

学位論文題目 Practical User-Defined Analysis-Based Pointcuts in an Aspect-Oriented Programming Language
(邦訳: アスペクト指向プログラミング言語における実用的な利用者定義の解析に基づくポイントカット)

取得年月 2009年3月 **学位種別** 博士(学術) **大学** 東京大学

氏名
推薦研究会
推薦文

青谷 知幸 (北陸先端科学技術大学院大学情報科学研究科 助教)
プログラミング

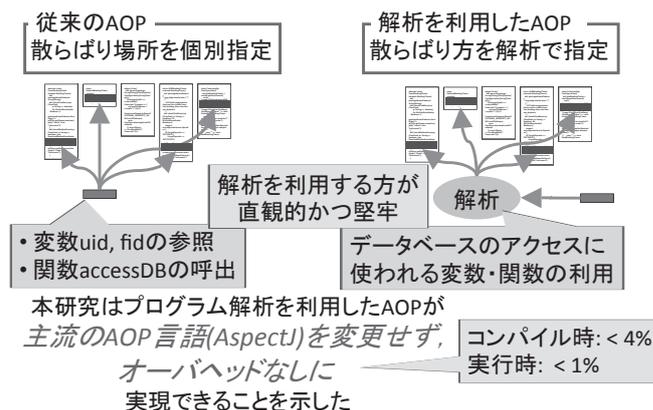
論文は産業界でも利用が広がっているアスペクト指向言語 AspectJ のコンパイラを拡張し、言語の互換性を保ちつつ、実行時速度をほとんど低下させずに強力なポイントカット機構を実現したものであり、実用性・将来性ともに高いものである。

アスペクト指向プログラミングを用いると、ロギングやトランザクション管理、例外処理のような、オブジェクト指向プログラミングなどの既存のプログラミング技術ではうまくモジュールとして分離できなかった処理をモジュール化できる。このとき登場する新しいモジュールでは処理の内容に加えて、いつ・どこでその処理を実行するかを指定する。この指定の手段をポイントカットと言う。AspectJをはじめとする主流のアスペクト指向プログラミング言語において、ポイントカットはクラスやメソッドまたはフィールドの名前を列挙することで記述される。

名前の列挙によってポイントカットを記述する方法には、プログラムが変更によってポイントカットが壊れてしまうという問題がある。クラスやメソッドまたはフィールドの名前が変更されればその通りにポイントカットを変更しなければならないし、新しいクラスやメソッドまたはフィールドが追加されたらポイントカットにそれらを付け加えなければならない。

この問題はプログラム解析の結果をポイントカットとして用いることで回避できる。このようなポイントカットを、プログラム解析を利用したポイントカットと呼ぶ。例としてデータベースの内容の更新を行うメソッドの実行をトランザクション管理することを考える。データベースの内容の参照と更新を行う低レベルAPIを使っているメソッドの集合をかき集めてくるプログラム解析があったとき、そのプログラム解析を指定することで解析結果の集合に含まれるメソッドすべてを指定したことになるポイントカットがあれば、データベースの内容の更新を行うメソッドの名前が変わったり新しく増えても、プログラム解析の結果が自動的にそのプログラムの変更を反映するため、ポイントカット自体を変更する必要がない。

本研究ではプログラム解析を利用したポイントカットを実用化するためのコンパイルおよびコンパイル時検査の技術を提案した。これまでもさまざまな研究によって実現手法が提案されてきたが、(1) 主流のアスペクト指向プログラミング言語と統合できない、(2) 効率の良いコンパイル手法がない、(3) プログラム解析の実装が困難、(4) ポイントカットの誤りを発見するのが困難という実用上の問題があった。本研究は (1') 既存の



アスペクト指向プログラミング言語 AspectJ を変更することなくプログラム解析を利用したポイントカットを実現するための記述方法の発見、(2') 部分評価を応用した効率の良いコンパイルの枠組みおよび (3') プログラム解析のための豊富なライブラリやツールを利用する枠組みの設計、(4') 想定されないポイントカットの組合せを検出する型システムの提案によってこれらの問題を解決した。コンパイルの枠組みを AspectJ の処理系の 1 つである abc コンパイラを拡張して実装し、Web 上でソースコードとともに公開している (<http://www.graco.c.u-tokyo.ac.jp/ppp/projects/scope>)。

(平成 22 年 4 月 30 日受付)