

科学研究費助成事業（科学研究費補助金）研究成果報告書

平成24年5月18日現在

機関番号：14301

研究種目：若手研究（B）

研究期間：2009～2011

課題番号：21700029

研究課題名（和文） 高級な通信・移動機能を備えた実用的アンビエント計算言語

研究課題名（英文） A Practical Ambient-calculus Based Language with High-level Communication/Migration Supports

研究代表者

馬谷 誠二（UMATANI SEIJI）

京都大学・大学院情報学研究科・助教

研究者番号：40378831

研究成果の概要（和文）：本研究では、アンビエント計算に基づく実用的プログラミング言語の開発を行った。開発した言語は、 π 計算におけるチャンネル通信に相当する機能、静的スコープによる遠隔地の指定、動的束縛の構文による明確な分離といったプログラマにとって直観的に理解しやすい機能を備えている。さらに、高水準言語を実現するためのより低水準なアンビエント計算実行フレームワーク、広域分散環境上で正しく実行するための実装技術の開発も行った。

研究成果の概要（英文）：In this research, we developed a practical ambient-calculus based programming language. The developed language has several intuitive features easily understandable to programmers: channel communication of the pi calculus, identification of remote locations with lexically scoped names, and syntactically distinguished constructs for dynamic bindings. Furthermore, we also developed execution frameworks of lower-level ambient calculus and general implementation techniques for correctly executing its programs in wide-area distributed environments.

交付決定額

（金額単位：円）

	直接経費	間接経費	合計
2009年度	1,100,000	330,000	1,430,000
2010年度	1,000,000	300,000	1,300,000
2011年度	700,000	210,000	910,000
総計	2,800,000	840,000	3,640,000

研究分野：総合領域

科研費の分科・細目：情報学・ソフトウェア

キーワード：プログラム処理系，プロセス計算，分散システム，コード移動

1. 研究開始当初の背景

通信ネットワークや分散システムのさらなる拡大に伴い、分散ソフトウェアの複雑さは増加する一方である。このような状況において、プログラマがそれら複雑さを適切に理解し、正しいアプリケーションを記述するには、それにふさわしい抽象モデルを提供する新たなプログラミング言語が必要である。

そのような抽象モデルを提供する形式的

計算モデルのひとつにアンビエント計算がある。アンビエント計算では、分散システムはアンビエントの階層構造として表現され、コード(アンビエント)の移動は階層構造に対する変更操作として実現される。

アンビエント計算やその拡張計算モデル、プログラミング言語に関する既存研究は数多く存在するが、現実のシステムを記述するためのプログラミング言語としての観点からは、そのほとんどが以下のような問題を抱

えている:

- (1) すべてのアンビエントの境界が潜在的にファイアウォールになり得ると仮定し設計されているため、アンビエント間の通信・移動の機能が低レベルすぎる。そのため、あらゆる通信・移動の経路上に前もって仲介役を担うアンビエントを配置する必要がある。しかし、現実にはすべての通信がファイアウォールをまたがる事はなく、状況に応じて、より簡潔かつ柔軟に記述できる方が望ましい。
- (2) 移動先での資源アクセスを可能とするために、言語には動的な機能が不可欠であるが、既存研究のほとんどはプログラムの動的な機能とそれ以外の機能を明確に分離していない。しかし、プログラム中、実行時コンテキストに依存する動的な部分と、それ以外の静的な性質をもつ部分を区別できる方が可読性の点からは望ましい。さらに、この点は処理系の静的解析や型システムにも悪影響を与えている。例えば、アンビエントの移動可能性を判定するためのある型システムにおいては、実行時にしか分からない情報を静的な名前の属性に含めて型解析するため、本来移動するはずのないアンビエントであるのに誤って移動する可能性がある、と判定するケースが多い。

これら問題点の解決を試みる研究はいくつか存在する。例えば $lsd\pi$ は、名前の静的スコープを用いた遠隔通信を可能にする事により記述性を向上させている。しかし、 $lsd\pi$ は逆に移動計算に必要な不可欠な動的な機能を備えておらず、またネットワークはより単純な構造に限定されている。他に、 λ 計算に移動機能を追加し、名前の静的スコープにより遠隔通信を可能とする研究も存在する。

2. 研究の目的

(1) 概要

本研究は、アンビエント計算に基づく実用的な分散プログラミング言語の開発を目的とする。ここでいう実用的な言語とは、単なる形式的モデルの研究のためのツールではなく、プログラムの記述性・可読性・保守性に加え、処理系の実行性能まで考慮した、現実のシステム開発に利用可能なプログラミング言語を指す。

より具体的には、1. で述べた既存研究の抱える問題点を克服できるよう、 π 計算相当のチャンネルを用いたアンビエント間通信、静的スコープによる遠隔通信・移動、構文的に明確

に分離された動的資源アクセスといった、プログラムの直観的理解を助ける機能を備えた言語を設計する。また、移動コードや通信メッセージの安全性を保証するための型システムの構築も目標とする。さらに、プログラムの振舞いに関する情報を型に含める事で、効率良い処理系の実装にも型システムを積極的に用いる。

(2) 本研究の特色および独創的な点

本研究の特色は、従来のプロセス計算ベースの言語と比較し、より高レベルな抽象化に基づいて言語設計、処理系の実装技術の開発を行う点にある。特に、言語設計において動的な機能をそれ以外の静的な性質を持つ機能から明確に分離しようとする点は、既存研究にはないいくつかの有益な効果をもたらす。まず、通信や移動先の指定を柔軟に行える上、直観的で分かりやすいため、プログラマが本来意図する動作をそのままコードに反映でき、バグの混入を防ぐ事ができる。また、言語処理系の開発においても、プログラム解析・型システムといったコンパイラ技術をより効果的に適用する事が可能となる。

(3) 本研究の効果・意義

本研究は、モバイルエージェントシステム、グリッド計算、クラウドコンピューティングなどに代表される多種多様な分散システムを構築するための新たな言語基盤を提供する。既存の汎用言語を用いてシステムを構築する場合と異なり、形式的モデルに基づく安全なコードを使って分散資源へアクセスするため、たとえば、システム利用者は安心して移動してくるプログラムを実行できる。また、提供される高レベルかつ柔軟な言語機能は、開発プロセスにおける大幅なバグの削減を可能にする。さらに、将来的には、プロセス計算を用いた仕様記述・プログラム検証などの関連技術とのシームレスな連携により、大規模分散システム開発全体における生産性の大幅な向上が期待できる。

3. 研究の方法

本研究は、おおまかに以下の3つのフェーズに分けて行った。

- (1) 言語仕様の設計: プログラムの記述性を重視しつつ、型システム・処理系の実行性能についても十分考慮した上で、実現可能かつ厳密な言語シンタックスおよびセマンティクスを設計する。
- (2) 型システムの構築: 移動コード・通信メッセージの安全性だけでなくプログラムの振舞いも情報として含んだ適切な型表現を選択する。おもに実装効率の改

善に利用する。

- (3) 処理系の実装: 様々なプログラム解析手法や型システムから得られる情報を利用し、通信・移動の最適化を行う。

実際には処理系の実装については、言語仕様の設計が済んだ段階で型システムに依存しないを並行して進めた。

(1) 言語仕様の設計

「2. 研究目的」で説明した言語機能のうち、静的スコープによる遠隔通信・移動と明確に分離された動的資源アクセスに関し、基本的な部分の設計については、研究代表者が本課題に取り組む以前の研究成果を利用する。ただし、先行研究においては移動・通信の記述性に重点を置いていたため、ファイアウォールを表現するのに必要な機能については未検討のままであった。そこで、以下の機能を追加する:

- ① 先行研究において提案済みの言語機能との統一性を持たせるため、ファイアウォールの記述にも名前スコープの概念を用いる。具体的には、「閉じた」アンビエントの概念を導入し、ある名前宣言のスコープの内側であっても、(ちょうど境界線上にいるプロセスを除き)閉じたアンビエントの内側からはその名前を参照できない事にする。外側との全てのやり取りは、境界線上にいるプロセスを仲介して記述する。
- ② 上述の「閉じた」アンビエントの概念だけでは不十分な場合には、他の汎用言語で書かれたより低水準な機能を用いたコードを直接埋め込めるようにする。

(2) 型システムの構築および処理系の改良

通信メッセージやアンビエントの型安全性を保証する研究は広く行われており、本研究ではそれらを参考にしつつ我々の言語にふさわしい型表現・型規則を考案し、安全性を確保する。一方、処理系の効率化のための型については先に行う実装の中身に依存するが、以下のような検討を進める。

まず、アンビエントが移動する可能性があるかをアンビエント本体の実行コードの型検査によりチェックする。動かないアンビエントが特定できれば、それらの移動に関わるコスト(データの逐次化、位置情報の管理・更新等)を省略できる。本研究では、さらに、近傍アンビエント間の相対的な位置関係に関する情報を型に含ませるよう拡張する。互いの相対的な位置が変わらないアンビエント同士をグループ化できれば、グループ全体が移動する場合であっても、その内部における通信・移動を含む多くのインタラクションにかかるコストを削減できると考えられる。

(3) 処理系の実装

先行研究で開発していた言語仕様検討のためのプロトタイプ言語処理系をベースに実装を行う。まず、先に述べたファイアウォール機能をこの実装に追加する。また、既存の実装では、他の研究者らによって提案された既存実装技術のいくつかを用いているが、それらの研究では想定されていなかった遠隔通信・移動機能を取り入れているため通信に余分なコストがかかる。そこで、プログラム中の名前の参照関係を解析する事により通信コストの削減を図る。具体的な効果としては、通信メッセージおよび移動コードの通信距離の短縮、アンビエントの位置情報管理の単純化などが挙げられる。解析手法は、手続き間の変数使用解析・別名解析といった従来手法を発展させたものを考案する。

4. 研究成果

(1) 高水準言語設計

- ① ネットワーク構成の制約に縛られる事なく、柔軟に通信やコード移動を記述するための機能を設計した。具体的には、通信のための言語機能を動的束縛とレキシカル束縛の2種類に分け、レキシカル束縛を用いた通信については、実装レイヤが面倒を見ることで透過的な遠隔通信・移動を行えるようにしている。そのような言語機能を備えた拡張アンビエント計算のシンタックスおよびセマンティックスを考案し、各種アプリケーションの記述性が向上されることを例示した。
- ② 物理的なネットワーク構成にとどまらず、XMLなどを含む論理的階層構造の抽象化にもアンビエントを利用可能であるというアイデアに基づき、Webアプリケーション開発を行うための言語組み込み機能の設計を行った。「HTTPサーバ、ブラウザ、DOMツリーのすべてが階層化されたアンビエントで表現できる」という単純なモデルを採用する事で、シンプルな機能設計を実現している。

(2) 高水準言語機能の実装技術の開発

- ① 上記(1)において述べた言語機能を備えた、共有メモリ上で動作するプロトタイプ処理系の実装を行った。この処理系は、Webアプリケーションを動作させる場合のサーバ側処理系としても用いられる。
- ② JavaScriptで書かれたアンビエント言語処理系の実装を行った。アンビエント計算の基本機能を備える他、ブラウザ上でのUIに関する各種処理(HTMLの表示、

イベント処理)も実行できる。

(3) 低水準処理技術の開発

各種アンビエント計算のうち、通信および移動の制御のきめ細かさを特徴とする Safe アンビエントを用いて、広域分散環境におけるアンビエント計算プログラムの実行方式に関する研究を行った。Safe アンビエントで記述されたプログラム(仕様)の安全性に関する研究は既に数多くなされている。そのため、我々の言語で書かれたプログラムを一旦、Safe アンビエントプログラムに変換してから実行すれば、安全性の保証が容易になるという利点もある。

具体的な成果としては、主に以下の4つが挙げられる。

① Java 言語用アンビエント計算フレームワークの開発

アンビエント計算の近接移動および膜融解の機能を汎用言語である Java から利用するためのフレームワークを構築した。これにより、アンビエント間の並行・協調動作を単純なモデルで表現しつつ、各アンビエント内の処理を Java で記述するという書き分けが可能となり、プログラムの記述性・再利用性が向上した。

② Scheme 言語用アンビエント計算ライブラリ開発

アンビエント計算の機能を Scheme 言語から利用するためのライブラリを構築した。①とは異なり動的かつインタラクティブな Scheme 言語をベースに採用することで柔軟な分散アプリケーションの開発が容易な環境を提供している。たとえば、プログラマブルな分散チーム開発ツールなどは、このライブラリを使った Scheme 対話環境上で容易に実現できる。

さらに、さまざまな実装方式の改良を行った。主な改良点は次のとおり：(1) ノード間コード移動の実現方式を、継続渡し方式から二返戻値法によるスタック巻き戻し方式に変更することで、通常時の計算速度を向上させた、(2) 特定のパターンにしたがうアンビエントの生成を軽量化し、コード移動に伴う通信量や物理スレッド生成数の軽減を行った。

③ Scala 言語用アンビエント計算フレームワークの開発

①で開発した Java 言語用アンビエント計算フレームワークは、各アンビエント内の処理を Java で記述できるという特徴をそなえているが、一方で、コード移動の実装方法が可搬ではない点および局所的な変数アクセルの実現に事前変換処理が必要なため利便性が良くない点が課題として残っていた。それらの課題を解決するための方法の一つとして、Scala 言語をベースにフレームワーク

の再構築を行った。その結果、可搬性、利便性にすぐれたより使いやすいフレームワークを実現できた。また、ここで提案する実現方法は Java フレームワークに対しても(記述性の面で一部妥協が必要となるが)還元することが可能な方式となっている。

④ 大規模分散環境向け実装技術の開発

上記3つのフレームワーク・ライブラリの基盤となる技術として、アンビエント計算の諸機能を大規模分散環境やモバイル環境で実現するための実装方式の設計を行った。この方式がアンビエント計算モデルを正しく実装できていることについての厳密な数学的証明も行った。

さらに、提案する基盤技術を、上記②の Lisp 言語における実装に適用し、その実現可能性・有効性を確認した。実装にあたっては設計レベルでは詳細化していなかった点(コード移動のタイミングの細かな制御など)を考慮し、間欠的なネットワーク遮断にも対応可能な動作を実現できている。

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7件)

- ① 岡田翔太, 馬谷誠二, 林奉行, 八杉昌宏, 湯浅太一, Safe Ambients のための Java フレームワーク, 情報処理学会論文誌 プログラミング, 査読有, 4巻3号, 2011, 26-41, <http://id.nii.ac.jp/1001/00074635/>
- ② Seiji Umatani, Masahiro Yasugi, Taiichi Yuasa, Abstract Machines for Safe Ambients in Wide-Area and Mobile Networks, Proceedings of the 13th International Conference on Coordination Models and Languages, 査読有, LNCS6721, 2011, 172-186, DOI: 10.1007/978-3-642-21464-6_12
- ③ 馬谷誠二, 八杉昌宏, 湯浅太一, 動的名前解決による通信先・移動先の柔軟な指定が可能な分散アンビエントシステムの設計, コンピュータソフトウェア, 査読有, 27巻2号, 2010, 50-61, http://japanlinkcenter.org/JST.JSTAGE/jssst/27.2_50

[学会発表] (計11件)

- ① 岡田翔太, 馬谷誠二, 林奉行, 八杉昌宏, 湯浅太一, Scala Actor ライブラリによる Safe Ambients フレームワークの実装, 情報処理学会第87回プログラミング研究会, 2012年1月23日, 久米島町イーブ情報プラザ(沖縄県)

- ② 林奉行, 馬谷誠二, 八杉昌宏, 湯淺太一, Safe アンビエントに基づく分散アプリケーション開発用 Lisp 環境, 日本ソフトウェア科学会第 28 回大会, 2011 年 9 月 28 日, 沖縄産業支援センター (沖縄県)
- ③ Seiji Umatani, Masahiro Yasugi, Taiichi Yuasa, Abstract Machines for Safe Ambients in Wide-Area and Mobile Networks, 13th International Conference on Coordination Models and Languages (COORDINATION 2011), 2011 年 6 月 7 日, Reykjavik, Iceland
- ④ 林奉行, 馬谷誠二, 八杉昌宏, 湯淺太一, Safe アンビエントに基づく分散アプリケーション開発用 Lisp 環境, 第 13 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 (PPL2011), 2011 年 3 月 9 日, 定山溪ビューホテル (北海道)
- ⑤ 岡田翔太, 馬谷誠二, 林奉行, 八杉昌宏, 湯淺太一, Safe Ambients のための Java ライブラリ, 第 82 回 情報処理学会・プログラミング研究会, 2011 年 1 月 20 日, 宮古島市中央公民館 (沖縄)
- ⑥ 外山真, 馬谷誠二, 八杉昌宏, 湯淺太一, アンビエント計算に基づく Web アプリケーション開発環境, 第 12 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 (PPL2010), 2010 年 3 月 4 日, 琴参閣 (香川県)
- ⑦ 宮本琢也, 馬谷誠二, 八杉昌宏, 湯淺太一, 広域分散環境で動作する Safe アンビエント処理系, 第 12 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 (PPL2010), 2010 年 3 月 3 日, 琴参閣 (香川県)
- ⑧ 馬谷誠二, 八杉昌宏, 湯淺太一, 高級アンビエント計算システムを用いた分散プログラミング, 夏のプログラミング・シンポジウム 2009, 2009 年 9 月 2 日, ラフォーレ那須 (栃木県)

6. 研究組織

(1) 研究代表者

馬谷 誠二 (UMATANI SEIJI)

京都大学・大学院情報学研究科・助教

研究者番号: 40378831