様式Ｆ－１９

科学研究費助成事業（学術研究助成基金助成金）研究成果報告書

平成 25 年 8 月 22 日現在

機関番号：14401
研究種目：挑戦的萌芽研究
研究期間：2011～2012
課題番号：23650015
研究課題名（和文）　オープンソースソフトウェアの氏・素性分析システムの開発

研究課題名（英文）　Developing Origin Analysis System for Open-Source Software

研究代表者
井上 克郎（Katsuro Inoue）
大阪大学・大学院情報科学研究科・教授
研究者番号：20168438

研究成果の概要（和文）：
オープンソースソフトウェアを対象として、氏・素性分析を行うシステム Ichi Tracker の開発を行った。システムを複数のオープンソースソフトウェアに対して適用することにより、着目したソフトウェアの派生関係を含めた、氏・素性を検出することが可能になる一方、もともとは同じソフトウェアであっても、進化の過程に伴って徐々に異なるソフトウェアへと変化していく状況を確認できた。
研究成果の概要（英文）：
We developed Ichi Tracker for open source software, a system for analyzing the origin of the code, and applied the system to open-source software among the Internet. Through the experiments, we found that the system depicts the origin and its derivation of similar software systems. We also found that the software system evolution changes causes software differ as time goes by.

交付決定額

（金額単位：円）

	直接経費	間接経費	合　計
交付決定額	1,700,000	510,000	2,210,000

研究分野：ソフトウェア工学
科研費の分科・細目：情報学・ソフトウェア
キーワード：オープンソースソフトウェア、分析システム

１．研究開始当初の背景

近年、OSS の開発は非常に活発で、いろいろな分野で高品質のソフトウェアを利用することが可能になってきている。企業にとっても、新たなソフトウェア製品を作る際には、OSS 全体やその一部を再利用することが日常的に行われており、製品の開発時間の短縮や信頼性の向上に非常に役立っている。OSS を利用するためには、そのライセンスを正しく認識し、それに従った利用方法を取る必要がある。間違った利用を行った際には、多額の賠償を請求されたり、製品回収を要求されたりする恐れもある。

今、利用したいソースコードファイルがあった時、そのソースコードの氏・素性を知ることは非常に重要である。「そのソースコードは、最初にいつ、誰によって作られたか。その時のライセンスは、どのようなものだったか（起源情報）」あるいは、「どこでどのように再利用されたか。また、その際、コードやライセンスはどのように修正されたか。（利用履歴情報）」といった、再利用に重要な属性を、ここでは氏・素性と呼ぶ。氏・素性が分かれば、開発者は、安心してそのソースコードを再利用して、製品開発を行うことができる。逆に、氏・素性がわからず、単にソースコードファイルがあるだけでは、安心して再利用することはできない。

オープンソースシステムでは、多くのソー

スコード片が繰り返しコピーされ、再利用されてきている。今、手元にあるソースコード片が、オープンソースシステムのどの部分からコピーされ、再利用されているかを知ること（氏・素性分析）は、そのソースコード片を再利用する上で非常に重要な情報である。氏・素性を理解することにより、開発者は安心してソースコードを再利用し、開発を進めることができる。

しかしながら、これらの情報は一般に広く知られてはおらず、個別に調査を行うしか方法がなかった。また、すでに提供されているソースコードを網羅的に調査するコストは非常に大きく、現実的な調査を行う方法はなかった。

２．研究の目的

近年、OSS の分析の研究は様々な視点から活発に行われている。例えば、International Working Conference on Mining Software Repository (MSR)では、OSS の類似性、コードクローン、ライセンスなどの分析結果の発表が、毎回、多数行われている。我々も、これらの分析技術を開発し、MSR や他の学会で多数発表してきてきた。

本課題では、これまで開発してきた分析技術を統合的に用いるとともに、新たにソースコードの収集や起源を分析する手法を開発し、実際に利用できるシステムの構築を行う。

これにより、OSS に対する氏・素性分析という概念を新たに確立する。また、その概念に従った分析技術を開発し、世の中への普及を目指す。

３．研究の方法

目的を達成するため、以下の手順で動作するシステムの構築を行い、実際に動作させて目的が実現できているか確認する。

本システムは、質問生成、類似コードファイル選別、属性抽出、属性集約整理の各サブシステムから構成されており、構造を持つテキストの検索、得られたソースコードファイルからの属性の抽出などに関して、新たな手法を開発し、全体として有用に機能するシステムとする。

最初にソフトウェア開発者は、氏・素性を調べようとする対象のソースコードファイル Q を本システムに入力する。入力されたソースコードファイルから、各ソースコード検索エンジンにふさわしいコード断片やキーワード、パラメーターを抽出し、それをそれぞれのソースコード検索エンジンに質問として投入する。現時点では、Google Code Search や Koders、Krugle を利用することを想定する。ここでは、ソースコードの構造をうまく探索するための手法の開発が検索エンジンごとに必要である。各ソースコード検索エンジンは、投入された質問にもっともふさわしいソースコードファイルをそれぞれのランク手法に応じた順序で出力するので、それを取得する。

各コード検索エンジンから得られたソースコードファイルと Q とを比較し、同じ構造を持つものを選別する。その際には、コードクローン検索技術や特徴抽出技術などを効果的に用いる。得られた類似コード片を含むソースコードファイル群を、類似ソースコードファイル群 X とする。X の各ファイルから種々の属性の抽出を行う。現在考えているのは、①ソースコードの配布ライセンスの検出、②ソースコードの著作権表示であるコピーライトの検出、そして、③作成日時であるタイムスタンプの検出である。それぞれについての詳細は次の通りである。

①ライセンスの検出は、現在、我々が研究中の複数知識ベースを用いたライセンス検出システムに対して、リファレンスとなるライセンスデータベースの拡充、ディレクトリや WEB サイトなどの置かれた間接的なライセンス記述の認識、多少の言葉の揺れや欠損での認識などの大幅な改良を施し、認識率の革新的向上を目指す。

②単にソースコード中のコメントにあるコピーライト文からその権利者を抽出するだけでなく、ソースコード検索エンジンに登録された著者やオーナー情報などを合わせて抽出を行う。

③そのソースコードファイルの作成日時に関して、ソースコードのタイムスタンプやコピーライトの日時、また、検索エンジンから得られる情報を収集する。

得られた情報を集約、整理して、開発者にわかりやすい形に整形する。特に注意すべき問題点、例えば、ライセンスの矛盾（組み合わせて使うファイル間で、許されないライセンスの組み合わせがあるなど）等の警告を出すようにする。集約整理した情報を Q の氏素性情報として開発者に提示する。

４．研究成果

計画に基づき、システム Ichi Tracker を開発した。

本システムは、与えられたコード片に対して、そのコードを含むファイルをユーザーに返す。まず、与えられた質問ソースコード片から外部検索エンジンに与えるキーワードを抽出する。次にそのキーワードを外部検索エンジンに与える。その際、Google Code Search, Koders, SPARS/R を外部検索エンジンとして利用した。それぞれの検索エンジンから得られた結果をマージし、十分な結果が得られたなら、それらの結果と質問ソースコード片との間のコードクローン分析を行い、十分似ているファイルのみを残す。

作成した Ichi Tracker に対して、いくつかのテストケースを用いて評価を行った。以下、3 つのテストケースについて述べる。

1) texture.java
1600 行の Java プログラムであり、3D ゲームソフトにおいて物体の描画を行う際に使われるものである。これをシステムの入力として与えた。
3 種類の検索エンジンに対して複数回の検索を行うことにより、ファイル中のキーワードからは 29 種類、ファイル名単体からは 26 種類の関連するソースコードを得た。その結果、ファイル群は 3 つの集合（入力としたファイルと完全に同一であるものと、3 割程度異なるものと、45%程度異なるもの）へ明確に分かれることがわかった。
また、入力ファイルはオリジナルの開発プロジェクトにおいて R3800 として認識されるバージョンであり、オリジナルのバージョン履歴から、ふぁいろがどのように派生してきたか、ということが明確にわかった。

2) kern_malloc.c
4.4BSD オペレーティングシステムで開発されている 1082 行の C プログラムであり、これを入力とした。最大 4 回の検索を行った結果、67 種類の関連するソースコードを得た。
これらを類似度と時間軸の 2 軸によって分類、整理した結果、前出の texture.java のようなクラスタ分類はできなかったものの、入力として与えたファイルが作成された時点からの経過時間が長ければ長いほど、ファイルの類似度が低くなっており、徐々に改変がなされていることが明白となった。また、比較的最近改変がなされているものについては、適用されるライセンスが変更されていることがわかった。この変更は、大本のプロジェクトの後継プロジェクトが採用したライセンス変更に追従したものであり、昔開発されたソースコードが時代の流れに従って徐々に改変され、ライセンスも変更されていることが明白に分析結果から得られた。

3) SSHTools
Java で記述された SSH 実装全体をシステムに対する入力（全 442 ファイルのうち、一定の大きさ以上である 339 ファイル）とした。入力は 1 ファイルづつ順に与え、どの程度類似するファイルをシステムが認識するかを確認した。この結果、34 ファイルは今プロジェクトに固有であること、それ以外についてもほとんどが関連ファイル数 10 以下であることがわかった。
関連ファイルの中身について、そのライセンス等も含めてシステムにて分析を行ったところ、多数のプロジェクトからそれぞれ寄せ集めて SSHTools を構成していることがわかった。さらに、引用元のプロジェクトが採用しているライセンスも複数にわたっており、SSHTools が採用するライセンスと同一のものだけでなく、異なるライセンスのものが混在していることがわかった。すなわち、SSHtools 全体としてはライセンスの整合性が取れていない疑いがあり、仮に利用する場合には、どの範囲を利用するのかを明白にしたうえで、自分が利用したいライセンスと、利用するファイルに付与されているライセンスが相反しないか、十分に考慮する必要があることがわかった。

これらの適用事例を通じて、多くの見識を得た。まず、作成したシステムは、利用者に対してライセンスや派生関係など、氏・素性情報を容易に、かつ機械的に出力できることを確認した。既存の検索エンジンを採用することにより、コンパクトな構成ながら、世界中で開発されている多数のオープンソースソフトウェアを網羅的にかつ現実的な時間で調査できるシステムとなっていることがわかった。一方で、関連ファイルを探索する際には、キーワード検索を工夫することなくファイル名を単に用いる手法も有用であることがわかった。また、本システムによって得られた結果は、誤った結果（関連がないフ

ァイルについて、関連があると判断する）を導かないことを確認した。

以上のことから、今回作成した Ichi Tracker は当初の目標を首尾よく達成できており、かつ、それを実際のオープンソースソフトウェアに対して適用することによって、多くの知見をシステムによって誰でも容易に得ることができ、現状のオープンソースソフトウェアに内在する問題も明示することができた。

５．主な発表論文等
（研究代表者、研究分担者及び連携研究者には下線）

〔雑誌論文〕（計１件）
Pei Xia, Yuki Manabe, Norihiro Yoshida, Katsuro Inoue: ″Development of a Code Clone Search Tool for Open Source Repositories″, コンピュータソフトウェア, Vol.29, No.3, pp.181-187 (2012).

〔学会発表〕（計２件）
Pei Xia, Yuki Manabe, Norihiro Yoshida, Katsuro Inoue: ″Development of a Code Clone Search Tool for Open Source Repositories″, 情報処理学会研究報告, Vol.2011-SE-174, No.2, pp.1-8 (2011).

Katsuro Inoue, Yusuke Sasaki, Pei Xia, Yuki Manabe: ″Where Does This Code Come from and Where Does It Go? - Integrated Code History Tracker for Open Source Systems -″, Proceedings of 34th International Conference on Software Engineering, pp.331-341 (2012).

６．研究組織
(1)研究代表者
井上 克郎（Katsuro Inoue）
大阪大学・大学院情報科学研究科・教授
研究者番号：20148438

(2)研究分担者
松下 誠（Makoto Matsushita）
大阪大学・大学院情報科学研究科・准教授
研究者番号：60304028

石尾 隆（Takashi Ishio）
大阪大学・大学院情報科学研究科・助教
研究者番号：60452413