
発表概要

分散 Java プログラミングのためのアスペクト 指向言語

西澤 無我[†] 千葉 滋[†]

本発表で、我々は分散プログラム内に存在する横断的な関心事を、他のモジュールから分離して記述することが可能なアスペクト 指向言語を提案する。横断的な関心事は、複数のクラスまたはモジュールにまたがってしまう処理のことを指し、既存のオブジェクト 指向プログラミングではうまく扱えない。アスペクト 指向プログラミングは、このような横断的な関心事の分離をはかる技術である。分散プログラム内にも横断的な関心事は存在し、この関心事が分散プログラミングを困難なものにしている理由の 1 つだといえる。ネットワーク対戦ゲームでは、あるプレイヤーがウィンドウ上のボタンをクリックすると全員のウィンドウが更新される、などの例がよくある。これは、オブジェクト 指向プログラミングの Observer パターンを用いて設計・実装されるが、subject と observer とが異なるホスト上にあるケースは少なくない。このような Observer パターンの実装は異なるホスト間を横断する関心事であるといえる。本言語は、このような分散プログラム内に存在する横断的な関心事を、1 つのアスペクトと呼ばれるモジュールにまとめて記述することを可能にする。汎用的なアスペクト 指向言語である AspectJ などは、ネットワーク越しの横断的な関心事を直接的にサポートしていない。AspectJ は、単一のホスト上の複数のクラスにまたがる関心事を 1 つのモジュールにまとめて記述することが可能な言語である。一方、本言語は、AspectJ で解決可能なクラス間にまたがる横断的な関心事はもちろん、異なるホスト間にまたがる横断的な関心事も 1 つにまとめて記述できる言語である。

An Aspect-Oriented Programming Language for Distributed Programming in Java

MUGA NISHIZAWA[†] and SHIGERU CHIBA[†]

We present an Aspect-Oriented Programming (AOP) language, which enables developers to separate crosscutting concerns in distributed Java programs. Crosscutting concerns are ones cutting across several classes or modules and Object-Oriented Programming cannot support it. AOP is the technology for separating such concerns. Separating crosscutting concerns are found in distributed computing domain and they make distributed programming difficult. For example, there is a typical network games may have to update all players' windows when someone clickes somewhere in the window. This would be implemented according to Observer pattern known in OOP, but the subject and observer would be distributed into several hosts. This process is a typical concern cutting across multiple hosts. Our language makes it possible to separate these crosscutting concerns in distributed programs into a module called an aspect. AspectJ, which is a general-purpose AOP language, does not support these crosscutting concerns across network. It is a language that can separate crosscutting concerns across two or more classes on the same host into one module. On the other hand, our language can separate crosscutting concerns across different hosts, as well as crosscutting concerns supported by AspectJ.

(平成 15 年 8 月 4 日発表)

[†] 東京工業大学数理・計算科学専攻

Department of Mathematical and Computing Sciences,
Tokyo Institute of Technology