

ソフトウェア構築ログのリポジトリとその応用

永井郁哉 指導教員：山本晋一郎

1 はじめに

ソフトウェアは複数のソースファイルと様々なライブラリから多くのオプションを付けてコンパイルすることで構築される。この構築過程は `make` コマンドを代表とする自動構築ツールに管理される。開発者がソフトウェアを未対応の環境に移植する場合、自動構築ツールを利用する。このとき、ソフトウェアの構築に失敗、あるいはソフトウェアの構築はできたが期待する動作をしないことがあり、この原因はソフトウェアの構築過程に存在する。原因を特定するために構築時に使用したオプションの情報や構築過程の情報等が必要になるが、これらの情報は構築ルール記述ファイルや構築ログから取得することができる。

構築ルール記述ファイルの種類は自動構築ツールの数だけ存在し、このファイルを1つ取っても文法の自由度が高いため構築ルール記述ファイルから情報を取得することは困難である。近年のソフトウェアの大規模化に伴い構築ログも大規模化し、構築ログからも情報を取得することは困難である。

本論文ではこの問題に対し、構築ログのリポジトリと構築ログに出現するファイルのリポジトリを生成することで解決を図る。構築ログを適切に解析しリポジトリに格納することができればリポジトリから原因を特定するための情報を取得することは容易である。構築ログに着目した理由は自動構築ツールに何を使用しても統一的に調査することができるからである。構築ログのリポジトリからは構築過程に使用したコマンドやオプションの情報を取得することが容易となり、構築ログに出現するファイルのリポジトリからはどのファイルからどのようなファイルが生成されるか、あるいはコマンドの引数に生成されたファイルを取る場合、そのファイルがどこで生成されたかという情報を容易に取得することが可能となる。

2 リポジトリの生成

我々はソフトウェアの構築過程を表現するモデル [1] を提案し、ソフトウェア構築ログ解析器を作成した。本論文では解析器を拡張し、解析結果とモデルからソフトウェア構築ログのリポジトリと構築ログに出現するファイルのリポジトリを生成する。具体的な解析器の拡張内容は引数を取るオプションを知識として手で与えていたがこれを自動化し、構築ログに現れるディレクトリの情報を扱えるよう拡張した。図1は構築過程を表現するモデルである。ソフトウェアの構築過程はコマンド、オプション、引数の3項組から成り立つと考える。コンパイルされるソースファイルとそのコマンド実行による生成物はファイルである。ファイルを引数に取るコマンドを実行することで新たなファイルが生成される関係を `generate` とする。

生成されるソフトウェア構築ログのリポジトリを図2に示す。ソフトウェアの構築ログはどのディレクトリでコマンドを起動するかという実行ディレクトリから構成される。実行ディレクトリはそのディレクトリにおいてどのようなコマンドを実行するかというコマンドラインから構成される。

生成される構築ログに出現するファイルのリポジトリを図3に示す。ファイルリストはソフトウェアの構築ログにファイル

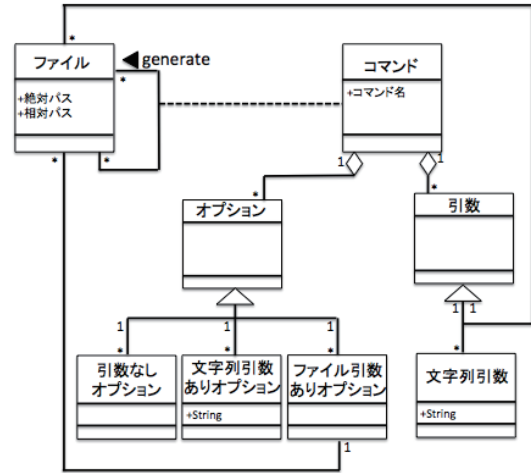


図1 ソフトウェア構築過程のモデル

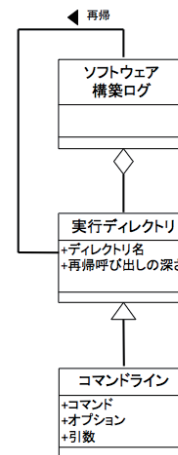


図2 ソフトウェア構築ログのリポジトリ

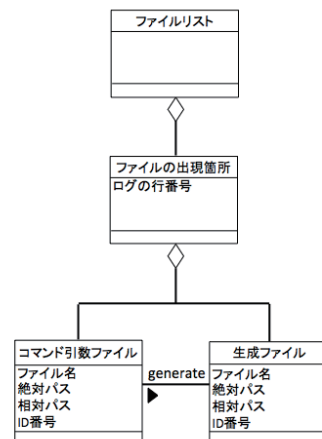


図3 構築ログに出現するファイルのリポジトリ

表 1 ソフトウェアの詳細

	ファイル数	ログ総行数	コンパイル時マクロ	リポジトリの生成
ed-1.9	8	9	○	○
bc-1.06	20	24	○	○
combine-0.4.0	52	819	○	○
gnugo-3.8	67	329	○	×

が出現する箇所構成される。ファイルの出現箇所はコマンド引数ファイルと生成ファイルの 2 要素で構成される。コマンド引数ファイルから生成ファイルは構築されるためこの関係を generate とした。

3 リポジトリの応用

本論文のリポジトリは様々なことに利用できる。1 つ目にどのファイルからどのようなファイルが生成されるかというファイルの依存関係を可視化することに利用できる。可視化を行うことでソフトウェアの構築過程の理解支援をすることができる。

2 つ目に Sapid の SDB を構築ログから生成することができる。構築ログから SDB を生成することが可能となることで Sapid の適用範囲を拡大することが期待できる。

3 つ目に SPIE を拡張することができる。SPIE を拡張することでさらに解析対象のソフトウェアの情報をユーザに提供することが可能となる。

3.1 SPIE の拡張

ソフトウェアブラウザ SPIE[2] とは Sapid によって SDB に格納されたソースプログラム、関数、大域変数、局所変数やマクロといった情報のクロスリファレンス表を自動生成するために使用される。Sapid とは C 言語に対する CASE ツールを構築するためのプラットフォームであり、C ソースプログラムを解析し、その結果を独自のリポジトリ (SDB と呼ぶ) に格納する。

SPIE の問題点はコンパイル時に与えられるマクロの情報が表示されないことである。コンパイル時のマクロはソフトウェアのバージョンや構築する環境によってコンパイルするコードを変更する目的で使用し、コンパイル結果に影響を与える。したがってソフトウェアの構築において非常に重要な役割を果たす。

この問題に対し、ソフトウェア構築ログのリポジトリを利用する。リポジトリからコンパイル時に渡されるマクロとそのマクロの影響があるファイルの情報を取得し、SPIE にコンパイル時のマクロの情報をクロスリファレンス表に追加するアプリケーションを実装することで解決を図る。これにより開発者はソースプログラムの情報に加え、コンパイル時に与えられるマクロの情報も手に入れることが可能となる。

4 評価

評価は ed-1.9, bc-1.06, combine-0.4.0 と gnugo-3.8 の 4 つのソフトウェアを使用する。4 つのソフトウェアの詳細とこれらに対しリポジトリが生成可能か調査した結果を表 1 に示す。ed-1.9, bc-1.06 と combine-0.4.0 はリポジトリの生成に成功した。また、SPIE にコンパイル時のマクロの情報を組み込むことが可能か調査を行ったところ同様に成功した。

gnugo-3.8 はリポジトリの生成に失敗した。gnugo-3.8 の構築ログの一行をわかりやすく改行したソースコード 1 に示す。

ソースコード 1 gnugo-3.8 の構築ログ

```
1 if gcc -MF .deps/getopt.Tpo -c -o getopt.o getopt.c;
2 then mv -f .deps/getopt.Tpo .deps/getopt.Po;
3 else rm -f .deps/getopt.Tpo; exit 1; fi
```

このような if 文の場合、どちらのパスを通るかは自動構築ツールが判断すべきであり、ソフトウェア構築ログ解析器が判断すべきではないと考える。したがって、この構築ログは解析の対象外でありリポジトリを生成することができなかった。

5 終わりに

本論文ではソフトウェア構築ログのリポジトリと構築ログに出現するファイルのリポジトリを生成し、リポジトリを利用したアプリケーションの作成を行った。リポジトリを生成することでソフトウェアの構築過程に関する情報を容易に取得することが可能となり、アプリケーションの作成によって SPIE を拡張した。SPIE の拡張によりソースプログラムの情報に加え、ソースプログラムに影響を与えるコンパイル時のマクロの情報を表示することが可能となった。

今後の課題を以下に挙げる。第一に構築ログ解析器を拡張することである。構築ログに出現するコマンドやオプションの解析ができない場合があるため、拡張する必要がある。第二にリポジトリを利用したアプリケーションの作成である。本論文で示したリポジトリの利用方法以外に環境の違いによる構築プロセスの比較やソフトウェアのバージョンの違いによる構築プロセスの比較に利用できることが考えられる。

参考文献

- [1] 永井郁哉, 大久保弘崇, 粕谷英人, 山本晋一郎. ソフトウェア構築ログ解析器を用いたソースプログラム解析. ソフトウェアエンジニアリングシンポジウム 2014 論文集, 第 2014 巻, pp. 196-197, aug 2014.
- [2] 大橋洋貴, 山本晋一郎, 阿草清滋. ハイパーテキストに基づいたソースプログラム・レビュー支援ツール. 電子情報通信学会技術研究報告. SS, ソフトウェアサイエンス, Vol. 98, No. 295, pp. 15-22, sep 1998.

発表実績

- 永井郁哉, 大久保弘崇, 粕谷英人, 山本晋一郎. 実行履歴に基づくソフトウェア構築モデルの提. ソフトウェアエンジニアリングシンポジウム 2013 論文集, 第 2013 巻, pp. 1-2, sep 2013.
- 永井郁哉, 大久保弘崇, 粕谷英人, 山本晋一郎. ソフトウェア構築ログ解析器を用いたソースプログラム解析. ソフトウェアエンジニアリングシンポジウム 2014 論文集, 第 2014 巻, pp. 196-197, aug 2014.