

Title	ソースプログラムの差分解析と波及解析に基づく開発状況の把握法に関する研究
Author(s)	林崎, 浩典
Citation	
Issue Date	2000-03
Type	Thesis or Dissertation
Text version	author
URL	<a href="http://hdl.handle.net/10119/1328">http://hdl.handle.net/10119/1328</a>
Rights	
Description	Supervisor:落水 浩一郎, 情報科学研究科, 修士

# ソースプログラムの差分解析と波及解析に基づく 開発状況の把握法に関する研究

林崎 浩典

北陸先端科学技術大学院大学 情報科学研究科

2000年2月15日

キーワード: java, プログラム解析, 状況把握.

## 1 本研究の背景と目的

ソフトウェア開発保守現場では、迅速な開発のために並行に作業を進めることが求められており、競合をいち早く察知することは重要である。開発者自身に関連する競合が発生している状況を把握できれば、効率の良い作業が期待できる。しかしながら、開発者が、過去のバージョンからの変更の経緯を知るためには、リポジトリから成果物の差分や作業ログ情報を得て、手作業で解析を行い、必要な情報を見つけ出すことをしなければならない。このような作業は、コストが高い上に手作業であるために、誤りも生じやすい。プログラム理解は多大な労力を費やす作業であり、理解を支援するための多くの研究がなされている。それとともに、プログラムの変更部分や作業量を把握し、開発者が理解の必要な部分の見当をつけるための手法の考案も重要であり、変更作業の視覚化や変更履歴の追跡に関する研究の発展が期待される。

本研究では、各開発者に追加的な負担を強いることなく、個人の好みのスタイルによる作業の結果を抽出するために、ソースプログラムの解析に着目し、開発状況を把握するための手法を考案する。具体的には、異なる（例えば、CVSの）ブランチ間で、将来的にプログラムのマージや動作に影響を及ぼすような波及を特定し、その波及の全体として競合を察知できるような、開発状況の把握を支援する手法の開発を行う。

## 2 競合の検知

本研究では、開発者がバージョン管理システムを利用することにより、同じバージョンに対するブランチを作成し、作業を進める並行開発を対象とする。プログラム解析の対象は、オブジェクト指向言語 Java とする。

Java プログラムの解析器は、データフロー等のプログラムの意味的な関係の表現をもつ必要がある。オブジェクト指向システム依存グラフ（以下、OSDG）を利用し、競合の検知を行う。以下に競合検知の手順を示す。

- システム依存グラフの作成

Java 言語で記述されたソースプログラムを解析し、OSDG に変換する。OSDG を作成する際に、解析の精度を向上させるため、実行時に型が決定されるようなメソッド呼び出しは、近似する必要がある。また、後の解析ステップで、OSDG 上に表現されないメソッドの上書きなどの情報も解析し、利用する。

- 差分抽出

同一ブランチ上における OSDG のバージョン間の差分を抽出する。抽出の結果として、変更前・後の OSDG の節間に対応関係を作成することにより、挿入や削除といった変更作業の内容を記録しておくことが可能となる。また、OSDG を利用していることから、節による差分の他に、グラフの文中にある辺（例えば、メソッド呼び出し関連）による差分の抽出も可能である。それらの差分を利用して、特定のブランチの状況を把握することができる。

- 波及解析

開発者が開発しているプログラム中に、他の開発者からの影響を受けるような依存関係を持つ部分を探し出すことにより、影響の波及を解析する。変更は単一ではなく複数箇所に加えられており、変更箇所の及ぼす影響は、広範囲に渡ると考えられる。ただし、依存関係は、メソッド呼び出し、メンバ変数の参照、継承関連とする。

- 競合の検知

差分抽出と波及解析の結果を利用して、プログラム中に依存関係があり、変更の影響を受けるときに、競合を検知する。具体的には、メソッド・メンバ変数へのアクセスに影響する変更、継承関連の変更、戻り値に影響を与える変更がある場合を競合とする。

## 3 プロトタイプシステムの設計と実装

競合検知の部分的な解析法（クラス宣言の変更）を設計の対象範囲とし、プロトタイプシステムを開発した。プロトタイプシステムでは、ブランチ間での変化しつつある状況を追跡可能にするユーザ・インタフェースが実現されている。このインタフェースではマー

キング機能、詳細情報機能が備わっており、利用者は、数多くある競合の種類を容易に見分けることができ、直感的に状況を把握することができる。

マーキング機能は、ソースプログラムの影響を受ける部分にマーキングすることで、影響の存在することを通知する。詳細情報機能は、マーキングされた部分に対応する競合箇所、バージョン名、競合の発生した理由が表示する。さらに、変更の影響を与える側と受ける側を対応させて表示することで、比較しながら把握することを可能にしている。実装は、CASE ツールプラットフォームの Japid を利用しており、最終的な出力は、HTML として得られるため WWW ブラウザ上で閲覧可能である。

## 4 まとめと今後の課題

本研究では、開発状況の把握のために OSDG 間での差分抽出、波及解析、競合の検知の方法を定義し、同一ブランチ上の OSDG の差分による把握法と異なるブランチの OSDG 間の競合検知による把握法を提案した。また、ソースプログラムにおける競合検知の基礎となる手法を提案し、その方法を応用する一例として、未来版管理システムの汚染マーク機構における細粒度化について考察した。今後の課題は、提案した解析について全体の実装を行い、評価実験の環境を整え、複雑な依存関係の理解を支援するユーザ・インタフェースを改善することである。