

実行履歴に基づくソフトウェア構築モデルの提案

永井 郁哉¹ 大久保 弘崇² 粕谷 英人² 山本 晋一郎²

概要: 異なる環境でのソフトウェア構築を行うために、ソフトウェア構築時の動作情報を収集し、利用することを考える。そこで、本論文ではソフトウェア構築のモデル化し、解析を行う。モデル化をするためにコマンドラインの構文規則を決定し、その構文規則に則った解析器を作成する。これによってビルドに使用されるコマンドやファイルの生成関係を読み解くことにかかる時間を短縮することができる。

Proposing the Software Build Model based on Execution History

IKUYA NAGAI¹ HIROTAKA OHKUBO² HIDETO KASUYA² SHINICHIRO YAMAMOTO²

Abstract: We collect the operating information of the software build and consider to utilize it to build the software in the different environment. In this paper, we modelize the software build and analyze it. We decide the syntax rule to modelize the commandline and generate the parser along the syntax rule. This causes that we can reduce time to understand command used by software build and the relation to create the files.

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コンパイル処理により構築される。一般にその工程は make コマンドを代表とする自動構築ツールにより管理される。また、ソフトウェアを構築する OS やライブラリなどの様々な環境への対応も、自動構成ツールにより管理される。その対応は自動構築ツールに与える設定で行うが、設定の作成、すなわちソフトウェアの移植性の確保は開発者の努力に依存する部分が多い。ソフトウェアの配布物が未対応の環境に移植を試みる場合、自動構築ツールの設定の修正という作業が必要である。これらの作業を行うためにはソフトウェアの構築工程の内容、すなわち自動構築ツールの動作や、どのファイルからどのようなファイルが生成されるのかといった依存関係に関する知識が必要となる。

我々はこの知識を獲得する作業に着目し、その計算機支援を行うツールの実現を考える。知識の獲得源、すなわちツールの入力としては、第一に自動構築ツールの設定ファイルそのもの、例えば make コマンドであれば Makefile、が重要である。しかし現在はさまざまな自動構築ツールが存在し、またそれらの機能は豊富であるため、支援ツールの実現は相当に困難である。

もう一つの情報源として、自動構築ツールの動作ログが利用できる。自動構築ツールが起動する具体的なコンパイルコマンドをオプションとともに記録すれば、設定ファイルのフォーマットや自動構築ツールの仕組みに依存せず、実際のソフトウェアの具体的な構築状況が取得できる。

自動構築ツールの設定ファイルは、対応するあらゆる環境に関する構築工程が定義されているが、構築ツールの動作ログは、構築を行った特定の環境に関する情報しか得られない。しかし逆に、異なる環境下での構築動作の差異を抽出する、構築に失敗した場合に期待される正常な構築動作と比較する、などの目的には、ログに基づく方法がより適していると考えられる。

本論文では、自動構築ツールの動作結果を表すモデルを

¹ 愛知県立大学大学院 情報科学研究科
Graduate School of Information Science and Technology,
Aichi Prefectural University
² 愛知県立大学 情報科学部
School of Information Science and Technology, Aichi Prefectural University

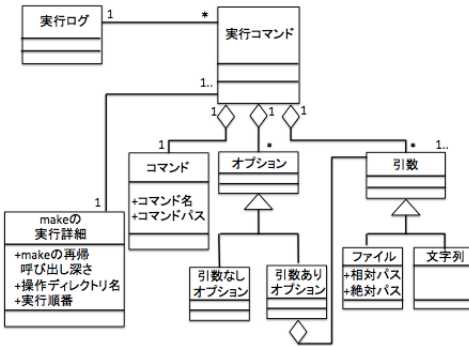


図 1 実行ログの UML クラス図

提案する. また, 具体的に make コマンドを対象とし, ログを解析してモデルを構築するツールを実現する. モデルを用いることで, 構築工程の比較や可視化など, 構築工程の理解に必要な知識を効果的に取得できるようになる.

2. ソフトウェア構築のモデル

make の実行ログを表すモデルの UML クラス図を図 1 に示す. make の実行ではさまざまなコマンドが起動される. これらのコマンドの実行は, 実行コマンド, オプション, 引数の 3 つから構成されると考える. この 3 項組を持つクラスが「実行コマンド」である. また, make はコンパイラだけでなく cp や rm などのファイルやディレクトリを操作するコマンドも実行する. さらに, make コマンドはサブディレクトリに関して再帰的に起動されることがある. この情報を「make の実行詳細」に分離した. 実行コマンドの任意個の集まりが, 一つの実行ログをなす.

3. ソフトウェア構築ログの解析器

ログを解析しモデルを生成する解析器は純粋関数型言語 Haskell を用いて実装した. Haskell には解析器を作成するための Parsec [1] というモジュールが提供されている.

コマンドは, どのようなオプションが認識されるか, またそれらは引数を取るか否かは個々に異なる. コマンドソースを Sapid [2] で解析し, 各コマンドに対して引数を取るオプションと引数を取らないオプションを自動で判定する機構を実装した. ただし, この機能はコンパイルに使用される主要なコマンドや getopt ライブラリを利用するコマンドに限定している.

3.1 ソフトウェア構築ログの構文規則

コマンドラインの構文規則を, 拡張 Backus-Naur 記法により図 2 のように定めた.

解析器の内部で用いる, モデルの表す抽象データ型の定義を図 3 に示す. *EnteringDir* と *LeavingDir* は, make コマンドを実行した際に出力されるディレクトリの情報である. これらはどこのディレクトリでコマンドを実行した

〈コマンドライン〉 := 〈コマンド〉 〈オプション〉* 〈引数〉+
 〈コマンド〉 := [A-Za-z0-9]+
 〈オプション〉 := “-” 〈オプション文字〉* 〈引数〉*
 〈引数〉 := 〈引数文字〉+

図 2 実行コマンドの構文規則

```
data Oneline = Line Cmd [Option] [Argument]
             | EnteringDir DirName Level
             | LeavingDir DirName Level
data Cmd     = Cmd String
data Option  = Option String [Argument]
data Argument = Argument String
type DirName = String
type Level   = Int
```

図 3 抽象データ型

かを意味する. *Level* は make を再帰的に呼び出した深さを示す.

4. おわりに

本論文では, ソフトウェアの自動構築ツールが管理するソフトウェア構築の工程に着目し, その内容を表すモデルを提案した. make コマンドを対象とし, 実行したログを解析し, モデルを生成するツールを実現した.

現在, 以下の評価方法を検討している. オープンソースのソフトウェアを用いて, 本手法が様々なソフトウェアに適用できるか確認する.

また, 関連研究と比較を行い, 提案したモデル化によってどの程度の効果が得られるかを評価する.

今後の課題を以下に挙げる. 第一に, ログの解析部をさらに強化する必要がある. 図 2 に示したコマンド行文法は多くのコマンドに適用できるが, tar コマンドのオプションのように, この構文では対応できないものも存在する. また, シェルの組み込みコマンドもこの構文では対応できない. これらの処理を追加する必要がある. 第二に, 本研究の目的を考えると, 構築に失敗したログに関して, より多くの情報を取得することが必要であるといえる. 実行ログの比較を行うような応用を考慮して, モデルを拡張する. 第三に, 提案したモデルの応用である. 例えば, モデルの表す依存関係を可視化することでソフトウェア構築の理解を支援する.

謝辞 本論文は科研費 22300011 と 24300006 の助成を受けたものである.

参考文献

- [1] Daan Leijen and Erik Meijer, “Parsec: Direct style monadic parser combinators for the real world”, Technical Report UU-CS-2001-27, 2001
- [2] 福安 直樹, 山本晋一郎, 阿草清滋, “細粒度リポジトリに基づいた CASE ツール・プラットフォーム Sapid”, 情報処理学会論文誌, Vol.39, No.6, pp.1990-1998, 1998