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

令和　　元 年　　６ 月 ２１ 日現在

機関番号：１４３０１
研究種目：基盤研究(C)（一般）
研究期間：2016～2018
課題番号：１６Ｋ００００９６

研究課題名（和文）柔軟かつ利便性の高いアクセス制御機能を備えたプログラミング言語

研究課題名（英文）A Programming Language That Supports Flexible and Convenient Access Control

研究代表者
馬谷　誠二（Umatani, Seiji）

京都大学・情報学研究科・助教

研究者番号：４０３７８８３１

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，分散アプリケーションにおけるコンポーネント間の相互作用の安全性の向上を図るため，柔軟かつ利便性の高いアクセス制御機能を備えたプログラミング言語の開発を行った．特に，アクセス制御論理(ACL)に基づくアクセス制御機能を，静的オブジェクト指向言語Java，および動的言語Luaの拡張機能として設計し，そのプロトタイプを実装した．さらに，動的言語で書かれたプログラム中の機密データへのアクセスを効率良く制御するために必要となる静的解析手法についても開発を行った．

研究成果の学術的意義や社会的意義
(1) アクセス制御論理(ACL)を用いることの利点や適切な役割の追求，(2) ACLで表現されたセキュリティ機能を備えた拡張オブジェクト指向言語，(3) (2)で表現されたプログラムのための解析手法，が本研究の主な学術的，社会的意義である．
堅牢なセキュリティの確保が容易ではないと考えられていた領域において，柔軟かつ洗練された計算モ デルに基づいたアクセス制御を導入．キュリティの専門家ではな い一般のプログラマでも，自身のプログラム中の任意のコンポーネントに対しこれまでより簡単にアクセス制御機能を導入し，外部の危険性に対する頑健性を持たせることが可能となった．

研究成果の概要（英文）：In this research, in order to improve the security of interaction between components in distributed applications, we developed a programming language with flexible and convenient access control mechanisms. In particular, we designed an access control mechanism based on access control logic (ACL) as an extension of Java, an statically-typed object-oriented language, and Lua, a dynamically-typed language, and then implemented their prototypes. Furthermore, we have also developed a static analysis method required to control efficiently access to confidential data in a program written in a dynamically-typed language.

研究分野：計算機ソフトウェア

キーワード：プログラミング言語　セキュリティ　アクセス制御　動的言語　静的解析

様　式　Ｃ－１９、Ｆ－１９－１、Ｚ－１９、ＣＫ－１９（共通）

１．　研究開始当初の背景

近年，web アプリケーションや携帯端末アプリケーション，その中でも特に，他人によって書かれたコンポーネントを再利用したり，他のアプリとの協調動作が必要なプログラムのコンポーネント間の連携に関する欠陥に起因する脆弱性が問題となっている．実際，最新の JavaScript 仕様にはオブジェクトへの参照を無効化する機能が採用されたり，コンポーネントの実行をサンドボックス化する技術など，アクセス制御に関する研究が盛んである．また，現状の Android アプリ(Dalvik バイトコード)のアクセス制御については，様々な問題点が指摘されており，重要な課題となっている．

これらのアプリケーションを対象としたアクセス制御に関する既存技術の多くは，プログラミング言語に組込みの機能ではないため，機能面で単純過ぎたり，アドホックな手法なため扱いにくかったりすることが殆どである．そのような機能の上に安全なシステムを構築するには，セキュリティに関する豊富な専門知識が必要となり，一般のプログラマにとっては敷居が高い．さらに，言語に組込みの機能ではないため，プログラム実行性能へかかる負担も大きい．

一方，分散計算の分野においては，コンポーネント間の協調動作やファイアウォール技術の安全性に関する研究が盛んに行われており，中でも，アンビエント計算は，主体(プロセス)，場所，資源，それらの間の協調動作を統一的に扱う，シンプルかつ柔軟な表現力を備えた計算体系である．研究代表者はこれまでに，アンビエント計算の機能を備えた高水準プログラミング言語の開発を行ってきた．プログラマは，高水準なアンビエント計算の機能を用いてプログラムを書けるため，複雑な協調動作を，簡潔かつ直観的に書くことができる．

それらの先行研究で明らかとなった分散コンポーネントの制御に関する知見の多くは，より一般的な web アプリケーション，Android アプリなどへも応用できるはずであるというのが，本研究課題の着想に至った経緯である．

他にも，アクセス制御に関する理論的研究としては，特定の実行モデルに依存せずアクセス制御を扱うための論理体系としてアクセス制御論理(access control logic, 以下 ACL と呼ぶ)が知られている．一般のアプリケーションは様々な言語を用いて記述されており，ACL はそれらの間の相互作用を表現する一種の共通インタフェースとして捉えることも可能である．

２．研究の目的

前述のとおり，アンビエント計算はコンポーネント間の連携の扱いに優れた計算体系であるが，アクセス制御を行う必要のあるすべてのコンポーネントやアプリケーションを専用の言語を使って記述するというのは，現実的ではない．そこで本研究では，システム全体はあくまでアンビエント計算モデルによって制御しつつ，他の言語（本研究の主な対象は，JavaScript や Java などのオブジェクト指向言語とする）で書かれた各コンポーネントとの間のアクセス制御については，ACL を共通インタフェースとして相互の振舞いを理解し合うことにより，全体としての安全性を確立する言語基盤の開発を行うことにする．開発にあたっては，対象とするそれらの言語が本来備える表現力を損わず，また，既存のコードにあまり手を加えなくても済むようコードの再利用性にも留意する．

ACL は非常に強力な論理体系であり，それによって表現されたアクセス制御に関する諸性質が満たされていることを保証するための言語機構（静的解析や動的検査など）を実現するのは挑戦的な課題と言える．特に，通常のオブジェクト指向言語では，ACL における主体(principal)に相当する情報を直接的に扱う手段は提供されていないため，アンビエント計算や ACL の表現する実行モデルに対応するオブジェクト動作モデルを正確に定義することも重要な課題である．また，システム全体の安全性を確保するには，複数言語によって記述されたアプリケーションが全体としてどのように振舞うのかに関する正確な情報が求められる．そのためには，複数コンポーネント間をまたがる機密情報の流れを正確に把握するための解析手法を構築し，言語基盤の一部として組込む必要がある．

そこで本研究では，以下の 3 つを具体的な研究課題として挙げることとする．

(1) アンビエント計算モデルと ACL の融合．研究代表者の先行研究において開発したアンビエント計算言語に対し，ACL 検査機能の追加を行う．言語処理系が，ACL の記述に沿った各アンビエントの動作を実現するには，処理系内部のデータ表現等，様々なレベルでの改良が必要となる．

(2) 汎用オブジェクト指向言語における柔軟なアクセス制御機能を備えた参照の実現．ACL をサポートするための機能をオブジェクトへの参照に追加する．具体的には，「誰が」「どのような権限をもって」アクセスするのかや，「誰からの」「どのような操作から」保護するか，をすべて参照に付加された言語組込みの属性として直接記述できるようにする．また，既存コードにおける通常の参照の使用パターンを解析し，機能の追加された参照へと置き換えたり新たに追加したりするための手法を構築し，可能な限り既存コードの再利用性を確保する．

(3) 両言語(モデル)間の相互作用を考慮した解析手法の開発．システム全体の実行時のデータの流れを解析・検査するための手法を開発し，提案する言語基盤による安全性確保のための機構として用いる．伝統的な言語で書かれたプログラム中のデータの流れを解析する研究

は既に数多くなされているが，ここでは，アンビエント計算と(JavaScript のように動的な性質を含んだ)汎用オブジェクト指向モデルの間の相互作用まで考慮して精密に解析することを目標とする．

３．研究の方法

柔軟かつ扱いやすいアクセス制御機能を備えたプログラミング言語を実現するにあたって，研究目的で述べた以下の課題に取り組んだ．まず，%研究代表者の先行研究により実現済みのアンビエント計算言語を用い，(1) アンビエント計算モデルと ACL の融合，(2) 汎用オブジェクト指向言語における柔軟なアクセス制御機能を備えた参照の実現に取り組んだ．これらは計算モデルの間の連携も考慮する必要があるため相互に密接に関連しており，両方の設計を同時に進めた．その後，それぞれの実装を行うとともに，(3) 両言語間の相互作用を考慮した解析手法の開発を行った．

４．研究成果

(1) アクセス制御の実現に必要となる単純な仕組みをプログラミング言語の言語機能として組み込むことで，開発者自身が各システムやアプリケーションに適したアクセス制御のロジックをプログラム中に記述できるような言語の設計を行った．具体的には，Java の実行モデルに対しアクセス制御機能を追加した拡張オブジェクト指向言語を提案した．オブジェクト指向言語の基本的な実行の仕組みとアクセス制御のための仕組みを結びつけることで，「システムが資源へのアクセスを正しく制御する」ことを，外部サービスやブラックボックス化された専用の機構に依存せず，「基本的な言語機能が正しく実装されて いる」ことだけに依存し保証できることが，提案言語の特徴である．
(2) (1)では，静的な型システムを持つプログラミング言語においてアクセス制御に必要となる基本機能の設計を行ったが，本年度は，動的な言語に対して同様の機能を提供する手段の設計と実装を行った．具体的には，アクセス制御論理に基づいた IoT 向け分散型アクセス制御フレームワークを Lua 言語上で実現した．提案するアクセス制御論理によって，分散型アクセス制御で必要となる複雑な機能を厳密な論理体系上で推論・検証可能な形式で定式化することができ，またその表現力によって，ユーザは簡潔かつ柔軟なアクセス制御ポリシーの記述が可能となる．本フレームワークは，従来用いられている主要なアクセス制御機構であるアクセス制御リストやケーパビリティなどを含む広範囲な制御ポリシーをカバーできると考えている．
(3) 動的言語のアクセス制御を実現するために必要となる静的解析記述として，抽象解釈に基づく情報流解析技術の研究を行った．具体的にはコンパイル先の低レベルな中間表現の解析では正確に読み取れないプログラムの振舞いを解析するために，ソース言語レベルの実行を織り交ぜることにより解析の精度を保証する新たな仕組みを考案した．研究期間全体では，静的な型システムを持つプログラミング言語においてアクセス制御に必要となる基本機能の設計，動的な言語に対して同様の機能を提供する手段の設計と実装を行った．具体的には，アクセス制御論理に基づいた IoT 向け分散型アクセス制御フレームワークを Lua 言語上で実現した．提案するアクセス制御論理によって，分散型アクセス制御で必要となる複雑な機能を厳密な論理体系上で推論・検証可能な形式で定式化し，その表現力によって，ユーザは簡潔かつ柔軟なアクセス制御ポリシーの記述が可能である．本研究課題で開発した言語処理系およびフレームワークは，従来用いられている主要なアクセス制御機構であるアクセス制御リストやケーパビリティなどを含む広範囲な制御ポリシーをカバーできるものであると考えている．

５．主な発表論文等

〔雑誌論文〕（計 2 件）

① 中村 真也, 鵜川 始陽, 馬谷 誠二, 規則違反コードの構造を反映した木パタンを用いるコード検査器, 情報処理学会論文誌 プログラミング, 査読有, Vol.9, No.4, 2016, 1-15
② Hiroshi Yoritaka, Ken Matsui, Masahiro Yasugi, Tasuku Hiraishi, Seiji Umatani, Probabilistic guards: A mechanism for increasing the granularity of work-stealing programs, Parallel Computing, 査読有, Vol.82, 2018, 19-36
DOI: 10.1016/j.parco.2018.06.003

〔学会発表〕（計 13 件）

① Hiroshi Yoritaka, Ken Matsui, Masahiro Yasugi, Tasuku Hiraishi, and Seiji Umatani, Extending a Work-Stealing Framework with Probabilistic Guards, Ninth International Workshop on Parallel Programming Models and Systems Software for High-End Computing

(P2S2) 2016, 2016

② Daisuke Muraoka, Masahiro Yasugi, Tasuku Hiraishi, and Seiji Umatani, Evaluation of an MPI-Based Implementation of the Tascell Task-Parallel Language on Massively Parallel Systems, Ninth International Workshop on Parallel Programming Models and Systems Software for High-End Computing (P2S2) 2016, 2016

③ 馬谷 誠二, 藤原 康史, 五十嵐 淳, 階層的グループ化に基づき Android アプリの安全性を向上するバイト コード書換えツール, 日本ソフトウェア科学会第 33 回大会, 2017

④ 馬谷 誠二, アクセス制御機能の組み込まれた拡張オブジェクト指向言語, 第 58 回プログラミング・シンポジウム, 2017

⑤ 谷口 力斗, 馬谷 誠二, 鵜川 始陽, データフロー解析結果を付加した構文木に対するパターンマッチによるコード検査, 第 113 回情報処理学会プログラミング研究会, 2017

⑥ 寄高 啓司, 八杉 昌宏, 平石 拓, 馬谷 誠二, 優先度ならびに重みを用いたワークスティールフレームワークの性能改善, The 1st. cross-disciplinary Workshop on Computing Systems, Infrastructures, and Programming (xSIG 2017), 2017

⑦ 重本 孝太, 八杉 昌宏, 平石 拓, 馬谷 誠二, HOPE コンパイラのプロトタイプ実装, 2017 年並列/分散/協調処理に関する『秋田』サマー・ワークショップ (SWoPP2017), 2017

⑧ 良本 海, 八杉 昌宏, 平石 拓, 馬谷 誠二, 仮想環境を考慮した要求駆動型負荷分散, 日本ソフトウェア科学会第 34 回大会, 2017

⑨ 五十嵐 琢磨, 馬谷 誠二, アクセス制御論理に基づく IoT 向け分散型アクセス制御フレームワーク, 第 15 回 ディペンダブルシステムワークショップ (DSW 2017), 2017

⑩ 西牟禮 亮, 八杉 昌宏, 平石 拓, 馬谷 誠二, 並列分散フレームワークの耐障害性評価のための通信障害模擬機能, 第 20 回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(PPL2018) (ポスター発表), 2018

⑪ 馬谷 誠二, JVM 上の動的言語のための抽象解釈, 情報処理学会第 121 回プログラミング研究会, 2018

⑫ 佐多 育斗, 八杉 昌宏, 平石 拓, 馬谷 誠二, 分割統治型総和の部分的計算結果を効率よく利用する方式の研究, 情報処理学会第 121 回プログラミング研究会, 2018

⑬ 馬谷 誠二, JVM 上の動的言語のための抽象解釈の実装, 第 60 回プログラミング・シンポジウム, 2019

６．研究組織