リポジトリ分析用共通データスキーマに関する考察

松下 誠†

オープンソースソフトウェアに代表されるリポジトリ中心型のソフトウェア開発を対象とした分析を行う場合に必要とされるデータ分析用のデータスキーマについて，必要される用件や，その後に用いられる分析手法について述べる．

A Discussion Common Data Schema for Repository Mining

Makoto Matsushita†

This paper discusses about data schemas for software repositories, which are commonly used on open-source software development, including schema requirements and mining methods.

1. はじめに

オープンソースソフトウェア等のソフトウェア開発に代表されるような「ネットワーク上に分散した開発者同士がゆるやかに連携し，ソフトウェア開発を行う開発形態」（以降，オープンソース開発）が広く注目されている．また，開発を支援するための各種ツール群も，オープンソース開発を支援対象とすることが増えてきている．

一般にオープンソース開発では，以下のような開発用ツールが用いられる．

- 相互の連絡のための電子メール，メーリングリスト，およびそれらの履歴を保管するメールアーカイブ．
- 成果物を管理するための版管理システム．システムが提供するツールやリポジトリ，あるいはリポジトリを閲覧するためのツールなど．
- バグや変更要求の内容を管理するための案件管理システム．

「オープンソース開発では何が行われているか，あるいは，何が行われていたか」を分析することにより，過去の開発状況の把握だけでなく，現在および将来のソフトウェア開発時における問題の解決（既存ソフトウェアの再利用，具体的な修正内容の流用など）をもはかることができる．

しかし，多くの一般的なソフトウェア開発とは異なり，オープンソース開発では，明確なプロセス管理，あるいは前述した版管理システム等以外のプロダクト管理を行うための枠組みは存在していない．また，不特定多数の開発者が参加するという開発形態の特殊性から，何らかの決まった開発管理システムを新たに導入することは困難であると考えられる．よって，オープンソース開発を対象とした分析を行うためには，前述した既存のデータを分類，整理するためのデータスキーマが必要となる．

そこで本稿では，データ分析のためのデータスキーマとその分析例について，筆者がこれまでに行ってきた事例に基づいて述べ，スキーマとしてどのようなものが必要か，あるいは分析手法としてどのようなものが今後考えられるかについて述べる．

2. データスキーマ

ここでは，具体例として[2]で述べている CoxR について説明する．CoxR は電子メールと版管理システムの 2 つを対象として，以下にあげる 3 つのスキーマを定義している．

- CVS 情報

  版管理システムとして CVS を仮定した上で，1 ファイルに対する 1 リビジョンについての情報を保持するためのもの．具体的には，固有の識別番号，ファイル名，リビジョン番号，更新作業者，更新日時，キーワード，関連識別番号がある．

- Mail 情報

  1 通の電子メールに含まれている内容を表すためのもの．具体的には，固有の識別番号，送信者，送信日時，サブジェクト，キーワード，関連識別番号，更新作業者，更新日時，更新ファイル名，リビジョン番号がある．

† 大阪大学大学院情報科学研究科

- 統合情報

  CVS 情報と mail 情報との間にある類似関係を保持するためのもの. 具体的には, 固有の識別番号, CVS 情報にある識別番号, mail 情報にある識別番号がある.

CVS 情報のうち, キーワードについては, 各リビジョンを登録する際に記録されたコメントの中から, 特徴的な単語 (ファイル名や機能名など) を採用している. また, 関連識別番号については, 日時や作業者などの情報から「同時に行われた」と判断できる他のデータの識別番号の集合を持つ. mail 情報のうち, キーワードについては CVS 情報のものと同様, メール本文に記述された内容から特徴的な単語を抽出してそれを採用している. また, 関連識別番号については, 当該メールと同一の議論を行っている他のメールの識別番号の集合を持つ. ファイル名などの情報については, キーワードと同様にメール本文中で明示的に記述されている場合はその情報を保持する. 統合情報は, CVS 情報として記録されている内容に該当するものが mail 情報に含まれている場合 (たとえば, あるリビジョンで行われた修正内容の是非についての議論, など), その関係を保持するためのものである.

3. 分析手法

リポジトリに対する分析手法については多くの手法がこれまで提案されているが, ここではその一例として, [1]で述べている開発コミュニティ分析について述べる. この分析では, 前述した CoxR で定義したスキーマに, GNATS を対象としたバグ情報を加えたものを用いて, ある時点におけるあるファイルを指定した際, それに関連する作業者, 議論, ファイルやリビジョンの集合を分析して提示することを行う.

集合を求めるための分析手法としては, キーワードやファイル名の一致, 時間軸上での相関関係といった単純な類似関係のほか, 既存の作業内容に基づいた作業者に対する役割分担の分析を用いて, 「このモジュールにおいては, この開発者が全体をリードしている」といった内容も用いている.

4. 考察

オープンソース開発を対象としたソフトウェア開発の分析等は今後ますます広がりを見せると思われる. しかし, たとえば以下のような問題があると考える.

4.1. データスキーマと分析手法の役割分担

オープンソース開発の分析を行う場合, 何を抽象的なデータ構造としてデータスキーマとして定めるのか, また, 何を分析手法として定め, 動的な要求に応じて実行するか, といった問題がある. たとえば[2]では, データスキーマでは元データとさほど変わらない単純なモデルを採用しつつ, 統合情報などのように事前に分析した結果をもデータスキーマ中に定義して事前に保持する, ということを行っている.

データスキーマを単純にすると, 分析手法を実装する上で負担が大きくなるが, データスキーマを多くの分析手法で共有することが可能となる. また, 分析手法を単純にすると, データスキーマとして定められた内容にその応用範囲が制限される一方, 分析を高速に行うことが可能となり, 一般的に膨大となりがちなオープンソース開発の分析を用いる際には大きな利点となる.

4.2. データスキーマや分析手法の汎用性

[1][2]ともに, CVS, GNATS などといったオープンソース開発で使われている特定のシステムに依存したデータスキーマ, 分析手法を用いているが, その一方, 「版管理システム」や「バグ管理システム」といった抽象データをまず仮定した上で, その上にデータスキーマ, 分析手法を構築する手法も考えられる. 後者は汎用性を高めることができる一方, 抽象的な版管理システムとはどういうものか, というデータスキーマを定めるときと同種の問題をさらに抱えてしまうという問題や, 汎用性を求めるあまり, 有用なデータスキーマを構築しにくくなるといった問題もある.

5. 今後への期待

分析の目的に応じてデータスキーマや分析手法は選択されるべきであるが, 可能ならばなるべく汎用性のあるデータスキーマや分析手法が生まれることを希望する.

参考文献

[1] 佐々木, 松下, 井上:開発履歴情報に基づいたダイナミックコミュニティ選定支援手法, 電子情報学会技術報告 (2005, to appear).

[2] Matsushita, Sasaki, Tahara, Ishikawa, and Inoue: Integrated Open-Source Software Deveopment Activities Browser CoxR, Proc. 3rd Workshop on Open Source Software Engineering, pp.99-103 (2003)