

Title	ソースコードの依存解析と視覚化手法に関する研究
Author(s)	横山, 浩一
Citation	
Issue Date	2001-03
Type	Thesis or Dissertation
Text version	author
URL	<a href="http://hdl.handle.net/10119/1437">http://hdl.handle.net/10119/1437</a>
Rights	
Description	Supervisor:落水 浩一郎, 情報科学研究科, 修士

# ソースコードの依存解析と視覚化手法に関する研究

横山 浩一

北陸先端科学技術大学院大学 情報科学研究科

2001年2月15日

キーワード： 視覚化, 依存解析, プログラム保守.

## 1 本研究の背景と目的

オブジェクト指向技術の進歩とともに、ソフトウェアの設計や実装に対して用いられる機会が増えてきた。そういったソフトウェアは、パッケージ・クラス・メソッドといった形で細分化され、モジュール化されており、直接ソースコードを読んでみても、ソフトウェアの構造や実行時の振舞を理解することは難しい。

そのような中、変更要求や、新しいプラットフォームへの移植に対応する必要が出てきた時、ソースコードを読まなくてはいけなくなる場合がある。しかも、古いソフトウェアで仕様書が残されていなかったり、仕様書の記述が古く、現状を正しく記述していないかも知れない。

そのような時、ソースコードからのソフトウェア理解を、コンピュータによって補助することは有効であると考えられる。実際に、その試みは行なわれており、RAD環境では大抵、クラスビューや構文エディタとしての機能を持っている。

これらのツールは図形や色彩を用いてソースコードの視覚化を行ない、コード理解を簡単にしてくれる。

しかし、これらのツールは単一の抽象度による情報呈示機能しか提供していない。

そのため、ユーザは異なる抽象度の情報を得ようとする、新たなツールや、新たな表示へ視点を移さなければいけない。

たとえば、大局的なクラス構造しか提供しないクラスビューから、局所的な構文情報をしめす構文エディタへ切り替えるような場合である。

そこで、抽象度の異なる2つの視野を緩やかに変化させて見る事により、抽象度によらない透過的な情報呈示が必要となる。

こういった課題に対して、多くのソフトウェアに一貫したモデルを定義することは、とても困難である。

そこで、オブジェクト指向言語のプログラムに限定して、依存関係や構造をさまざまな抽象度で呈示することが可能な、ソースコードブラウザとして‘SZ\_Browser’を提案する。

これは Java プログラミング言語によって書かれたソースコードをサポートし、異なる抽象度間をスムーズに変化するユーザインタフェースとを持つ。

これによって、ユーザはある抽象度における視点と他の抽象度における視点の間の関連を容易に把握することができるようになる。

## 2 依存解析

本論文で定義する依存関係とは、構成要素間に継承関係や参照関係が存在することとし、それらはソースコードを解析することによって得られるものである。

これは、仕様書無しにソースコードを読んで構造を理解する際に、これらの関係を捕捉の情報として利用することができる。

## 3 呈示方法

オブジェクト指向プログラムが持つクラスとクラス、操作と操作、クラスと操作の間にはそれぞれ関係があり、それはどの抽象度で関係を扱うかによって異なってくる。

それを一度に表示する事は非常に複雑となり、何の情報も得られない。

そこで、何らかの基準に基づいて関係の一部を抜き出して呈示する呈示方法が必要である。

その基準の一つとして、構成要素間の距離の利用を考えている。距離とは、要素間のつながりの強さ、関連の種類によって定義する。

距離を用いることにより、さまざまな抽象度において、興味を持った物だけを見せることができ、他の関係を隠してしまうことができる。

我々はこの概念を semantic zooming と呼ぶ。

semantic zooming を行なうことによって、重要な問題にだけ注目することができる。

たとえば、いずれかの関係について細部を知りたい場合、ズームインによってその関連を表示に加えることができる。

また、もっと高いレベルの意味的關係が見たければ、ズームアウトすることによって關係の数を減らすことができる。

また、semantic zooming のもう一つの利用方法は、異なる抽象度の2つの要素に対して、一方のズームレベルにあわせて、他方のレベルも拡大縮小を行なうものである。

たとえば、インスタンスの生成構文と生成されるクラスの間の關係を、クラスのレベルではクラス間に、メソッドのレベルではメソッドとクラスとの間に、メソッドボディのレベルでは生成構文とクラスとの間に見せるように、ズームングを行なう。

これによってそれぞれの構成要素が持つ、クラス、メソッド、コードにおける、階層的な関連を得ることができる。

## 4 情報視覚化

コンピュータグラフィクスによる視覚表示はこれまで、主に科学技術の計算結果を視覚化することにしか用いられてこなかった。しかし、一般に普及した大画面、高解像度、高性能、低価格なコンピュータにより、情報視覚化、ソフトウェア開発に拡張しようとしている。

card はさまざまな分野に適用できる抽象データとの対話手段・視覚表現として情報視覚化の参照モデルを提案した。

このアイデアは、さまざまな情報を抽象化し、内在する構造・傾向を明らかにすることで、映像のような視覚情報の直接的な表示ではなく、抽象的な情報を視覚化するためのモデルである。

次のステップは内部構造の粒度をもとに、ズームイン・ズームアウトという操作を行なう新しい技術の開発である。

われわれは先に示したコンセプトとテクニックをオブジェクト指向プログラムに応用した。

我々がこの論文で構築したモデルと手法はソースコードの持つ内部構造に関してセマンティックズームングが有効であることを示している。

そして、この技術を用いて今回作成したツールの方が、実際にソフトウェア構造の理解の補助となることから、プログラムの全体的な構造と内部の詳細構造の間関係を理解するためには、従来からある伝統的なクラスビューより優れていると考えている。

## 5 プロトタイプ実装

本研究では、対象とするデータとして Java プログラムコードを採用し、ソースコード理解に有効な依存関係を提案する。

また、Java プログラムの構成要素間の依存関係を、分析することができるアルゴリズムを開発した。そして、手動でのソースコード分析と、理解プログラムのパフォーマンスにおける比較について論じた。

そして、ツールを用いた依存関係と情報の段階的呈示を基礎とした、情報視覚化の枠組として、情報視覚化とその解析に必要な 3 つの相互関係と 4 つの状態を導入することによる、我々のモデルとツールの有効性と有用性を示す。

最後に、基礎となる研究結果の有効性を示すためにプロトタイプとして、一部機能を実装した。