

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25年 6月12日現在

機関番号：12601

研究種目：基盤研究（B）

研究期間：2009～2012

課題番号：21300006

研究課題名（和文） アスペクト指向プログラミングの展開的研究

研究課題名（英文） Expanding Aspect-Oriented Programming

研究代表者

増原 英彦 (HIDEHIKO MASUHARA)

東京大学大学院総合文化研究科・准教授

研究者番号：40280937

研究成果の概要（和文）：横断的関心事や環境に適応した振舞をモジュール化するアスペクト指向プログラミング・文脈指向プログラミング言語に関して、基礎的な計算体系の構築・型システムの改良による記述力の向上・セキュリティ応用のための言語提案・事象に基づく制御機構を持った言語提案による応用範囲の拡大といった研究を行った。これらの提案は Java, AspectJ を拡張した言語処理系を作成し、実際のアプリケーションプログラム記述を通して有効性を検証している。

研究成果の概要（英文）：We conducted research on aspect-oriented programming and context-oriented programming on several aspects, including construction of formalized semantics, improvement of expressiveness through revised type systems, proposal of language constructs for security concerns, and expanding application area through development of a novel language based on events. Those proposals are validated by implementing language systems by extending existing languages like Java and AspectJ, and by rewriting practical application programs.

交付決定額

(金額単位：円)

	直接経費	間接経費	合計
2009年度	2,400,000	720,000	3,120,000
2010年度	2,700,000	810,000	3,510,000
2011年度	2,500,000	750,000	3,250,000
2012年度	1,900,000	570,000	2,470,000
総計	9,500,000	2,850,000	12,350,000

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：横断的関心事・アスペクト指向・文脈指向・セキュリティ・意味論・

型安全性

1. 研究開始当初の背景
今日の社会基盤の重要な部分を構成するソフトウェアシステムは、その規模の拡大と

ともに、開発や保守を容易にするためにソフトウェアのモジュール化が重要な技術となっ

てきている。モジュール化の重要性は古くから指摘されており、オブジェクト指向プログラミング(OOP)のように階層性を利用したモジュール化機構は大きな成功を収めている。

これまでのモジュール化機構では解決できていなかった問題に、横断的関心事がある。例えば実行記録(ロギング)、例外処理、認証や暗号化などのセキュテリィ処理などは、従来の技術では多数のモジュールに渡った記述を強いられ、ソフトウェアの理解や保守を困難していた。アスペクト指向プログラミング(AOP)は横断的関心事のモジュール化に注目した技術として提唱された。プログラム中で特定の動作が起きた場合の追加・代替動作を指定するポイントカット・アドバイス機構を備えた AOP 言語 AspectJ の開発などにより、研究者と産業界の両方から注目されるようになってきている。

AOP 言語に対する注目度は高い一方で、現状はその基礎付けや、機能面での充実は充分に行われているとは言えない。研究代表者らはそれらの点について、AOP に対する基礎的な理論を与える研究およびポイントカット・アドバイス機構の機能拡張に関する研究を行ってきている。

AOP は産業界での応用が進んでいる。実際、大規模なアプリケーションフレームワークを AOP 言語を用いて再構築した例や、ワークステーションクラスタ向け Java 仮想機械の実現に AOP を用いた例や、JBoss や Spring などのアプリケーションフレームワークでの利用はよく知られているが、そのほとんどが AspectJ 言語を基本とした、保守的な AOP 言語によるものである。研究代表者らがこれまで研究してきたような用いた AOP 言語の

拡張機能を利用した応用はほとんど行われていない。

また AOP をとりまく研究動向として、近年モジュール化機構全般について理論から工学的側面に至るまでの分野で注目が高まっている。実際、プログラミング言語やソフトウェア工学に関する国際会議である POPL, OOPSLA, ECOOP, GPCE, ICSE などでは、近年、AOP をはじめとして *mixin layers*, *feature* 指向プログラミング, *traits*, 文脈指向プログラミング(COP), 生成的プログラミング, *family polymorphism* などの研究が数多く発表されているが、これらの研究はどれもモジュール化機構に注目しているものである。しかしながらほとんどの提案は、目的は共通するが手法が大きく異なる機構だと認識されており、お互いの類似性や共通性について深い考察をするには至っていない。

2. 研究の目的

これまで研究代表者らが行ってきた研究を出発として、AOP およびそれを取り巻く研究分野の動向を受けて、実証面、基礎面、および新展開 3 つの方面にそれぞれ発展させることを目標とする。具体的には以下のテーマについて研究を行う。

(i)実証面: 実用的なソフトウェアに対して AOP 言語の拡張機構を適用し、その有効性や問題を具体的に示し、AOP 言語の実用化にあたっての課題を明らかにする。具体的には《データ流ポイントカットを備えた Ruby 言語の AOP 拡張》を作成し、セキュリティ関連の処理を行うアスペクトの実用性を検証する。さらに、定義静的解析を用いたポイントカットを記述できる SCoPE コンパイラを利用して《動的検査を行うアスペクトの最適化》に対する効果を実証する。

(ii)基礎面: ポイントカット・アドバイス機構の型安全性について、安全性を損わずに適用可能範囲を拡大する方式である《型緩和織込》を提案し、その定式化と処理系試作を通じた有用性の検証を行う。純関数型言語に実用的な AOP 機構の追加した《AOP Haskell》言語を設計し、副作用や遅延評価を AOP によってどのように扱うべきかを明らかにするとともに、実用性のある言語処理系を作成・公開する。さらにこれまでに提案した継続に基いた結合点モデルを用いて、AOP 言語のコンパイルを整理した形で行う《Compiling with Continuation Join Points》方式の提案と、処理系の試作を行う。

(iii)新展開: これまで AOP とは独立に考えられてきた言語機構について、AOP との相違点を明らかにし、新しいモジュール化機構の創出に向けた研究を行う。そのための具体的なアプローチとして、文脈指向プログラミング(COP) と AOP の言語機構の相異点を整理し、両者を融合させた言語機構の提案と処理系試作を行う《COP と AOP の融合》の研究を行う。また、近年注目を集めている静的解析と動的検査を併用したプログラム検証技術と、AOP によるプログラム検証の関係を整理し、《プログラム検証技術と AOP の融合》による新しい言語パラダイムを提案を行う。

3. 研究の方法

本研究は、各テーマ毎にある程度独立して研究を進めることができるため、以下それぞれについて計画設定した。大部分のテーマは、研究代表者らのこれまでの研究を出発点として開始し、処理系の作成や評価実験を進めてゆく。《型緩和織込》《COP と AOP の融合》《プログラム検証技術と AOP の融合》は、従来の技術の整理を行いつつ、研究協力

者との間で議論を重ねて新しい提案を行ってゆく。

4. 研究成果

(1) アスペクト指向言語 AspectJ の型システムをより柔軟にする手法を複数提案した。具体的には、AspectJ の型システムに存在するいくつかの制限が強過ぎるため、いくつかの応用を不可能にしていたり不自然なプログラミングを強いられている場面があることを実例に即して指摘した。さらにこの問題を緩和するために、型緩和織込と総称アスペクトの2つの型システムに関する機構を提案した。前者は値を置換するようなアドバイスが同一結合点に複数適用される場合の問題を解決するもので、後者は Java 言語に存在する総称型(generic type)をアスペクト内部で使用した場合に起きる問題を解決する。これらの提案機構は既存の AspectJ コンパイラを拡張した処理系によって現実のプログラムに適用可能であることを示した。これらの提案は型システム上の細かな問題点を解決する研究ではあるが、AspectJ 言語はすでに広範なソフトウェア開発に利用されているため、その制限を型システムという基礎的な研究から緩和することができた意義は大きい。(雑誌論文⑥, ⑩, ⑮, ⑰, 学会発表①, ⑤, ⑥)。

(2) 分散システムにおけるデータの秘匿性を保証するためのアクセス制限機構を、アスペクトによって実現するための言 AspectKE*を設計・実現した。研究はデンマーク工科大学の Yang 氏、F. Nielson 教授、H. Nielson 教授との共同研究として進めた。医療データを扱うようなシステムでは、データのアクセスをきめ細かく行う必要がある。特に、利用者がアクセスできるデータに制限を加えるだけでなく、そこで得た情報がさらに他の利用者に渡らないよう、情報流を意識した制限を行

わなければいけない。本研究では分散タプル空間計算体系上にアスペクト指向拡張を行い、情報流を考慮したアクセス制限をアスペクトとして分離して記述させることに成功した。また計算体系の実現方法として、実行前に静的なデータ依存解析を行い、それを実行時の動的な情報流追跡と組み合わせる、より効率的な手法を提案した。これに基づいた Java 言語への変換器を試作し、いくつかの医療アプリケーションから採用した例題におけるアクセス制限を実現できることを示した。この研究は、分散タプル空間という理論的なモデルに基礎を置いているため、実際のシステムに適用するにはさらなる検討が必要となるが、この種の応用に対してアスペクト指向の可能性を示すことができたと言える(雑誌論文⑧, ⑩)。

(3) Java に基づく文脈指向プログラミング ContextJ の設計と実現の研究を行った。携帯端末や対話的なソフトウェアでは、ソフトウェアが置かれている状況に応じて各部分の振舞を変更する必要がある。文脈指向プログラミングは、そのような多様な振舞を、層と部分メソッドによって分離することを可能にする。

この研究は、これまでに Smalltalk や Lisp のような動的型付言語で検討されていた文脈指向プログラミングを、静的型付言語である Java 言語上で実現するための言語設計と実現方式の提案を行ったものである。Java 言語は Android 端末のアプリケーションや、アプリケーションサーバの記述言語として広く用いられているため、そのような言語での標準的な言語設計と効率的な実現方式を示した意義は大きい。なお、この研究は独ポツダム大学 Hasso-Plattner-Institut Appeltauer 氏、

Hirschfeld 教授らとの共同研究として進めた。(雑誌論文⑫, ⑬, ⑭, 学会発表⑦)

(4) 文脈指向プログラミング言語の形式化として ContextFJ 計算体系を作成し、基本的な型安全性の証明を行った。前項(3)で述べたように文脈指向プログラミングに対する研究は、注目が高まっている一方で、基礎理論の整備がほとんど行われていなかった。そのため多くの言語機能に関する議論は、実際の言語処理系の振舞を参照する以外の方法で行うことができていなかった。本研究では、そのような文脈指向プログラムの意味論を形式化することで、今後の議論を行う土台とする狙いを持って行われた。具体的には京都大学の五十嵐教授と独ポツダム大学

Hasso-Plattner-Institut の Hirschfeld 教授と共同で、Featherweight Java 流の形式的意味論を記述し、その型安全性の証明を行った。

記述対象とした言語機能は既存の文脈指向言語 ContextJ を参考にして行った。簡潔な形式化に成功したため、意味論は比較的自明なものに見えるが、現在これを基にしてより自由度の高い言語機能を導入した場合でも型安全性を保つことができる見通しを得るなど、今後の応用が広い研究だと言える。(雑誌論文④, ⑪, 学会発表③)

(5) 文脈指向プログラミング言語の適用対象を広げるため、新たに事象に基づく文脈切替手法を提案し、その実現として EventCJ 言語を設計・処理系の試作を行った。既存の文脈指向言語では文脈の切り替えを、プログラムの制御構造に沿う形で行っていたため、フレームワークを利用したプログラムや、外部事象によって文脈切替が起きるプログラムでは、不自然なプログラミングを強いられることとなっていた。本研究では事象(イベント)によって文脈を変更するという考え方にい

ち早く着目し、それに基づいた言語の設計と実現を行った。研究終了時点では、モデル検査による正しさの検証・ソフトウェア開発方法論を含めた設計方式の検討・他の文脈指向言語の言語機能の包摂方法の検討など、設計から応用まで広い範囲に研究が展開されており、今後さらなる成果を挙げることが期待できる。(雑誌論文①, ②, ③, ⑤, ⑦, ⑨, ⑭, 学会発表②, ④)。

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 18 件)

- ① 青谷知幸、紙名哲生、増原英彦、オブジェクト毎の層遷移を宣言的に記述できる文脈指向言語 EventCJ、コンピュータソフトウェア、査読有、採録決定済み、2013、<https://www.jstage.jst.go.jp/browse/jssst-char/ja>
- ② Tetsuo Kamina, Tomoyuki Aotani, Hidehiko Masuhara, A core calculus of composite layers, Proceedings of Foundations of Aspect-Oriented Languages (FOAL2013), 査読有、2012、pp.7-12、DOI:10.1145/2451598.2451601
- ③ Tetsuo Kamina, Tomoyuki Aotani, Hidehiko Masuhara, Introducing composite layers in EventCJ, IPSJ Transaction on Programming, 査読有、6(1)、2013、pp.1-8、http://www.ipsj.or.jp/e-library/digital_library.html
- ④ Atsushi Igarashi, Robert Hirschfeld, Hidehiko Masuhara, A type system for dynamic layer composition, Proceedings of 19th International Workshop on Foundations of Object-Oriented Languages (FOOL 2012), 査読有、2012、pp.13-24、<http://www.cs.uwm.edu/~boyland/fool2012/>
- ⑤ Tetsuo Kamina, Tomoyuki Aotani, Hidehiko Masuhara, Bridging real-world contexts and units of behavioral variations by composite layers, Proceedings of the Workshop on Context-Oriented Programming (COP'12), 査読有、2012、pp. 4:1-4:6、DOI:10.1145/2307436.2307440
- ⑥ Manabu Toyama, Tomoyuki Aotani, Hidehiko Masuhara, A per-type instantiation mechanism for generic aspects, Proceedings of the 3rd International Workshop on Variability and Composition (VariComp'12), 査読有、2012、pp.25-30、DOI: 10.1145/2161996.2162003
- ⑦ Tomoyuki Aotani, Tetsuo Kamina, Hidehiko Masuhara, Featherweight EventCJ: a core calculus for a context-oriented language with event-based per-instance layer transition, Proceedings of the Workshop on Context-Oriented Programming (COP'11), 査読有、2011、pp. 1:1-1:7、DOI:10.1145/2068736.2068737
- ⑧ Fan Yang, Hidehiko Masuhara, Tomoyuki Aotani, Flemming Nielson, Hanne Riis Nielson, Combining static analysis and runtime checking in security aspects for distributed tuple spaces, 13th International Conference on Coordination Models and Languages (Lecture Notes in Computer Science), 査読有、6721 巻、2011、pp.202-218、ISBN: 978-3-642-21463-9
- ⑨ Tetsuo Kamina, Tomoyuki Aotani, Hidehiko Masuhara, EventCJ: A context-oriented programming language with declarative event-based context transition, Proceedings of the 10th International Conference on Aspect-Oriented Software Development

- (AOSD'11)、査読有、2011、pp.253-264、
DOI:10.1145/1960275.1960305
- ⑩ Tomoyuki Aotani, Manabu Toyama, Hidehiko Masuhara, Supporting covariant return types and generics in type-relaxed weaving, Proceedings of Foundations of Aspect-Oriented Languages (FOAL2011)、査読有、2011、pp.13-17、
DOI:10.1145/1960510.1960516
- ⑪ Robert Hirschfeld, Atsushi Igarashi, Hidehiko Masuhara, ContextFJ: A minimal core calculus for context-oriented programming, Proceedings of Foundations of Aspect-Oriented Languages (FOAL2011)、査読有、2011、pp.25-29、
DOI:10.1145/1960510.1960515
- ⑫ Malte Appeltauer, Robert Hirschfeld, Michael Haupt, Hidehiko Masuhara, ContextJ: Context-oriented programming with Java、コンピュータソフトウェア、査読有、28(1)巻、2011、pp.272-292、
<https://www.jstage.jst.go.jp/browse/jssst/-char/ja>
- ⑬ Malte Appeltauer, Robert Hirschfeld, Hidehiko Masuhara, Michael Haupt, Kazunori Kawauchi, Event-specific software composition in context-oriented programming, Proceedings of the Conference on Software Composition (SC'10) (Lecture Notes in Computer Science)、査読有、6144 巻、2010、pp.50-65、
DOI:10.1007/978-3-642-14046-4_4
- ⑭ Tetsuo Kamina, Tomoyuki Aotani, Hidehiko Masuhara, Designing event-based context transition in context-oriented programming, Proceedings of 2nd International Workshop on Context-Oriented Programming (COP'10)、査読有、2010、pp.7-12、
DOI:10.1145/1930021.1930023
- ⑮ Hidehiko Masuhara, Atsushi Igarashi, Manabu Toyama, Type relaxed weaving, Proceedings of the 9th International Conference on Aspect-Oriented Software Development (AOSD'10)、査読有、2010、pp.121-132、
DOI:10.1145/1739230.1739245
- ⑯ Fan Yang, Hidehiko Masuhara, Tomoyuki Aotani, Flemming Nielson, Hanne Riis Nielson, AspectKE*: Security aspects with program analysis for distributed systems, Proceedings of the 9th Workshop on Aspects, Components, and Patterns for Infrastructure Software (ACP4IS'10)、査読有、2010、pp.27-31、
<http://opus.kobv.de/ubp/volltexte/2010/4122>
- ⑰ Tomoyuki Aotani, Manabu Toyama, Hidehiko Masuhara, StrongRelaxAJ: integrating adaptability of RelaxAJ and expressiveness of StrongAspectJ, Proceedings of Foundations of Aspect-Oriented Languages (FOAL2010)、査読有、2010、pp.1-4、
<http://www.eecs.ucf.edu/FOAL/index-2010.shtml>
- ⑱ Malte Appeltauer, Robert Hirschfeld, Hidehiko Masuhara, ContextJ: Improving the development of context-dependent Java applications, Proceedings of the International workshop on Context-oriented Programming (COP'09)、査読有、2009、
DOI:10.1145/1562112.1562117
- [学会発表] (計 7 件)
- ① Hidehiko Masuhara, Reflection on the power of pointcuts, Concurrent Objects and Beyond: From Theory to High-Performance Computing、2012 年

- 5月29日、神戸大学
- ② 紙名哲生, 青谷知幸, 増原英彦, JavaCat: Realizing context as fluent、第15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(PPL2013)、2013年3月5日、福島県会津若松 東山温泉「御宿東鳳」
 - ③ 井上裕昭, 五十嵐淳, Robert Hirschfeld, 増原英彦, 動的レイヤー合成のための型システム、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(PPL2013)、2013年3月4日、福島県会津若松 東山温泉「御宿東鳳」
 - ④ 青谷知幸, 紙名哲生, 増原英彦, 文脈指向プログラムの資源利用解析に基づく層活性解析法、情報処理学会第89回プログラミング研究会発表:2012-1-(4)、2012年6月、小樽市民センター
 - ⑤ Tomoyuki Aotani and Hidehiko Masuhara, A join point model for implicit conversion, Proceedings of the 5th Asian Workshop on Aspect-Oriented Software Development (AOAsia 5)、2009年11月17日、Auckland, New Zealand
 - ⑥ Hidehiko Masuhara, Can we use AOP for patching?: A weakness in AspectJ's type system and a remedy, a talk at the workshop on Trade-offs in Software Language Engineering、2009年6月11日、University of Twente, Netherlands
 - ⑦ Malte Appeltauer, Robert Hirschfeld, Michael Haupt, Hidehiko Masuhara, ContextJ: Context-oriented programming with Java、日本ソフトウェア科学会全国大会第26回大会、2009年9月16日、島根大学

[図書] (計 0 件)

[その他]

ホームページ等

<http://www.graco.c.u-tokyo.ac.jp/~masuhara/papers/index-j.html>

<http://www.graco.c.u-tokyo.ac.jp/~masuhara/papers/>

6. 研究組織

(1)研究代表者

増原 英彦 (MASUHARA HIDEHIKO)
東京大学大学院総合文化研究科・准教授
研究者番号：40280937

(2)連携研究者

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