

## 中大規模 OSS 開発における Information seeking のための可視化ツールの提案

沓 澤 脩<sup>†1</sup> 櫛 山 淳 雄<sup>†2</sup>

近年, GitHub や BitBucket などのソーシャルコーディングツールの誕生により, 誰でも興味のあるアプリケーションへの貢献が可能になった. しかし, 依然として新規参加者が必要な情報をプログラム上から見つけ出すコストは大きいままである. コンテキストに基づく情報の検索は Information seeking と呼ばれ, これはプログラム理解における重要なサブタスクであると認識されている. 本研究では, ソーシャルコーディングツール上に存在する情報を元に Information seeking を支援するための可視化ツールを提案する.

### A Proposal of Information Seeking Support Tool in Middle-Large-Scaled Open Source Software

SHU KUTSUZAWA<sup>†1</sup> and ATSUO HAZEYAMA<sup>†2</sup>

In recent years, everyone has been able to contribute to interested applications by the birth of social coding tool such as GitHub and BitBucket. However, there are great costs that new comers find required information from programs. Searches for information based on the context is called Information seeking, which has been recognized as an important sub-task in the program understanding. In this work, we propose a visualization tool for supporting the Information seeking by using the information of a social coding tool.

#### 1. はじめに

近年, GitHub<sup>?)</sup> や BitBucket<sup>?)</sup> などのソーシャルコーディングツールを用いた OSS(Open Source Software) 開発が盛んになってきている. これらのツールの誕生により, 誰でも自分の興味のあるプロジェクトのプログラムの取得, そのプロジェクトへの貢献が可能となった.

しかし, 取得したプログラムから変更のために必要な情報を取得することは容易ではない. プログラムの変更を行う際には, そのプログラムを理解する必要がある<sup>?)</sup> が, プログラムを理解するためには Information seeking が重要なサブタスクとなる<sup>?)</sup>. Information seeking(以下, IS) とは自分の置かれているコンテキストを考慮した上での意味のある情報検索を意味し, 開発期間の 60%を時間を消費していることが明らかになっている<sup>?)</sup>. IS は情報の欠落や点在により発生するものである.

そこで本研究では, ソーシャルコーディングツール

上の実体であるプログラムファイルに, その他の実体を関連付けて 1 つのグラフで可視化するツールの提案を行う. プログラムファイルを元とすることで IS 発生の原因になる情報の欠落を防ぎ, そのプログラムファイルに関する情報を関連付けることで, 情報の点在を防ぎ, 利用者の情報探索を支援することを目的としている. ここで, ソーシャルコーディングツール上の実体とは, 開発者, プログラムファイル, Issue, Pull request のことを指す. また, 今回のアプリケーション, 言語, ソーシャルコーディングツールの対象はそれぞれ, Web アプリケーション, Java, GitHub である.

#### 2. ソーシャルコーディングツールの情報構造

ソーシャルコーディングツールである GitHub は, Git によるバージョン管理の機能と, コミュニケーションの機能を有している. プログラム, Issue, Pull request はすべて開発者から生み出されるものであり, Pull request は開発者が変更を加えたプログラムファイルを対象にしている. そして, Issue では意思決定や, バグの報告などが行われている. このような関係を考えるとき, ソーシャルコーディングツールの情報構造はグラフ構造であるといえる.

<sup>†1</sup> 東京学芸大学大学院  
Graduate School of Education, Tokyo Gakugei University

<sup>†2</sup> 東京学芸大学  
Tokyo Gakugei University

### 3. 提案手法

本研究では、ソーシャルコーディングツール上の実体とその関連をグラフで可視化し、利用者の IS を支援するツールを提案する。実体とその関連を表??のように定めた。これは前節で述べたグラフ構造を表で表したものになる。

GitHub APIv3<sup>?)</sup> を用いることで、これらの実体の取得は可能である。しかし、それらの実体を関連付けることは API のみでは困難であるため、本提案ツールによって関連付けを行う。

表 1 ソーシャルコーディングツール上の実体と関連  
 Table 1 Entities and Relations of social coding tool

実体	関連
開発者 プログラムファイル	commit
Pull request プログラムファイル	change
Issue プログラムファイル	focus
プログラムファイル プログラムファイル	<シグネチャ>
<>無:その関連名を使用 <>有:該当するものを使用	

#### 3.1 関連付け機能

関連付けの方法は、Pull request や Pull request とプログラムファイルとの関連付けと、プログラムファイル同士の関連付けでは方法が異なる。前者は GitHub APIv3 を用いることで可能になる。後者についてはプログラムファイルに対し、静的解析を行う必要があるが、現在検討中である。今回は前者についてのみ述べる。

##### 3.1.1 Pull request との関連付け

GitHub で Pull request の内容を取り込むことを Merge と呼ぶ。本研究では Merge された Pull request を対象にする。GitHub では Pull request が Merge された際に、それが新たなコミットとなる。そのため、GitHub APIv3 を用いて対象リポジトリのコミットを取得する。その後そのコミットと Merge された Pull request を関連付ける。

##### 3.1.2 Issue との関連付け

Issue は実装機能の決定や、バグ報告などが行われている。この場で意思決定を行い、その機能を実装した際に Pull request で報告する様子がよく見受けられる。よって本研究では、プログラムファイルと紐付けられていた Pull request に Issue を紐付ける。また issue のコメントに直接プログラムファイル名が記述されている場合は、直接プログラムファイルと関連付ける。

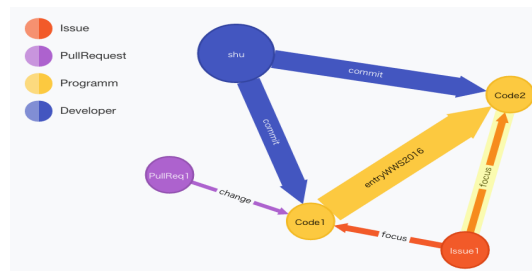


図 1 可視化イメージ  
 Fig.1 Image of visualization

#### 3.2 可視化機能

利用者のコンテキストに沿った検索要求と一致するグラフを可視化する機能である。可視化イメージは図??の通りである。これは表??を可視化したものである。本研究ではグラフ構造のデータを扱う DB として Neo4j<sup>?)</sup> を用いることとし、この図は Neo4j を用いて作成したものである。Neo4j は関連にもプロパティを持たせることが可能であり、図中の黄色の関連はシグネチャを表現しており、引数をプロパティとして持たせている。

#### 4. おわりに

本稿では、GitHub を対象としてソーシャルコーディングツール上のデータの活用し、IS を支援する可視化ツールについての提案を行った。本提案ツールは取得したプログラムに API を用いた関連付けと、静的解析を用いた関連付けを行う必要があるが、後者は未だ検討中である。後者の実装方法を明確にし、その後実装を行った後に、ツールの評価を行っていく予定である。

#### 参考文献

- 1) BitBucket: <https://bitbucket.org/>
- 2) GitHub: <https://github.com/>
- 3) GitHub APIv3 : <https://developer.github.com/v3/>
- 4) Liu, WenQian, et al., "A design for evidence-based software architecture research," REBSE2005, 2005.
- 5) Maalej, Walid, et al., "On the comprehension of program comprehension," TOSEM, Vol.23, No.4, 2014.
- 6) Neo4j. <http://neo4j.com/>
- 7) Sharif, Khaironi Y, et al., "An empirically-based characterization and quantification of information seeking through mailing lists during open source developers' software evolution," Information and Software Technology 57, pp.77-94, 2015.