

オープンソースソフトウェアの拡大再生産支援

渡 邊 卓 也

オープンソースソフトウェアは、それを再利用することによる拡大再生産によって量的に増大し続けている。本研究において提案するツール Selene は、類似プログラムを検索・提示することでコードの再利用を支援するツールである。プログラムに特有の構文・意味情報を用いるツールとは異なり、Selene では連想計算技術を用いてコード全体の字面上の一致度を計算することにより、大規模な貯蔵庫から類似プログラムを比較的高速に検索することが可能となった。

Supporting Expanded Reproduction of Open Source Software

WATANABE TAKUYA

Quantitative expansion of open source software has been made by expanded reproduction of itself through code reuse. We propose a code search engine Selene, which enhances code reuse by displaying relevant programs. Selene is different from existing tools which make use of structural and/or semantic information of programs, in that it calculates textual similarity of the entire code for which it employs associative calculation. This realizes relatively fast search from a huge repository of code.

1. はじめに

近年のオープンソースソフトウェアの爆発的な量的増大は、オープンソースソフトウェアへの需要の高まりや、それに伴う開発やそれを支援する共同体への参加者の増加といった要因で説明可能な部分も多い。しかし、それだけでなく、オープンソースソフトウェアの「オープン」な性質そのものによって、つまりコードを誰でも参照可能で、またそのコードを何らかの形で再利用することによって新たなソフトウェアの作成に利用することができるという性質により促進されているという側面も大きいものと考えられる。オープンソースソフトウェアの拡大再生産という好循環が働いているのである。

プログラムの作成中に既に存在する別のプログラムを参照、あるいはその一部を再利用することは作成効率の向上に有効であり、多くのプログラマが日常的に行っている。しかしながら、必要な時に常に目的のプログラムが手元にあり、かつすぐに見つけ出せるとは限らず、むしろ汎用の検索ツールや検索サービスを用いる等し、労力を掛けて検索しなければならない場合が多い。もし、何らかのプログラムを対象とした検索ツールにより、必要なプログラムをより少ない労力で発見できるならば、それによりオープンソースソフトウェアの拡大再生産が促進されることが期待出来る。

2. プログラム検索ツール Selene

本研究において提案するプログラム検索ツール Selene は、既存のツールとは異なる手法を用いることにより、再利用に適したプログラムをより容易に取得可能とすることを狙いとしている。プログラムの検索に特化した既存のツールは大きく二種類に分けることが出来る。一つはプログラムの構文・意味上の手掛かりを用いて検索するツール群であり、多くは統合開発環境上で動作する。もう一つは（一般には）少数の検索語（重要語、キーワード）を用いてそれを含むプログラムを探し出すツール群である。

Selene では、プログラムの構文・意味情報は用いない代わりに、プログラム全体を検索語として用い、その字面上の情報を全体的な類似度の算出に利用して検索を行っている。これは GETA²⁾ を用いた連想計算によるものであり、これによって大規模な貯蔵庫から比較的高速に、特定の言語に依存せずにある程度の精度を保って検索することを可能とするものである。

3. Selene の特徴

以下 Selene の特徴を具体例を用いて説明する。Selene は統合開発環境 Eclipse¹⁾ のプラグインとして動作する。図 1 において、プログラマは Java 言語で Swing ライブラリを用いて GUI アプリケーションソフトウェアを開発中であり、プログラムは途中まで書かれた

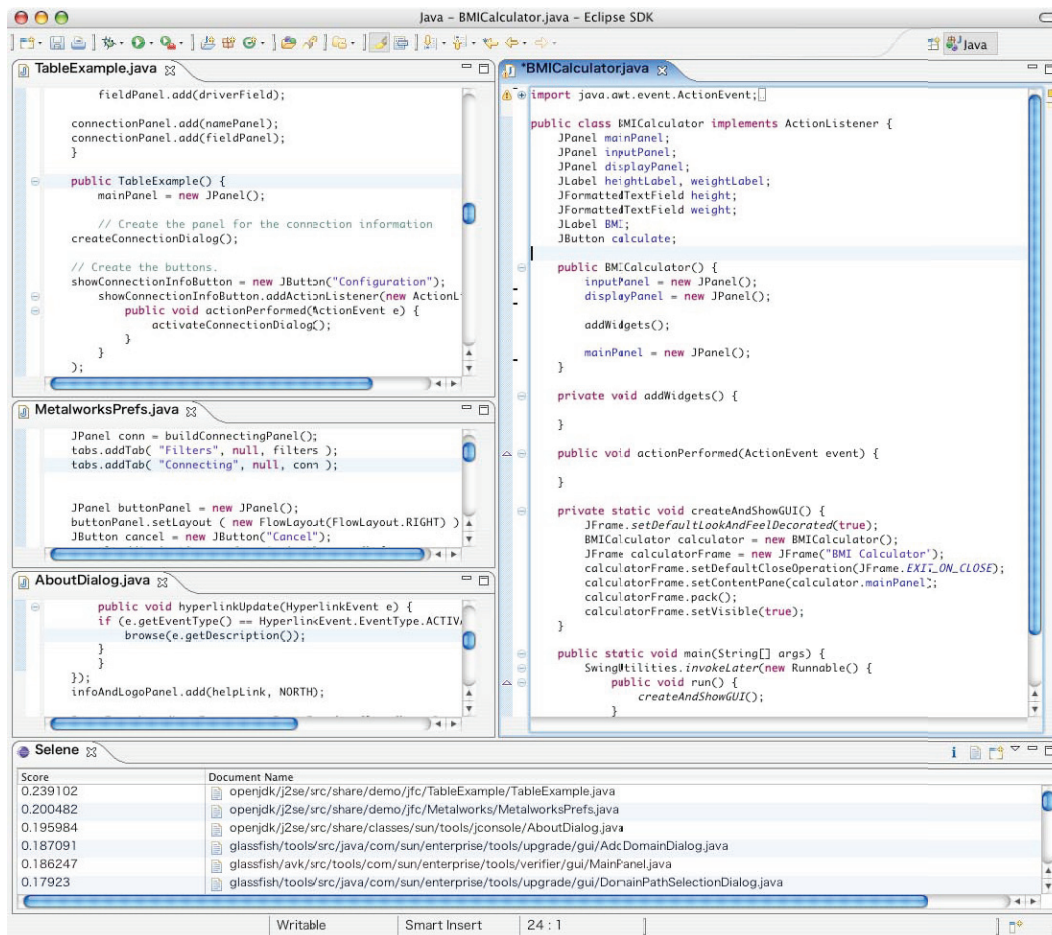


図 1 Selene の利用例

状態となっている。現在プログラマはコンストラクタを定義中で、既に使用する予定の JPanel オブジェクトのインスタンス化部分を書き終えており、次にこれらを組み合わせ、また内部の配置を指定するつもりである。ここで JPanel クラスの利用方法をよく憶えていない場合、API 定義文書を調査することも可能であるが、このようによく使われる部品は自分で以前に同様の場面で使用したことがある可能性があり、また既に誰かが同じ様に使用している可能性も高いので、そうしたプログラムを参考に出来ると便利である。

Selene はエディタでの編集操作が行われた場合に自動的にその編集集中のプログラムに類似したプログラムを検索し提示するので、プログラマはこうした場合にすぐに参考になりそうなプログラムを参照することが出来る。図 1 下側には類似度の順に貯蔵庫中の類似プログラムのファイル名が表示されており、左側にはそれらのうち上位三つの、特に関連が深いと判定された一部分が表示されている。狙い通り、JPanel を利用

している部分があり、左側上段からは add() メソッドによる JPanel オブジェクトの組み合わせ法、左側中段では setLayout() メソッドによる内部配置の指定法を読み取れる。プログラマはこれらを参考にして、あるいはそのまま複製して自分のプログラムを作成することが出来る。

もし参考になりそうなプログラムが提示されていなかった場合は、検索語として用いるプログラム全体の情報を均一に利用するのではなく、カーソル行とその周辺行を重視して検索することも出来る。また、指定したプログラムの全体を取得してエディタの編集対象として開くことも可能で、取得したプログラムから芋蔓的に検索を行うことが出来るので、比較的容易に目的のプログラムを見つけることが出来る。

参 考 文 献

- 1) Eclipse, <http://www.eclipse.org/>
- 2) 汎用連想計算エンジン, <http://geta.ex.nii.ac.jp/>