような適応的システムの構成原理に自己反映計算(リフレクション)がある。一般にプログラムの構造や実行過程に関する計算を、そのプログラムに対するメタレベル計算と呼ぶ。そして、プログラムが自身に対するメタレベル計算を行う場合が自己反映計算である。自己反映計算は拡張性や適応性を持つソフトウェア、例えば拡張可能プログラミング言語や適応的ミドルウェア等の実現に広く応用されている。

並列・分散計算システムにおいては、計算資源の配備や耐故障動作等の非機能的要件(本来実現したい機能を支援する動作についての要件)に関する記述をモジュール化する上で自己反映計算が効果的である。特に、システムを構成する計算主体(プロセス、オブジェクト、アクター等)がそれぞれ個別に自己反映的動作をする方式に比べ、複数の計算主体を総体として扱う広域自己反映計算(Group-Wide Reflection)と呼ばれる手法がより有効であることがわかってきている。

一般に自己反映計算の能力(自由度)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由度を大幅に制限することで安全性を担保している。また、型システムによる静的な安全性の保証は言語拡張やマルチステージプログラミング(実行時コード生成の一種)等で一定の成果を挙げているが、実行時情報を用いる適応的システムに対する効果は限定的である。

本研究で扱うアクターモデルは非同期メッセージ通信に基づく並行計算モデルである。同期に伴う競合やデッドロックが生じないこと、共有・分散メモリシステムを問わず様々な並列計算系に対応できること、加えて実用的な処理系(Scala, Erlang 等)が普及してきたことに伴い注目されている。純粋なアクターモデルは非同期メッセージ通信のみを提供す

る。これで記述しにくい同期機構については、様々な形の協調(coordination)モデルと呼ばれる拡張が提案されているが、汎用性に欠ける。

一方、動作中のプログラムの振舞を観測してその性質を検査する技術は実行時検証と呼ばれ、専門の国際会議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いことと実行時オーバーヘッドの問題はあるが、適応的システム等、一般に静的検証が難しい対象に適した手法である。自己反映的な並列計算システムを実行時検証の対象とする場合、システムが本来行う計算(ベースレベル計算)とメタレベル計算が並行に観測される。この場合、検証したい性質をいかに記述するかが観測に伴うオーバーヘッドの軽減とともに問題になる。

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。

(a)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。

(b) 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。

(c) 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。ここでは AOP 等の手法をモデル記述に適用することをで記述量の減少を図る。

(d) 検証手法とその評価：上記の形式化手法に加え、自己反動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

アクターモデルにおける広域自己反映計算系とその実行時検証 研究代表者が本研究課題開始時までに行ってきたアクターモデル上の自己反映計算モデル

の研究にもとづき、プログラミング言語 Erlang による広域自己反映計算機構の実装とその評価を行った。具体的には、変化する実行環境への動的適応機構を対象に、そのオーバーヘッドを評価項目とした。本研究の特徴は、複数の計算主体 (アクター等) からなる系に関する自己反映計算を扱うことにあるが、特にアクターモデルのような非同期通信にもとづく系の場合、通信の遅れを考慮する必要がある。本研究では複数のアクターからなる系の動的適応を楽観的なアルゴリズムを用いて実装し、適応に伴う同期に必要なメッセージ数や再実行のオーバーヘッドが現実的な範囲内にあることを明らかにした (雑誌論文 2, 学会発表 16)。

アクターモデルの形式検証のための枠組みを確立させるため、アクターモデルにもとづく並行計算系を証明支援系 Coq を用いて検証するためのライブラリ Actario を実装し、例題を通してその有効性を明らかにした。アクターモデルではアクターの名前 (アドレス) の一意性を仮定しているが、Coq のような証明支援系による検証を行う上で、その形式化が問題となることがある。Actario ではアクターの名前の生成方式を定め、その方式で生成される名前の一意性を Coq で証明している。これにより、複雑な系を表現する場合においても名前付けが問題とならないことを示した (学会発表 18)。

上記と並行して、現実的な分散システムについての検証事例として、必ずしも信頼できない実行系 (例えば人間等) に基づくシステムを並行計算系としてモデル化し、モデル検査の手法を用いて欠陥に対する耐性解析を行う手法を提案した (雑誌論文 3)。

以上にもとづく知見をもとに、プログラミング言語 Scala によるアプリケーション (分散型 Web アプリケーションおよびデスクトップアプリケーション) を対象とした実行時検証フレームワークを実装し、その有効性を評価した (学会発表 8)。

また、アクターモデルを含む、オブジェクト指向の考えかたにもとづいて構成された系の動作の可視化を支援する手法 (学会発表 17, 19)、およびアクターモデルに特化した視覚的なデバッグ支援の手法

(学会発表 2, 5) をそれぞれ提案し, 例題を通してこれらの有効性を明らかにした. これらの成果により, 形式検証と可視化の両面からアクターモデルにもとづくシステムの開発を支援することを可能にした.

小規模組み込みシステム向けの関数リアクティブプログラミング言語 本研究課題では, 広域自己反映計算にもとづく文脈指向的なシステムの動作を実行時検証の対象としており, その典型的な応用として無線センサーアクターネットワーク (WSAN) を挙げている. 個々のセンサーノードやアクターノードの動作も実行時検証の対象となるため, それらの動作を形式化する必要がある. 典型的なセンサーノードはマイクロコントローラ等で構成される小規模なシステムであり, その上のプログラムはポーリングと状態機械の組み合わせやコールバック (割込みハンドラ等) で構成される. そのためプログラムの記述が分断されてモジュール化への障害となるとともに, 形式的な扱いが難しくなる. 我々は小規模組み込みシステム向けの関数リアクティブプログラミング (FRP) 言語を設計・実装し, 実際にマイクロコントローラ上で評価を行ってその有用性を明らかにした (学会発表 12,14,20). 設計した言語は純粋な関数型言語であり, 実行時検証のための動作の形式化が可能になる.

本言語の実行モデルにアクターモデルを採用することで, 非同期性, 割り込みや例外, 分散処理等の FRP による表現を可能にすることを示すとともに (学会発表 1, 6, 10, 11), 自己反映計算機構の導入により, 実行時文脈に依存した処理の適切なモジュール化が可能になることを示した (学会発表 3, 7, 9). また, 本言語によって小型ロボット等の制御を宣言的に記述できるが, その際に必要な制御パラメータの決定方式として, 制御による目標を表す時相論理式の反証にもとづく手法を提案した (学会発表 15).

アクターモデルや関数リアクティブプログラミングの考え方は, 単純な仕組みで様々な系の振る舞いを記述できる言語機構を提供する. しかしそのためにパターンやクリシェと呼ばれる「典型的な振る舞い」を表すコード片がプログラム中に頻出する. こ

れらの記述を支援する上でマクロの考え方は有用である. 我々は言語処理系に衛生的マクロ定義機構を導入する新しい手法を提案し, 例を通してその有効性を示した (雑誌論文 1, 学会発表 4, 13).

5. 主な発表論文等

[雑誌論文] (計 3 件)

- [1] 高桑健太郎, 渡部卓雄, 多言語に対応した衛生的マクロ機構導入方式, コンピュータソフトウェア, Vol. 35, No. 2, pp. 33–39, 日本ソフトウェア科学会, Apr., 2018. https://doi.org/10.11309/jssst.35.2_33
- [2] 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, コンピュータソフトウェア, Vol. 33, No. 1, pp. 167–180, 日本ソフトウェア科学会, Feb., 2016. https://doi.org/10.11309/jssst.33.1_167
- [3] Naoyuki Nagatou and Takuo Watanabe, A Model-Checking Based Approach to Robustness Analysis of Procedures under Human-Made Faults, *International Journal of Industrial Engineering: Theory, Applications and Practice*, Vol. 22, No. 4, pp. 494–508, Aug., 2015. <http://journals.sfu.ca/ijietap/index.php/ijie/article/view/1794>

[学会発表] (計 21 件)

- [1] 渡部卓雄, アクターモデルと関数リアクティブプログラミングの融合による小規模組み込みシステム開発, 組み込み技術とネットワークに関するワークショップ (ETNET2018), 情報処理学会研究報告 Vol. 2018-EMB-47, No. 26, pp. 1–6, Mar., 2018 (査読なし). <http://id.nii.ac.jp/1001/00186360/>

- [2] Kazuhiro Shibana, Takuo Watanabe, Actor-verse: A Reversible Debugger for Actors, 7th ACM SIGPLAN International Workshop on Programming Based on Actors, Agents, and Decentralized Control (AGERE 2017), pp. 50–57, ACM, Oct., 2017 (査読あり). <https://doi.org/10.1145/3141834.3141840>
- [3] Takuo Watanabe, Reactive Reflection in an FRP Language for Small-Scale Embedded Systems, Workshop on Meta-Programming Techniques and Reflection (META 2017), Oct., 2017 (査読あり). <https://2017.splashcon.org/track/meta-2017>
- [4] 高桑健太郎, 渡部卓雄, 多言語に対応した衛生的マクロ機構導入方式, 日本ソフトウェア科学会第 34 回大会, Sep., 2017 (査読なし). <http://www.psg.cs.titech.ac.jp/posts/2017-09-19-JSSST2017.html>
- [5] 柴内一宏, 渡部卓雄, アクターシステムを対象としたリバースデバッキングのためのフレームワーク, 日本ソフトウェア科学会第 34 回大会, Sep., 2017 (査読なし). <http://www.psg.cs.titech.ac.jp/posts/2017-09-19-JSSST2017.html>
- [6] Takuo Watanabe & Yuji Iwasaki, An Actor-Based Execution Model of an FRP Language for Small-Scale Embedded Systems 7th Workshop on Computation: Theory and Practice (WCTP 2017), pp. 123-135, Sep., 2017 (査読あり).
- [7] 渡部卓雄, 小規模組み込みシステム向け FRP 言語のための自己反映機構, 情報処理学会第 45 回組込みシステム研究会 (SIGEMB45), 情報処理学会研究報告, Vol. 2017-EMB-45, No. 2, pp. 1–6, Jun., 2017 (査読なし). <http://id.nii.ac.jp/1001/00182176/>
- [8] Paul Lavery & Takuo Watanabe, An Actor-Based Runtime Monitoring System for Web and Desktop Applications, 18th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD 2017), pp. 385–390, IEEE, Jun., 2017 (査読あり). <https://doi.org/10.1109/SNPD.2017.8022750>
- [9] Takuo Watanabe & Kensuke Sawada, Towards Reflection in an FRP Language for Small-Scale Embedded Systems, 2nd Workshop on Live Adaptation of Software SYstems (LASSY 2017), ACM, Apr., 2017 (査読あり). <http://dx.doi.org/10.1145/3079368.3079387>
- [10] Takuo Watanabe & Kensuke Sawada, Towards an Actor-Based Execution Model of an FRP Language for Small-Scale Embedded Systems, 情報処理学会第 43 回組込みシステム研究会 (SIGEMB43), IPSJ SIG Technical Report, Vol. 2016-EMB-43, No. 6, pp. 1–5, Nov., 2016 (査読なし). <http://id.nii.ac.jp/1001/00175597/>
- [11] Takuo Watanabe & Kensuke Sawada, Towards an Integration of the Actor Model in an FRP Language for Small-Scale Embedded Systems, 6th International Workshop on Programming based on Actors, Agents, and Decentralized Control (AGERE 2016), Oct., 2016 (査読あり). <https://2016.splashcon.org/track/agere2016/>
- [12] Kohei Suzuki, Kanato Nagayama, Kensuke Sawada & Takuo Watanabe, CFRP: A Functional Reactive Programming Language for Small-Scale Embedded Systems, *Theory and Practice of Computation: Proceedings of Workshop on Computation: Theory and Practice WCTP2016*, pp. 1–13, World Scientific,

- Dec., 2017 (査読あり). https://doi.org/10.1142/9789813234079_0001
- [13] 星野友宏, 高桑健太郎, 渡部卓雄, OMeta のための衛生的マクロ定義機構導入方式, 日本ソフトウェア科学会第 33 回大会, Sep., 2016 (査読なし).
- [14] Kensuke Sawada & Takuo Watanabe, Emfrp: A Functional Reactive Programming Language for Small-Scale Embedded Systems, Modularity 2016, Constrained and Reactive Objects Workshop (CROW 2016), Companion Proceedings of the 15th International Conference on Modularity, pp. 46–54, ACM, Mar., 2016 (査読あり). <http://dx.doi.org/10.1145/2892664.2892670>
- [15] 峰尾太陽, 石井大輔, 渡部卓雄, 時相論理式の反証を用いた制御器のチューニング, 電子情報通信学会システム数理と応用研究会, 信学技報, Vol. 115, No. 480, MSS2015-79, pp. 61–66, Mar., 2016 (査読なし). <https://www.ieice.org/ken/paper/20160304gb7D/>
- [16] Sohei Takeno & Takuo Watanabe, A Reflective Implementation of an Actor-based Concurrent Context-Oriented System, 14th Workshop on Adaptive and Reflective Middleware (ARM 2015), pp. 7:1–7:6, ACM, Dec., 2015 (査読あり). <http://dx.doi.org/10.1145/2834965.2834972>
- [17] 大村裕, 渡部卓雄, CODE: Code Oriented Diagram Editor (ライブ論文), ソフトウェア工学の基礎 XXII (日本ソフトウェア科学会 FOSE2015), レクチャーノート/ソフトウェア学, Vol. 41, 近代科学社, pp. 235–236, Nov., 2015 (査読あり).
- [18] Shohei Yasutake & Takuo Watanabe, Actario: A Framework for Reasoning About Actor Systems, 5th International Workshop on Programming based on Actors, Agents, and Decentralized Control (AGERE 2015), pp. 4:1–4:10, Oct., 2015 (査読あり). <https://2015.splashcon.org/track/agere2015>
- [19] Yuu Ohmura & Takuo Watanabe, Design and Implementation of an Advanced Source Code Reading Tool for Framework-based Software, *Theory and Practice of Computation: Proceedings of Workshop on Computation: Theory and Practice WCTP2015*, pp. 106–111, World Scientific, Apr. 2017 (査読あり). http://dx.doi.org/10.1142/9789813202818_0007
- [20] 澤田賢祐, 鈴木康平, 渡部卓雄, 小規模組み込みシステムにおける FRP の応用に向けて, 電子情報通信学会ソフトウェアサイエンス研究会 (SIGSS), 信学技報, Vol. 115, No. 20, SS2015-1, pp. 1–5, May, 2015 (査読なし). <https://www.ieice.org/ken/paper/20150511WbAZ/>
- [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 (WATANABE, Takuo)
東京工業大学・情報理工学院・教授
研究者番号: 20222408
- (2) 研究分担者 なし
- (3) 連携研究者 なし