

平成16年度 学士論文

プログラム分散化のための  
アスペクト指向言語

東京工業大学 理学部 情報科学科

学籍番号 XX-XXXX-X

西澤 舞我

指導教員

千葉 滋 助教授

平成17年2月7日

## 概要

ここに Abstract を書く (A4 2 枚まで)

# 謝辞

本稿は以下の方々なくして、存在しえなかったでしょう。Addistant の開発、Addistant 2 の提案および本稿の編集になにかと心を砕いていただいた千葉滋講師、東京大学の光来健一氏、筑波大学の立堀道昭氏、横田大輔氏そして研究室のみなさん。心より感謝しています。

(具体的に何をしてもらったか書く)

# 目 次

第 1 章	はじめに	1
第 2 章	関連研究	3
2.1	Javassist . . . . .	3
2.2	Addistant . . . . .	3
第 3 章	Addistant 2 - 記述力強化版	4
第 4 章	実装	5
第 5 章	実験	6
付 録 A	プログラム例	8

## 図 目 次

4.1 Javassist の処理の流れ . . . . .	5
--------------------------------	---

# 表 目 次

5.1 Addistant 2 の実行時間 (秒) . . . . .	6
-------------------------------------	---

# 第1章 はじめに

今日、分散ソフトウェア、つまり複数の計算機上で動作するソフトウェアの必要性が高まる一方、その開発にかかるコストが問題になっている。これは、分散プログラムを作成する場合にネットワークなどの分散環境特有の問題に対処しなければならないためである。それらの処理の記述を含んだ分散プログラムは煩雑になり、非分散プログラムの作成に比べて、分散プログラムの作成や維持にかかる人的コストは飛躍的に大きくなりがちである。

分散プログラミングが煩雑であることの要因として、プログラムの分散に無関係な記述の中に分散に関わる処理が拡散して入り交じっている (crosscutting concerns) ことがあげられる。このようなプログラムは可読性が低く、変更も大変である。分散に無関係な記述が分かりにくくなる上、分散に関わる処理を変更するためにはプログラムのあちこちを修正しなければならない。

我々は分散プログラミングをアスペクト指向言語 (AOP) により支援する Addistant を開発してきた。ここで、アスペクト指向とは、オブジェクト指向との相補性を意識した概念である。オブジェクト指向とは、機能性という点に着目して全体をオブジェクトと呼ばれるモジュールに分割していく概念である。しかし、個々のモジュール内には、同様の処理が股がる可能性がある。この個々のモジュールに股がった同様な処理を、オブジェクトとは別の側面から考慮し、それをモジュール化する概念を**アスペクト指向**といい、そのモジュールを**アスペクト**という。

Addistant は次の特徴をもつ。

- Addistant の利用者に、各クラスごとそのオブジェクトを分散環境中のどこに配置するかを指定させる。Addistant では、分散に関わる処理がプログラム全体に拡散して入り交じることを避けるため、利用者は、分散に無関係な記述である非分散プログラムとは別に、分散に関わる記述をまとめて記述する。このまとめて別に書かれた分散に関わる記述を**分散アスペクト**と呼ぶ。
- 対象プログラムのバイトコードを変換して、分散アスペクトによって指定されたクラスが、遠隔ホストで動作している Java 仮想機械 (JVM) 上で実行されるようにする。Addistant は変換のために、対

象プログラムのソースコードを必要としない。変換されたバイトコードは正規の Java バイトコードであり、実行のために特別な JVM を必要としない。バイトコード変換には Javassist を用いた。

- 変換されたバイトコードを Addistant の実行系により遠隔 JVM に自動的に配布される。

Addistant では、遠隔オブジェクト参照を、従来の分散プログラミング・ツールで使われてきたアイディアを組み合わせで実現した。この実装は、**プロキシ・マスタ方式**に基づいたもので、Addistant がバイトコード変換により自動的に行う。

しかしながら、Addistant では、開発が進むにつれ、その利点とともに問題点も明らかになってきた。その問題点とは、分散アスペクトの記述力が十分ではなかったため、利用者が望む機能分散をうまく実現できない場合があることである。

そこで我々は、Addistant の分散アスペクトの記述力を大幅に高めた Addistant 2 を開発中である。本稿では Addistant 2 の典型的な利用方法の例をあげ、分散アスペクトの新しい記述方法を説明する。

本稿の残りは、次のような構成からなっている。(以下では、Addistant を Addistant 1 と呼ぶ。) 第 2 章は 分散プログラミング用アスペクト言語の必要性と Addistant 1 の性能を述べる。第 3 章では、Addistant 2 の分散アスペクト記述を、第 4 章では、Addistant 2 のこれまでの実装、第 5 章では、Addistant 2 を利用したソフトウェアの機能分散例、第 6 章では、関連研究を、そして第 7 章でまとめを述べる。



## 第2章 関連研究

### 2.1 Javassist

Javassist [1, 3] は...

Javassist は [2] で配布されている。

### 2.2 Addistant

## 第3章 Addistant 2 - 記述力強化版

## 第4章 実装

図 4.1 は...

図 4.1: Javassist の処理の流れ

## 第5章 実験

表 5.1 は...

表 5.1: Addistant 2 の実行時間 (秒)

	時間
X Window System	15.0
Addistant 1	3.0
Addistant 2	2.0

## 参考文献

- [1] Chiba, S.: Load-time Structural Reflection in Java, *Proceedings of the European Conference on Object-Oriented Programming*, pp. 313–336 (2000).
- [2] 千葉滋: Javassist Home Page, <http://www.csg.is.titech.ac.jp/~chiba/javassist/index.html>.
- [3] 千葉滋, 立堀道昭: Java バイトコード変換による構造リフレクションの実現, 情報処理学会 論文誌, Vol. 42, No. 11, pp. 2752–2760 (2001).

## 付 録 A    プログラム例