

**科学研究費助成事業 研究成果報告書**

平成 27 年 6 月 9 日現在

機関番号：12608

研究種目：若手研究(B)

研究期間：2011～2014

課題番号：23700033

研究課題名(和文)安全で容易なモジュール合成のためのプログラミング言語機構の研究

研究課題名(英文)A study on programming language mechanisms for easy-to-use safe module compositions

## 研究代表者

青谷 知幸 (Aotani, Tomoyuki)

東京工業大学・情報理工学(系)研究科・助教

研究者番号：20582919

交付決定額(研究期間全体)：(直接経費) 2,700,000円

研究成果の概要(和文)：モジュールとは、プログラムを構成する再利用可能な部品のことである。例えば関数やデータ構造、クラス、パッケージがモジュールとなる。本研究ではプログラムのより良いモジュール化のための先進的プログラミング手法であるアスペクト指向プログラミングと文脈指向プログラミングに焦点を当て、モジュールを安全かつ容易に合成するためのプログラミング言語機構の研究に取り組んだ。アスペクト指向プログラミングについては型安全性を保证するためのインターフェースを提案し、文脈指向プログラミングについては宣言的な文脈遷移規則の記述法と、異なる文脈遷移機構を一つのプログラミング言語の中で共存させるための理論的基礎を築いた。

研究成果の概要(英文)：Modules are independent and interchangeable parts of a program. Functions, data structures, classes and packages are typical examples. This study focused on programming techniques for advanced modularization, namely aspect-oriented programming (AOP) and context-oriented programming (COP), and addressed to develop programming language mechanisms for declarative and safe compositions of modules. We proposed interfaces for AOP called ITD interfaces to assure type safety. We also developed (1) a COP language called EventCJ that supports declarative and composable context transition rules and (2) core calculi to let a single COP language support multiple context activation mechanisms using context holders.

研究分野：プログラミング言語の設計と実装

キーワード：モジュラリティ プログラミング言語 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

## 1. 研究開始当初の背景

プログラムを互いに独立な機能ごとに分割することをモジュラープログラミングといい、分割物をモジュールという。モジュールに分割されたプログラムは並行開発がしやすく、保守性が高いという利点がある。

研究を始めた 2011 年、プログラムのモジュール化に関する研究が注目を集めていた。この分野では最高峰の国際会議である AOSD は 2002 年に産声を上げたが、当時モジュール化技術として最も注目を集めていたのは、アスペクト指向プログラミング (以降 AOP と略す) であった。AOSD という名前はアスペクト指向ソフトウェア開発 (Aspect-Oriented Software Development) の頭文字を取ったものである。以来様々な研究者によってモジュール化の研究が精力的に行われ、2011 年までにアスペクト指向以外の様々な技術が登場した。実行環境やプログラムの状態に応じて振舞いや提供する機能の変わるプログラムのためのモジュール化技術である文脈指向プログラミング (以降 COP と略す) はその 1 つである。モジュール化技術を広く扱う国際会議であることを明確にするために、AOSD は MODULARITY へと名前を変えようとしていた。

## 2. 研究の目的

本研究の目的は宣言的かつ安全なモジュール合成を実現するプログラミング言語機構の実現である。

分割されたモジュールを組み合わせてソフトウェアを作り上げるためには、モジュールの合成法が必要不可欠である。モジュール合成を誤ると、ソフトウェアは期待通りに動作しなくなる。しかし、細かく分割されたモジュールを誤りなく組み合わせるのは困難であった。宣言的モジュール合成は直感的なモジュール合成の指定を可能にする。また、安全なモジュール合成はモジ

ュールを合成した結果、プログラムの期待通りの振舞いをするソフトウェアになっていることを保証するものである。

## 3. 研究の方法

本研究では先進的モジュール化技術として AOP と COP に焦点を当てた。

AOP はプログラム中に散在して書かれる処理を一カ所に纏めてプログラムの管理や変更を行いやすくするプログラミング手法で、近年学术界と産業界の双方から注目を集めている。

COP はプログラムの提供する機能や機能の振舞いが、プログラムの実行環境やプログラム内の状態などの文脈に依存して変化するようなプログラムのための、モジュール化技術である。例えばスマートフォン向けのアプリケーション管理プログラム (Google Play など) では、Wifi 通信が可能な状態か否か、電池残量が十分か否かに応じて、新しいバージョンのアプリケーションが公開されているかどうかを検査する機能 (以降更新検査という) と、新しいバージョンのアプリケーションのインストールを実行する機能 (以降更新実行という) の利用可能性が変わる。Wifi 通信が可能でないとき、更新検査は利用できない。また電池残量が十分でない時、更新実行は利用できない。文脈ごとに利用できる機能の種類や機能自体の振舞いが異なるため、これらを文脈ごとに定義する記述法と、プログラム実行時の文脈に応じて文脈ごとの定義を合成する手法が必要である。これまでに提案されてきた COP 言語に倣って、定義法には層を、合成法には層活性機構を用いることにした。

## 4. 研究成果

2011 年度には COP 言語とイベント

機構の融合技術の形式的定義，および AOP 言語の表現力の向上とロバスト化技術に取り組んだ。

COP 言語のプログラマは文脈毎の振舞を独立に定義し，どの振舞を実行するかを指定する．本研究ではプログラム実行中に起こるイベントに反応して実行する振舞を切り替える仕組みを持つ文脈指向言語 EventCJ について，振舞を切り替えるための領域特化言語を作り，その意味を Featherweight Java を土台として形式的に定義した．成果を纏めた論文は文脈指向プログラミングに関する国際ワークショップ COP'11 に採択され，口頭発表した．また日本ソフトウェア科学会第 28 回大会と，国際ワークショップ AOAsia/Pacific'11 で口頭発表した．

AOP 言語の表現力が向上すると，より多くの散在する処理が纏められる．本研究では状態とそれに処理を一つに纏めるための機構を導入することで，アスペクト指向言語の表現力を向上させた．成果を纏めた論文はプログラムの組み合わせに関する国際ワークショップ VariComp'11 に採択された．プログラムをロバストにすると，プログラムの書き換えを行う際，変更を局所化できる．ソフトウェアの段階的な開発・進化が行いやすい．本研究では近年学术界と産業界の双方から注目を集めているアスペクト指向プログラミングで開発されたプログラムをロバスト化するプログラム変換技術を開発した．成果を纏めた論文は国際ワークショップ RAMSE'11 に採択された．

2012 年度には文脈指向プログラムの最適化と AOP 言語の表現力の向上，そして非アスペクト指向プログラムからアスペクト指向プログラムへの自動的な変換のための基礎技術に取り組んだ．文脈指向プログラムでは 1 つの関数名に対して複数の定義が存在し，定義の選択をプログラム実行中に

自由に変えることができる．従って関数を実行する際には逐一，現在選択されている定義を探し出さねばならず，そのための計算時間が必要になる．本研究で提案した最適化は，関数名と定義の組みが一通りに定まる部分をプログラムの実行前に見つけ出し，実行中に定義を探し出す時間をなくすものである．成果をまとめた論文は論文誌「コンピュータソフトウェア」に採択された．

AOP 言語の表現力の向上は，プログラムの設計情報をプログラム中に取り込むという発想に基づく．より具体的には，本研究ではプログラムが提供する機能の間のつながりを直接プログラムとして記述するための言語機構を提案した．機能の間のつながりを直接プログラムとして記述できるようにしたことで，プログラムの正しさの検証とプログラムの理解が容易になるため，プログラムの生産・保守性が向上する．成果をまとめた論文は AOAsia/Pacific'13 に採択された．

アスペクト指向プログラムへの変換とは，複雑にもつれた保守・拡張の困難なプログラムを，機能毎に分かれた保守・拡張の容易なプログラムに変換することである．この自動化のためには，プログラムを構成する部品を機能毎に分類することが必要不可欠である．本研究では自己組織化写像を用いた自動的な分類法を提案し，成果をまとめた論文は VariComp'13 に採択された．

2013 年度には AOP 言語におけるインターフェース定義の見直しと改良，そして COP 言語における層活性機構の統一に関する研究を行った．

型間宣言は AOP 言語が提供する機構の一つで，既存のクラスや構造体に新しいメソッドやフィールドが追加できる．だが型間宣言を用いてしまうと，モジュ

ールごとのコンパイルができなかった。従って、モジュールの合成も不可能である。本研究では型間宣言のインターフェースを定義することでモジュールごとのコンパイルとコンパイルされたモジュールの合成を可能にした。代表的な AOP 言語である AspectJ 言語に型間宣言のためのインターフェースを導入し、モジュールごとのコンパイルができない既存の AspectJ プログラムをこのインターフェースを用いるように書き換えて、モジュールごとのコンパイルができることを確かめた。

層活性機構はプログラム実行時に振舞を切り替えるための機構である。層活性機構には振舞の切り替えに関して得手不得手があり、理想的な COP 言語は複数の層活性機構をプログラマが選択的に使うことができるものである。本研究では複数の層活性機構の混ぜあわせを可能にするための理論的基礎を構築した。成果をまとめた論文は国際ワークショップ FOAL'14 に採択された。

2014 年度は引き続いて COP 言語における層活性機構の統一についての理論的基礎を研究した。これまでに 4 種類の層活性機構が提案されているが、それぞれには得手不得手があり、理想的な COP 言語は複数の層活性機構をプログラマが選択的に使うことができるものである。本研究では昨年度に構築した複数の層活性機構の混ぜあわせを可能にするための理論的基礎を深化させ、文脈指向プログラミング言語のための計算体系である ContextFJ の上に再構築した。この成果をまとめた論文は国際ワークショップ COP'14 に採択された。またこの成果は Java を拡張した文脈指向プログラミング言語 ServalCJ の設計と、ServalCJ を形式化した Featherweight ServalCJ に役立てられた。ServalCJ を用いたソフトウェア開発法についての論文

は国際会議 MODULARITY'14 に、ServalCJ 言語についての論文は国際会議 MODULARITY'15 にそれぞれ採択された。

#### 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 2 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, Vol.30, No.3, pp.130-147, 2013.

② Tetsuo Kamina, Tomoyuki Aotani, Hidehiko Masuhara, ``Introducing Composite Layers in EventCJ'', IPSJ Transactions on Programming, 査読有, Vol.6, No.1, pp.1-8, 2013.

[学会発表] (計 12 件)

① Tetsuo Kamina, Tomoyuki Aotani, Hidehiko Masuhara and Tetsuo Tamai., ``Generalized layer activation mechanism through contexts and subscribers'', In Proceedings of the 14th International Conference on Modularity (MODULARITY2015), pp.14-28, March, 18, 2015, Fort Collins, USA.

② Lin Wang, Tomoyuki Aotani and Masato Suzuki, ``Improving the Quality of AspectJ Application: Translating Name-Based Pointcuts to Analysis-Based Pointcuts'', In Proceedings of 14th International Conference on Quality Software

(QSIC 2014), pp.27-36, October 2, 2014, Dallas, USA

③ Tetsuo Kamina, Tomoyuki Aotani, and Atsushi Igarashi, "On-Demand Layer Activation for Type-Safe Deactivation", Proceedings of international workshop on Context-Oriented Programming (COP'14), Article No.4, July 29, 2014, Uppsala, Sweden.

④ Tomoyuki Aotani, Tetsuo Kamina, and Hidehiko Masuhara, "Unifying Multiple Layer Activation Mechanisms Using One Event Sequence", Proceedings of international workshop on Context-Oriented Programming (COP'14), Article No.2, July 29, 2014, Uppsala, Sweden.

⑤ Tetsuo Kamina, Tomoyuki Aotani, Hidehiko Masuhara, and Tetsuo Tamai, "Context-Oriented Software Engineering: A Modularity Vision", Proceedings of international conference on Modularity (MODULARITY'14), pp.85-98, April 24, 2014, Lugano, Switzerland.

⑥ Tomoyuki Aotani, Tetsuo Kamina, and Hidehiko Masuhara, "Context holders: realizing multiple layer activation mechanisms in a single context-oriented language", Proceedings of international workshop on Foundations of Aspect-Oriented Languages (FOAL'14), pp.3-6, April 22, 2014, Lugano, Switzerland.

⑦ Tetsuo Kamina, Tomoyuki Aotani, and Hidehiko Masuhara, "A unified context activation mechanism", Proceedings of international workshop on Context-Oriented Programming

(COP'13), Article No.2, July 2, 2013, Montpellier, France.

⑧ Tetsuo Kamina, Tomoyuki Aotani, and Hidehiko Masuhara, "A core calculus of composite layers", Proceedings of international workshop on Foundations of Aspect-Oriented Languages (FOAL'13), pp.7-12, March 26, 2013, Fukuoka International Congress Center, Fukuoka, Japan.

⑨ Manabu Toyama, Tomoyuki Aotani, and Hidehiko Masuhara, "A per-type instantiation mechanism for generic aspects", Proceedings of international workshop on Variability & Composition (VariComp'12), 25-30, March 26, 2012, Potsdam, Germany.

⑩ Tetsuo Kamina, Tomoyuki Aotani, and Hidehiko Masuhara, "Bridging real-world contexts and units of behavioral variations by composite layers", Proceedings of international workshop on Context-Oriented Programming (COP'12), pp.1-6, June 11, 2012, Beijing, China.

⑪ Fan Yang, Hidehiko Masuhara, Tomoyuki Aotani, Flemming Nielson, and Hanne Riis Nielson, "Combining static analysis and runtime checking in security aspects for distributed tuple spaces", Proceedings of international conference on Coordination models and languages (COORDINATION'11), pp.202-218, June 7, 2011, Reykjavik, Iceland.

⑫ Tomoyuki Aotani, Tetsuo Kamina, and Hidehiko Masuhara,

“Featherweight EventCJ: a core calculus for a context-oriented language with event-based per-instance layer transition”, Proceedings of international workshop on Context-Oriented Programming (COP’11), pp.1-7, July 25, 2011, Luncaster, UK.

〔図書〕（計 0 件）

〔産業財産権〕

○出願状況（計 0 件）

名称：

発明者：

権利者：

種類：

番号：

出願年月日：

国内外の別：

○取得状況（計 0 件）

名称：

発明者：

権利者：

種類：

番号：

出願年月日：

取得年月日：

国内外の別：

〔その他〕

ホームページ等

## 6. 研究組織

### (1) 研究代表者

青谷 知幸（ Tomoyuki Aotani ）

東京工業大学大学院情報理工学研究科数  
理・計算科学専攻 助教

研究者番号： 20582919

(2) 研究分担者

( )

研究者番号：

(3) 連携研究者

( )

研究者番号：