ソースコードの編集内容を入力とした ソフトウェア部品の自動検索

島田 隆次† 市井 誠† 早瀬 康裕† 松下 誠† 井上 克郎†
†大阪大学大学院情報科学研究科

1 はじめに

ソフトウェア開発において，クラスやモジュールなどソフトウェア部品 (以下，単に部品) の再利用はソフトウェアの品質や生産性の向上に効果的だと言われている [1]．再利用の際に開発者は，部品を取得するためにキーワード検索による部品検索システムを用いることが多い．しかし適切なキーワードを選ぶためには検索する部品についての知識が必要になることがあり，再利用を困難にしている．また，開発者が検索を指示しないと検索が行われないため，開発者が存在を期待していない部品は検索されることがなく，再利用が行われないといった問題もある．

これらの問題に対して，Ye らにより開発者の指示なしにシステムが自動的に検索を行う部品自動検索が提案されている [2]．部品自動検索では，ソースコードの編集中にシステムが自動的に検索に必要な情報を収集して検索を行い，再利用可能な部品を開発者に提示する．

Ye らの手法は，メソッドのドキュメントコメント (機能等を説明するコメント) とシグネチャ(引数と戻値の型) に基づき自然言語に対する解析手法である LSA[3] を用いて類似部品を検索する．しかし検索に用いる情報が上記 2 種のみであるため，メソッドを書き始めた時にしか検索が行えないという問題がある．

そこで本稿では，より多くの状況で検索を行えるようにするために，利用しているメソッドや出現する識別子とその型の情報もソースコードから取得し，検索に用いるシステムを提案する．具体的には，メソッドの中身を書いている時やクラスの定義を書いている時にも自動検索を行うことができる．

Automatic Search for Software Components Based on Editing Contexts of Source Code
†Ryuji SHIMADA, †Makoto ICHII, †Yasuhiro HAYASE, †Makoto MATSUSHITA, †Katsuro INOUE
†Graduate School of Information Science and Technology, Osaka University

2 提案システム

本節では提案する検索処理とシステムの構成について述べる．

2.1 検索処理

本システムではクラスを部品として扱い，部品の特徴となりうる情報 (以下，特徴情報) を元に，蓄積された部品群から編集中のソースコードと類似した部品を検索する．特徴情報はドキュメントコメントに含まれる単語，利用しているメソッド名，ソースコード中に出現する変数を表す組<変数名，型>のいずれかである．検索にはベクトル空間モデルの応用である潜在的意味インデキシング LSI[4] を用いる．LSI では文章とそれに含まれる単語の出現回数を共起行列として表現するが，本システムでは文章に部品を，単語に特徴情報を対応させる．以下に手順を示す．

2.1.1 索引の構築

自動検索の実行に先立って，蓄積された部品群に含まれる特徴情報を解析し，検索に用いる索引を構築する．まず部品のソースコードを構文解析して特徴情報を抽出する．次に各部品の特徴情報を基に共起行列を作成する．この行列は行に部品，列に特徴情報をとり，ij 要素が i 番目の部品における j 番目の特徴情報の出現頻度である．また，この行列の行ベクトルを部品ベクトルと呼ぶ．この行列を入力として LSI を行う．

2.1.2 検索クエリの生成

編集中のソースコードから特徴情報を抽出し，検索条件を指定する検索クエリを生成する．検索クエリは組<特徴情報，重み>の集合である．重みはその検索においてその特徴情報が重要なほど大きな値をとる実数値である．

まずシステムが開発者によるソースコードの編集を監視し，セミコロンの入力，文の変更後の別の文へのカーソル移動，ドキュメントコメント変

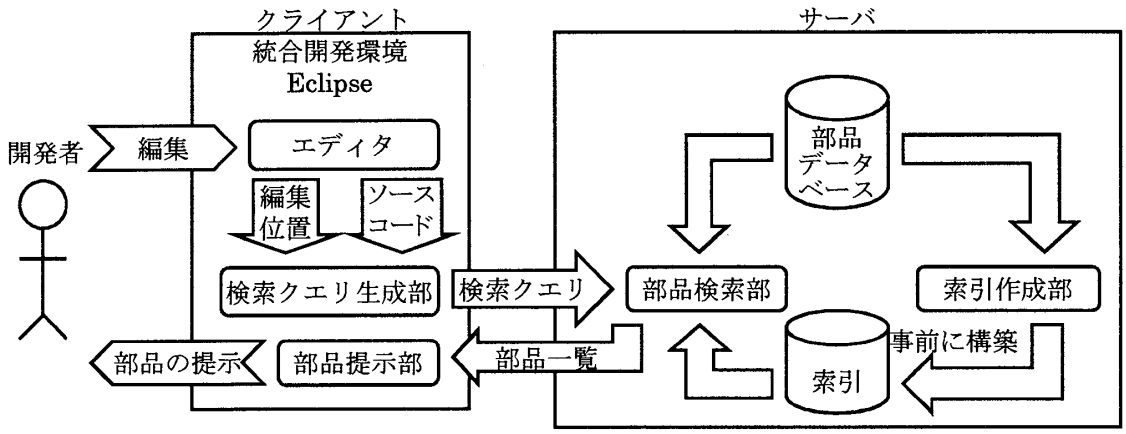

図 1: 提案システムの構成とデータの流れ

更後のコメント外へのカーソル移動などの編集の区切りを検出する．これらの区切りが検出されたら，システムは編集中のソースコード内にある特徴情報を抽出し，編集位置に近いほど大きな重みを付け，検索クエリを作成する．ただし同じ特徴情報が複数ある場合，重みはそれらの合計値を使う．

2.1.3 類似部品の検索

検索クエリを擬似部品ベクトルに変換し，LSIの手法に従って検索を行う．擬似部品ベクトルは，検索クエリに含まれる特徴情報の重みを，部品ベクトルと同じ順で並べたものである．

2.2 システム構成

システムの構成と検索処理におけるデータの流れを図 1 に示す．

システムはクライアントとサーバから成る．クライアントは開発者が編集している情報の取得と検索結果の提示を行い，サーバは蓄積された部品群に対する検索を行う．クライアントは Eclipse[5] をプラグインによって拡張したものである．

索引作成部は，部品のソースコードを格納した部品データベースを解析し，検索に用いる索引を作成する．検索クエリ生成部は，ソースコードの編集を監視し，検索クエリを生成する．生成された検索クエリは部品検索部に送信され，部品検索部は索引を用いて部品検索を行い，検索結果の部品を部品データベースから得る．部品提示部は得られた部品一覧を開発者に提示する．

3 まとめと今後の課題

本稿ではより多くの状況で利用できる自動検索システムを提案した．今後の課題として，システムの実装を進めること，検索時間と検索精度の評価のための大量の部品群を対象とした適用実験がある．

謝辞

本研究は，日本学術振興会科学研究費補助金萌芽研究（課題番号:18650006）の助成を得た．

参考文献

[1] Basili, V., Briand, L. and Melo, W.: How reuse influences productivity in object-oriented systems, *Comm.ACM*, Vol. 39, No. 10, pp. 104–116 (1996).

[2] Ye, Y. and Fischer, G.: Reuse-Conducive Development Environments, *Automated Software Engineering*, Vol. 12, No. 2, pp. 199–235 (2005).

[3] Landauer, T. and Dumais, S.: A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge, *Psychological Review*, Vol. 104, No. 2, pp. 211–240 (1997).

[4] Deerwester, S., Dumais, S., Furnas, G., Landauer, T. and Harshman, R.: Indexing by latent semantic analysis, *J.Am.Soc.Inf.Sci*, Vol. 41, No. 6, pp. 391–407 (1990).

[5] Eclipse, `http://www.eclipse.org/`.