

Title	抽象実行に基づく Java プログラムの発展的プロトタイプ タイピングに関する研究
Author(s)	尾崎, 弘幸
Citation	
Issue Date	2004-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/949
Rights	
Description	Supervisor:片山 卓也, 情報科学研究科, 博士

要 旨

プロトタイピングは、ソフトウェア開発において、実装工程といった遅い段階からの無駄なバックトラックコストを減らす。しかし、全体を見通した開発と途中段階での全体としての実行を両立することは難しい。したがって、プロトタイプ構築の遅い段階からのバックトラックコストが発生する。

そこで、本論文では、これらを両立できる発展的プロトタイピング技法を提案する。そのアイデアは、抽象解釈に基づいた段階的詳細化によるプロトタイプの構築である。抽象化したオブジェクトから構築を始め、オブジェクトを詳細化することで構築を進める。詳細化によるプロトタイピングが全体を見通す開発を実現する。さらに、オブジェクトを実行時に抽象化することでプロトタイプ全体としての実行を実現する。実行時にオブジェクトを抽象化し、プロトタイプを実行するメカニズムを抽象実行と呼ぶ。

抽象実行の実現には、オブジェクトの詳細化を厳密に決めることが重要である。本論文では、その詳細化の形式的な定義を与える。意味論に踏み込んで詳細化の整合性を保つように定義する。これにより、抽象実行時の振舞いの不整合を防ぐ。

次に、形式的なオブジェクトの詳細化に基づいて、抽象実行支援ツールの実装を与える。ツール支援なく抽象実行を実現することは非常にコストが高い。なぜなら、抽象実行では、データ、メソッド、クラスといった様々な言語要素の変換（抽象化）を必要とするからである。本論文では、仲介オブジェクトを導入し、機械的な抽象実行を実現している。リフレクション技術と XML 技術を用いて仲介オブジェクトを機械的に生成することができるので、効率的にプロトタイプを構築することが期待できる。

最後に、本技法の詳細化の考え方を拡張することで fragile base class problem の起きない安全なクラス継承を実現できることを示す。クラス継承は、オブジェクト指向技術の核となるメカニズムの一つである。しかし、fragile base class problem として知られる致命的な問題がある。本論文では、機能詳細化に機能追加を導入し、クラス継承を実現する。機能詳細化と機能追加によるクラス継承の単調性を証明することで、fragile base class problem が起きないことを示す。