

平成 19 年度 情報工学コース卒業研究報告要旨

阿草・結縁 研究室	氏 名	上 原 伸 介
卒業研究題目	ソフトウェア理解支援のための ソフトウェアナビゲーションマップ	

本研究では、プログラム依存グラフ (PDG: Program Dependence Graph) から取得できる情報に基づきソースコードを多粒度で表現し、対話的なナビゲーションによるソフトウェア理解支援を可能とする「ソフトウェアナビゲーションマップ」を生成する手法を提案する。

ソースコードレビューやデバッグにおいてソフトウェアの動作を理解するには、関数の呼び出し関係を辿りながらその処理内容を理解する方法が一般的である。呼び出し元となる関数の機能を把握するためには、呼び出し先の関数で実現される機能を理解しなければならない。一方では次にどの関数が呼ばれるかに注意を払いつつ、もう一方では現在の関数を読み解く必要があるため、呼び出し関係が複雑である場合には、単純に呼び出し関係を辿ることによる理解は困難である。

呼び出し関係を辿ることによる理解を支援できるソフトウェア理解支援手法としては、スライシングやコールグラフ図示などがある。スライシングは、式や変数など細かい粒度の呼び出し関係を表すため、処理の詳細を把握するには適しているが、大局的な流れを把握するには適さない。またコールグラフ図示は、関数間の呼び出し順序など粗い粒度の呼び出し関係を表すため、大局的な流れを把握するには適しているが、処理の詳細を把握するには適さない。処理の詳細と大局的な流れの双方に対して、その理解を支援する手法が必要である。

本研究では、利用者の着目点に応じて呼び出し関係の粒度を変更することにより、ソフトウェアの大局的な流れと詳細を同時に閲覧可能な可視化手法として、ソフトウェアナビゲーションマップを提案する。ソフトウェアナビゲーションマップでは、利用者が着目している、あるいは次に着目する必要がある箇所を詳細に表示し、利用者が過去に着目した箇所はその概要を表示する。提案手法により、利用者は自身のソフトウェア理解作業において、過去の読解作業の文脈を失うことなく、作業を継続することができる。

また、提案手法によりソースコードを解析して、ソフトウェアナビゲーションマップを生成、操作できるツールを開発した(図 1)。さらに、実際に本ツールを利用してソースコードの可視化を行い、提案手法の有用性について議論した。

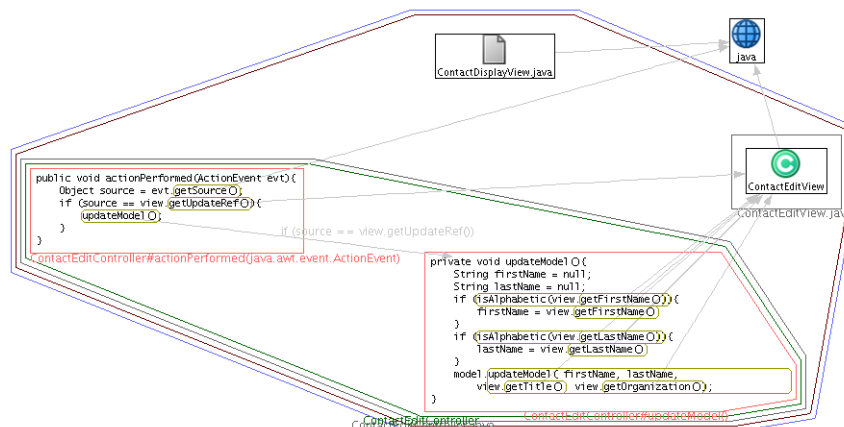


図 1:ソフトウェアナビゲーションマップの例