

平成 21 年 5 月 29 日現在

研究種目： 若手研究(B)  
 研究期間： 2006～2008  
 課題番号： 18700026  
 研究課題名（和文） ソフトウェア再利用性向上のための型理論に関する研究

研究課題名（英文） Study on Type Theory to Promote Software Reuse

## 研究代表者

五十嵐 淳 (IGARASHI ATSUSHI)  
 京都大学・大学院情報学研究科・准教授  
 研究者番号： 40323456

研究成果の概要：オブジェクト指向言語においてソフトウェアの再利用性の向上を支援する型システムの研究を行い、主に以下に挙げる成果を得た。(1)相互再帰的なクラスを継承により拡張するための言語機構およびそれに対する型システムを考案し、型システムの安全性の厳密な証明を与えた。(2)再帰的(自己参照的)インターフェースを表現できる型システムに対して従来から指摘されてきた欠点を解決した。(3)漸進的型付けと呼ばれる、動的型付けと静的型付けをひとつの言語上で混在させるための言語機構の形式化を簡単なオブジェクト指向言語に対し行い、その安全性を証明した。

## 交付額

(金額単位：円)

	直接経費	間接経費	合計
2006 年度	1200000	0	1200000
2007 年度	1100000	0	1100000
2008 年度	1100000	330000	1430000
総計	3400000	330000	3730000

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： プログラム言語論・オブジェクト指向・型システム

## 1. 研究開始当初の背景

現代社会において、計算機システムおよび、その上で動作する、ソフトウェアの重要性は増すばかりである。パーソナルコンピュータ、携帯電話、冷蔵庫・電子レンジといった家電などの個人使用の電気・電子機器から、銀行のオンラインシステム、インターネットを介して提供される、ショッピング、確定申告などのサービス、何をとってもソフトウェアは不可欠である。今や、目的に応じた正しく動作するソフトウェアを的確に供給することは、社会が機能するための要件であるといってもよい。

ソフトウェアに対する機能的要請は非常に短期間にも変化が発生しやすい、という特性があり、このような耐えず変化する環境の下で質の高いソフトウェアを製作することは非常に困難である。そのため、ソフトウェアを工業製品のように部品化し、**再利用性**を高めることで、仕様の変化に対しても、一部の部品を取り換えるだけで、新しいソフトウェアを構築できるような技術が必要不可欠である。80年代から盛んになった**オブジェクト指向**によるソフトウェア開発は、クラスとオブジェクトというソフトウェア単位と、継承というソフトウェアの

拡張を支援する仕組みを用いることにより、高い再利用性を目指した手法であり、これまで一定の成功を収めてきた。

一方で、ソフトウェア利用者にとってより重要な性質は**安全性**である。安全性を保証するための技術のひとつとして、型という非常に単純な仕様記述からプログラムの構文的な整合性を検査することによって、ソフトウェアのある種のバグをそのプログラムを実行することなく検知するという**型システム**というプログラミング言語処理系に備える仕組みがある。しかし、これまでのオブジェクト指向プログラミング言語に備わった型システムは、安全性の見積もりが保守的過ぎて、正しく動くソフトウェア部品に対してもバグの可能性を検知し、その結果として、ソフトウェア部品の再利用性が損なわれるという問題が見られる。

## 2. 研究の目的

本研究の主な目的は、近年のソフトウェア開発において中心的に使われている**オブジェクト指向プログラミング言語**のための、**再利用性を向上をサポートする型システム技術**を確立することにある。

高い再利用性を達成するためには、工業部品がより細かい複数の部品を組み合わせることで成り立っているのと同様に、オブジェクト指向言語でのソフトウェア部品の記述単位であるクラスに入れ子構造を導入し、小規模な部品から大規模な部品まで同一の抽象機構で表現できることが重要であると考えている。そのため、本研究では、より具体的に、以下の研究を行った。

1. クラスの入れ子構造を許し、かつ、入れ子構造全体を継承により拡張できるような**拡張されたオブジェクト指向言語を設計**する。
2. 応募者がこれまでに研究してきたオブジェクト指向言語の形式的モデル **Featherweight Java** を用いて、上のような言語のための型システムの**正しさ(安全性と再利用性)**を、**数学的に証明**する。
3. また、**言語処理系のプロトタイプを実装**し、実際に再利用が妨げられている例を記述し、実用性に関する検証を行なう。

## 3. 研究の方法

(1) 入れ子構造を持つクラスのあるオブジェクト指向言語の形式化: そもそも、入れ子構造を導入したオブジェクト指向言語の意味論は自明ではないため、その操作的意味論を数学的に形式化することで、その言

語で書かれたプログラムの挙動に関する知見を得る。

(2) 型システムの設計と再利用性向上の確認: 前項で得られた言語意味論に対する、型システムを考案し、その形式的定義を与える。これまでに記述が困難であると知られている例を記述できることを確認し、十分な再利用性が確保できているかの基準とする。

(3) 型システムの安全性の証明: 前項での研究結果として得られる型システムの安全性を数学的に証明する。設計した型システムが不完全であるという問題がこの時点で判明する可能性もあるが、前年度までに再利用性の確認をしているため、それを損なわない程度に機能を制限し安全性を確保することは、容易になると考えている。

## (4) プロトタイプの実装

上記の理論的な成果があがり次第、プロトタイプ処理系の実装を開始する。

## 4. 研究成果

上記(1)~(3) に関して以下の(a)~(e)のような研究成果を得た。これらの成果などにより、研究代表者は**2006年**に第**20回日本IBM科学賞**(コンピュータサイエンス分野)、**2009年**に文部科学大臣表彰若手科学者賞、と第**1回マイクロソフトリサーチ日本情報学研究賞**(基礎情報学部門)を受賞するにいたった。(4)に関しては実装のための理論的課題が残ったため、簡単なプロトタイプインタプリタの作成に留まった。

(a) **入れ子構造を導入したオブジェクト指向言語の意味論と型理論**: オブジェクト指向言語でのソフトウェア部品の記述単位であるクラスに、小規模な部品から大規模な部品まで同一の抽象機構で表現するために、入れ子構造を導入した言語を設計し、その操作的意味論を数学的に形式化した。また、この言語意味論に対する、型システムを考案し、その形式的定義を与えた。この際、単に安全性を確保するだけではなく、プログラマにとって、より柔軟な抽象化を支援するための、型付けの機構である **variant path types** を考案した。そして、**variant path types** の理論がプログラムの操作的意味論に対して安全性を確保することを証明した。この成果は **FOOL/WOOD, OOPSLA** といったオブジェクト指向に関するトップクラスの国際ワークショップ・会議で発表された(論文[5,8])。その後、**variant path types** を実際の言語に対して適用するための課題として残されていた、型検査アルゴ

リズムの構築とその正しさの証明を行った。これにより、与えられたプログラムが型付けできるかどうかを自動的に判定できるようになる。この成果は未発表であるが上述の結果とあわせて学術雑誌論文として投稿予定である。

**(b) 軽量族多相のためのコンパイル理論と型理論的分析:** 従来から研究をしてきた、拡張可能な相互再帰的クラス定義を支援するための軽量族多相機構を、Java のようなそれを持たない言語で実装するためのコンパイル手法の理論を構築し、そのコンパイル手法の正当性を証明した。この成果は、これまでの軽量族多相の理論とあわせて、プログラミング言語に関する学術雑誌として最も権威のある学術論文誌のひとつである *Journal of Functional Programming* に発表された(論文[3])。また、この機構を Java 5 などに見られる総称クラスと *F bounded* 多相の機構に帰着させた。具体的には、軽量族多相のモデルである *.FJ* から、総称クラスのモデルである *Featherweight GJ* への変換を定義し、変換が型付けと操作的意味を保存することを証明した。この結果、Java 5 の機能だけでは忠実に帰着できないことがわかり、それを補完するための型付け機構を考案した。この研究により、軽量族多相機構の多くの部分が、従来から研究されてきた型理論により説明可能になるとともに、軽量族多相の本質的な利点が明らかになった。本成果は Java などの理論に関する国際ワークショップ *FTfJP*、そして *Journal of Object Technology* に発表された(論文[4,6])。

**(c) オブジェクト指向言語のためのユニオン型の理論:** 従来から研究をしてきた、オブジェクト指向言語におけるユニオン型の理論について考察を進め、既存の言語機構との相互作用について明らかにした。この成果も *Journal of Object Technology* に発表された(論文[7,9])。

**(d) ThisType の理論の実用的側面からの発展:** *ThisType* はオブジェクトが自分自身のインターフェースを指すために使われる型概念であり、軽量族多相と深い関係がある。しかし *ThisType*、軽量族多相ともに、オブジェクト指向言語の特徴である *dynamic dispatch* と相性が悪く、部分型を生かしたプログラミングを妨げている、という問題が指摘されてきた。我々は、この問題に対し、型理論的な考察に基づき、*local exactization*, *nonheritable method* という言語機構による解決を提案した。提案機構

を定式化し、安全性の数学的証明を行った(論文[2])。

**(e) 漸進的型付けの研究:** また、上記の研究から派生して、プログラム再利用性の向上のためには、Ruby, Python といったスクリプト言語のようなプロトタイピングに向いているが、安全ではない言語で書かれたプログラムから、Java のような安全な言語にスムーズにプログラムを進化させるための機構が必要である、という発想に至り、漸進的型付けの枠組みの研究を行った。具体的には、我々は *Siek* と *Taha* による、両者の長所を活かすための漸進的型付け (*gradual typing*) の枠組みを Java のようなオブジェクト指向言語に適用するための基盤理論を研究し、動的型付け・静的型付けされるコードの混在を許しながらも、実行時エラーは静的型付けされた部分からは発生しない、という型システムの構築に成功した。これは、動的型付けされたコードを安全に再利用するための枠組みとして利用可能である。本成果は国内雑誌に発表された(論文[1])。ただし、ジェネリクスを含む言語に関しては今後の検討課題として残った。

## 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 10 件)

[1] 伊奈 林太郎, 五十嵐 淳. Featherweight Java のための漸進的型付け. *コンピュータソフトウェア* 26(2):18-40, 2009 (査読有)

[2] Chieri Saito and Atsushi Igarashi. Matching *ThisType* to subtyping. In *Proceedings of the 24th Annual ACM Symposium on Applied Computing (SAC2009)*, pages 1851-1858, 2009. (査読有)

[3] Chieri Saito, Atsushi Igarashi, and Mirko Viroli. Lightweight family polymorphism. *Journal of Functional Programming*, 18(3):285-331, 2008. (査読有)

[4] Chieri Saito and Atsushi Igarashi. The essence of lightweight family polymorphism. *Journal of Object Technology*, 7(5):67-99, 2008. (査読有)

[5] Atsushi Igarashi and Mirko Viroli. Variant path types for scalable extensibility. In *Proceedings of the ACM Conference on Object-Oriented Programming*,

*Systems, Languages, and Applications (OOPSLA 2007)*, pages 113-132, 2007. (査読有)

[6] Chieri Saito and Atsushi Igarashi. The essence of lightweight family polymorphism. In *Proceedings of the 9th Workshop on Formal Techniques for Java-like Programs (FTfJP 2007)*, pages 27-41, 2007. (査読有)

[7] Atsushi Igarashi and Hideshi Nagira. Union types for object-oriented programming. *Journal of Object Technology*, 6(2):47-68, 2007 (査読有)

[8] Atsushi Igarashi and Mirko Viroli. Variant path types for scalable extensibility. In *Proceedings of the International Workshop on Foundations and Developments of Object-Oriented Languages (FOOL/WOOD 2007)*, pages 38-49, 2007 (査読有)

[9] Atsushi Igarashi and Hideshi Nagira. Union types for object-oriented programming. In *Proceedings of the 21st Annual ACM Symposium on Applied Computing (SAC2006)*, pages 1435-1441, 2006. (査読有)

[10] Atsushi Igarashi and Mirko Viroli. Variant parametric types: A flexible subtyping scheme for generics. *ACM Transactions on Programming Languages and Systems*, 28(5):795-847, 2006. (査読有)

〔学会発表〕 (計 6 件)

[1] Chieri Saito and Atsushi Igarashi. Matching *ThisType* to subtyping. The 24th Annual ACM Symposium on Applied Computing (SAC2009), pages 1851-1858, 2009 年 3 月. アメリカ合衆国ホノルル.

[2] 伊奈 林 太郎, 五十嵐 淳. Featherweight Java のための漸進的型付け. 第 10 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(PPL2008), 2008 年 3 月 5 日. 宮城県仙台市.

[3] Atsushi Igarashi and Mirko Viroli. Variant path types for scalable extensibility. The ACM Conference on Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA 2007), 2007 年 10 月. カナダ・モントリオール.

[4] Chieri Saito and Atsushi Igarashi. The essence of lightweight family polymorphism. The 9th Workshop on Formal Techniques for Java-like Programs (FTfJP

2007), 2007 年 7 月. ドイツ・ベルリン.

[5] Atsushi Igarashi and Mirko Viroli. Variant path types for scalable extensibility. The International Workshop on Foundations and Developments of Object-Oriented Languages (FOOL/WOOD 2007), 2007 年 1 月. フランス・ニース.

[6] Atsushi Igarashi and Hideshi Nagira. Union types for object-oriented programming. The 21st Annual ACM Symposium on Applied Computing (SAC2006), 2006 年 4 月. フランス・ディジョン.

〔図書〕 (計 1 件)

[1] 五十嵐 淳. プログラミング in OCaml ~関数型プログラミングの基礎から GUI プログラミングまで. 374 ページ. 技術評論社, 2007

〔その他〕

ホームページ等

<http://www.sato.kuis.kyoto-u.ac.jp/~igarashi/>

## 6. 研究組織

### (1) 研究代表者

五十嵐 淳 (IGARASHI ATSUSHI)

京都大学・大学院情報学研究所・准教授  
研究者番号： 40323456