メソッド名に用いられる動詞-目的語関係の辞書作成にむけて

鹿島　悠[†1]　早瀬康裕[†1]　井上克郎[†1]

オブジェクト指向プログラムにおいて，メソッド名はプログラムの動作を表わす重要な部分であるため，メソッド名に不適切な命名が行なわれた場合，プログラムの理解が困難になることが知られている．本研究では，メソッド名やシグネチャに用いられる動詞-目的語の組を収録した辞書の作成方法を提案する．

A Dictionary of Verb-Object Relations in Method Names

Yu Kashima,[†1] Yasuhiro Hayase[†1] and Katsuro Inoue[†1]

In Object-Oriented Programs, names of methods have an important roll because they notice actions of a program. Therefore, irrelevant names of methods make program comprehension difficult. This paper proposes an approach to create a dictionary of verb-object relations in method names.

1. はじめに

オブジェクト指向プログラムにおいて，メソッド名はプログラムの動作を表わす重要な部分である．メソッド名は動詞や動詞句で始まることが推奨され[3)]，その後に目的語が続く場合が多い．例えば，changeValue()というメソッドでは change が動詞，Value がその目的語である．もし，メソッドに不適切な命名がなされた場合，プログラムを利用しようとしている開発者に誤解や混乱を招く．

そこで，本研究では開発者に適切なメソッドへの命名を促すために，メソッド名に頻繁に使われる単語の例を収録した辞書を作ることを目指す．具体的には，多くのプログラム中で使用されている，動詞-目的語の組を収録した辞書を作成する手法を提案する．プログラム解析ツールの出力には大量の識別子が出現するため，提案する辞書によってプログラム解析ツールをより簡単に使用することが出来るようになると期待される．

2. 提案手法

図 1 に本手法の概要を示す．本手法の入力は Java プログラムのソースコード集合であり，出力はプログラム中の動詞-直接目的語-間接目的語の関係を表わす三つ組を記述した辞書である．ただし，間接目的語は空である場合がある．

本手法では，まず，Java ソースコード集合を解析し，メソッドのシグネチャやメソッドを定義しているクラスの名前を取得する．その後メソッド名をアンダースコアや CamelCase で分解し，単語列に分解する．その単語列に対し品詞解析を行い，単語ごとの品詞情報を決定する．そして，切り分けた単語のうち，連続する名詞と，形容詞の後に名詞が続く部分を結合し，1つの名詞として扱う．また，クラス名とメソッドの引数は，全て一つの名詞として扱う．

本手法では，動詞-目的語の組の抽出ルールを事前に複数作成しておく．そして，品詞情報を付加されたシグネチャとクラス名に対して抽出ルールを適用し，動詞-目的語の組の抽出を行なう．個々の抽出ルールは，特定の品詞列で構成されるシグネチャとクラス名を対象に，どの単語を動詞-直接目的語-間接目的語と

図 2 抽出ルールの適用例

†1 大阪大学 情報科学研究科
Osaka University

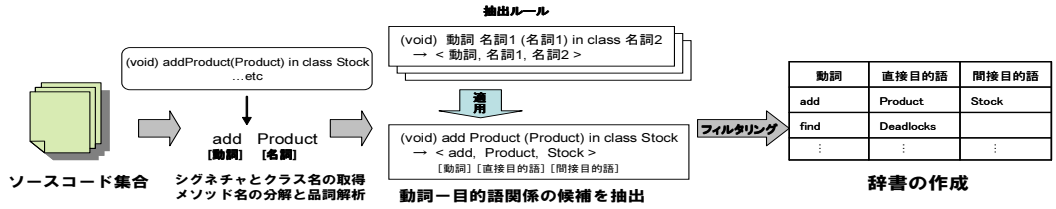

図 1 手法の概要

して抽出するかを定めたものである．例を**図 2** に示す．事前に定めた抽出ルールが図中の [1] である．ソースコード集合より (void)addProduct(Product) in class Stock というシグネチャとクラス名を取得したとき，[2] のように品詞情報が決定される．そして，[2] のシグネチャは [1] のルールに適合するため，[3] の通り動詞-直接目的語-間接目的語の組として，<add，Product，Stock>を得る．

最後に，抽出された動詞-直接目的語-間接目的語の組のうち，多くのソフトウェアで出現した組は一般的な動詞-目的語の関係であると考えられるので，辞書に収録する．

3. 予 備 実 験

抽出ルールを作成するため，予備実験を行った．予備実験では，sourceforge.net で公開されている Java で記述されたソフトウェアの中から無作為に 11 個を選び，それらのソフトウェアに含まれるソースコードを読み，人手で抽出ルールを作成した．具体的には，メソッド名の先頭語が動詞であるものを対象に，メソッド名が動詞のみまたは動詞と動詞の直後に続く名詞だけで構成され，引数が 0 個または 1 個の場合や，メソッド名中に直接目的語と間接目的語が書かれていると考えた場合の抽出ルールを作成した．作成したルールの例を以下に示す．

- (名詞 1) 動詞 名詞 1() in class 名詞 2
  → <動詞，名詞 1，名詞 2>
- (*) 動詞 名詞 1 前置詞 名詞 2 (*) in class *
  → <動詞，名詞 1，名詞 2>

(*は任意の語句を表す)

抽出ルールの作成後，作成した抽出ルールを用いて対象とした 11 個のソフトウェアのソースコードから動詞-目的語の組の抽出を行なった．その結果抽出できた動詞，直接目的語，間接目的語，動詞-直接目的語-間接目的語の組の，総数と複数のソフトウェアに跨って出現した数を**表 1** に示した．

表 1 各品詞と動詞-目的語の組の出現回数

		出現ソフトウェア数	
	総数	単独	複数
動詞	155	77	78
直接目的語	5378	4875	503
間接目的語	2869	2719	150
動詞-直接目的語-間接目的語	14501	14433	68

以下に，予備実験により得られた知見を記す．まず，ソフトウェア毎に識別子中で使用される単語が異なるため，同じ三つ組が出現することは非常に少ないことが分かった．一方，単一のソフトウェア内では似通った単語と命名方法が使われている場合があった．例えば，あるソフトウェアでは,「add + 形容詞 + Consist」という識別子名が多く出現していた．さらに，ソフトウェア中で使われる動詞について調査したところ，get や add， find などの一般的な動詞はほとんどのソフトウェアに出現していたが，adapt や contribute といった 1 つのソフトウェアにしか出現しない動詞も存在していた．

4. 将来の課題

より多くの関係を取得するために，メソッド名が前置詞を複数含む場合に対する抽出ルールや，複数の引数を持つメソッドに対する抽出ルールの作成が必要である．また，適用実験として，提案手法を大規模なソフトウェア集合に対して適用し，得られた関係を評価する必要もある．さらに，作成する辞書を効果的に開発者へ提供する方法も考案する必要がある．

謝辞 本研究は科研費 (21700031) の助成を受けた．

参 考 文 献

1) Fry, Z.P., Shepherd, D., Hill, E., Pollock, L. and Vijay-Shanker, K.: Analysing source code: looking for useful verb-direct object pairs in all the right places., *IET Software*, Vol. 2, No. 1, pp.27–36 (2008).
2) Hill, E., Pollock, L. and Vijay-Shanker, K.: Automatically capturing source code context of NL-queries for software maintenance and reuse, *Proc. 31st ICSE*, pp.232–242 (2009).
3) Sun Mycrosistems: Sun Java Tutorial, `http://java.sun.com/docs/books/tutorial/java/javaOO/methods.html`.