

特別研究報告

題目

識別子の命名支援を目的とした動詞-目的語関係の辞書構築

指導教員

井上 克郎 教授

報告者

鹿島 悠

平成 22 年 2 月 16 日

大阪大学 基礎工学部 情報科学科

内容梗概

ソースコード中の識別子の名前はソフトウェアが対象としているドメインの知識とプログラムの要素との対応付けを行う重要な手掛りとなっている。そのため識別子へ不適切な命名がなされたソースコードは、適切な命名がなされたソースコードに比べ、読解に大きな時間を要する。開発者が識別子に対し不適切な命名をしてしまう原因として開発中のソフトウェアが対象としているドメインの知識や類似プログラムの作成経験が不足している場合が考えられる。これらの知識や経験を開発者全員に学習させるのは、ドメインの知識が複雑であり、開発者の人数が多くなっているという理由により難しい。

ここで、オブジェクト指向プログラムでは、オブジェクトに対して操作を行う処理が多数出現する。操作を動詞、オブジェクトを目的語と考えるとこの処理には動詞-目的語の関係が出現していると言える。そのため動詞-目的語の関係を提示することでオブジェクトや操作の命名の支援を行うことができると考えた。

しかし、ソースコード中に出現する動詞-目的語関係は自然言語における動詞-目的語関係と異なっている場合が多く、自然言語に出現する動詞-目的語関係を提示しても適切な支援を行えない。この問題の解決のためにはソースコード中から抽出した動詞-目的語関係を提示することが有効である。

そこで本研究では、適切な命名支援を行うためにオブジェクト指向プログラムのソースコード中に出現する動詞-目的語関係を抽出し、その関係を体系的に整理した辞書の構築を行った。オブジェクト指向プログラム中でオブジェクトに対する操作が記述されるのはメソッドであり、動詞-目的語関係はメソッドに出現すると考え、メソッドから動詞-目的語関係の抽出を行った。具体的には、メソッド中に含まれる単語の品詞解析を行う。また、特定の品詞を持つ単語で構成されるメソッドから動詞-目的語関係を抽出するパターンを事前に定義しておく。そしてメソッドに対する品詞解析の結果と抽出パターンとを照合することでメソッドから動詞-目的語を抽出する。そして、抽出された動詞-目的語関係に対しふるい分けを行い、多くのソフトウェアで出現した関係を残すことで、あるソフトウェアだけにしか出現しない関係を除去する。最後に、残った関係を辞書に収録し、動詞-目的語関係の辞書を構築する。

評価実験では提案手法を用いて4つのドメインの辞書を構築した。そして辞書に収録された各三つ組が命名支援のための辞書に収録される動詞-目的語関係として適当であるかどうか

かをアンケートにより調査した。その結果ソースコード中から抽出した動詞-目的語関係の辞書が命名支援に有効であることを確認した。

主な用語

静的解析

動詞

目的語

識別子

命名支援

目次

1	はじめに	5
2	背景	7
2.1	オブジェクト指向プログラム中の単語間の関係	7
2.1.1	オブジェクト指向プログラムのクラスとメソッド	7
2.1.2	オブジェクト指向プログラムに出現する動詞-目的語の関係	8
2.2	オブジェクト指向プログラム中で用いられる識別子の命名規則	8
2.2.1	ソースコード中での複合語の表記法	8
2.2.2	オブジェクト指向プログラムのメソッドの命名規則	10
2.2.3	オブジェクト指向プログラムの特殊なメソッド名	10
3	提案手法	11
3.1	概要	11
3.2	ソースコード集合の解析	11
3.3	各メソッドから動詞-直接目的語-間接目的語の三つ組を抽出	11
3.3.1	メソッド名の分解	14
3.3.2	メソッド名の品詞解析	15
3.3.3	戻り値の型名・引数・クラス名の品詞解析	15
3.3.4	抽出パターンを用いた動詞-直接目的語-間接目的語の三つ組の抽出	16
3.4	抽出した動詞-直接目的語-間接目的語のフィルタリング	19
4	適用実験	21
4.1	実験目的	21
4.2	実験対象	21
4.3	実験方法	26
4.4	実験結果	27
4.4.1	Web Application	28
4.4.2	XML	31
4.4.3	Database	34
4.4.4	GUI	37
4.5	妥当性	40
4.6	辞書に収録するのに適当でない理由	40
4.7	その他被験者から頂いたコメント	41

4.8 考察	41
5 関連研究	42
6 まとめと今後の課題	43
謝辞	44
参考文献	45

1 はじめに

近年、ソフトウェアの肥大化に伴いソフトウェア保守のコストの増大が問題となっている。ソフトウェア保守を行う際、保守作業者は保守を行うソフトウェアを詳細に理解する必要がある。そのためにはソースコードを読解する必要がある。その際保守作業者は識別子の名前から関数や変数の役割や機能を類推することが多い。そのため、識別子に対し不適切な命名がなされた場合、適切な命名が行われたソースコードに比べて、ソースコードの理解により多くの時間が掛ってしまうことが知られている [8]。

開発者が識別子に対して不適切な命名をしてしまう理由として、類似プログラムの作成経験や開発対象のドメインの知識が不足している場合が考えられる。このような知識や経験の不足を埋めるために開発者に対して学習を行うのは、(1) ドメインの知識が複雑である (2) 経験を学習させるのは難しい (3) 開発者の人数が多いため全員に学習するには時間や手間がかかりすぎる、といった理由から難しい。そこで、開発者が適切な命名を探るための参考となる辞書があることが望ましい。

ソースコード中では、自然言語と同様に様々な単語を組み合わせることで、意味を表現している。例えば、関数やメソッドの名前は、動詞とその目的語の組み合わせで処理の意味を表現している場合がある。よって単語間の関係を例示し、開発者に適切な組合せを探る手掛かりを与えることにより識別子の命名支援を行うことができると考えられる。

ソースコード中に出現する単語間の関係と自然言語での単語間の関係は異なっている場合が多い。例えば、Java 標準 API [1] 中の `java.util.Calendar` クラスの `complete()` メソッドは、`complete` が動詞、`Calendar` がその直接目的語となっていると考えられるが、この動詞-目的語関係は自然言語ではあまり見られない。このため、自然言語を対象とした辞書に含まれる動詞-目的語関係を例示しても、その関係はソースコード中に出現する関係とは異なっている場合があり、十分な命名支援が行えないと考えられる。

この問題は、ソースコード中に出現する識別子間の関係を体系付けて整理した辞書を新規に構築することにより解決できる。我々の研究グループでは単語間の関係をソースコード中から抽出した辞書を構築する手法を研究している。我々の研究グループの早瀬ら [16] はプログラム中の単語の上位/下位関係、修飾関係を収録した辞書を開発した。

本研究では、オブジェクト指向プログラムでは、あるオブジェクト A に対して操作 B を行うという処理が数多く出現することに着目する。この処理は B を動詞、A を目的語と考えると、動詞-目的語の関係として表現されていると言える。従って操作やオブジェクトに対する適切な命名の支援に、これらソースコード中に出現する動詞-目的語関係を体系的に整理した辞書を用いることができると考えた。

本研究では、入力としてソースコード集合を与え、与えられたソースコード集合中の動

動詞-目的語関係を体系的に整理した辞書を作成する手法を提案する。動詞-目的語関係はオブジェクトに対する操作が記述されるメソッドとメソッド名、戻り値、引数、メソッドを定義しているクラスに出現する。これらに出現する単語に対し品詞解析を行い、品詞解析の結果と事前に定義した動詞-目的語関係を抽出するパターンとの照合を行うことで動詞-目的語関係を抽出した。さらに、抽出した動詞-目的語関係が入力されたソースコード集合中でどの程度一般的な関係であったかを評価し、抽出した動詞-目的語関係の中でも一般的な関係を収録した辞書を構築した。

提案手法を実装し、実装を用いた適用実験を行った。適用実験では、実装したプログラムを用いて、いくつかのドメインの辞書を作成し、辞書間の比較を行って、ドメイン間の違いがどのように辞書に反映されていたかを調査した。また、作成された辞書をドメインの専門家に評価して頂いた。その結果作成した辞書に収録された動詞-目的語関係がドメイン特有の関係やオブジェクト指向プログラム全般で出現する関係を収録していることを確認した。

以降、2節では本研究の背景となった概念や関連研究に関して述べる。2.1.2節では、オブジェクト指向プログラム中の動詞-目的語関係について詳説する。3節では、辞書の構築手法に関して述べる。4節では、適用実験とその考察を述べる。6節では、本研究のまとめと今後の課題に関して述べる。

2 背景

近年、ソフトウェアの肥大化によるソフトウェア保守コストの増大が問題となっている。保守を行うためには、保守対象のソフトウェアを詳細に理解しなければならず、そのためにはソースコードを読解しなければならない [6].

ソースコードを読解する際、保守作業者は様々な手掛りを利用するが、ソースコード中に出現する識別子の名前は重要な手掛りの一つとなっている。具体的には、保守作業者がソースコードを理解する際には、そのプログラムのドメインの知識とプログラム要素との対応付けに識別子の名前を用いる [11, 14]. このため、識別子に適切な名前が用いられていない場合は、ソースコードの理解に莫大な時間を費すことになってしまう [8]. 従って、識別子への適切な命名は保守コストの引き下げのために大変重要である。

オブジェクト指向プログラム中の識別子は、いくつかの単語を複雑に組み合わせて意味を表わしている場合が多く、適切な命名を行うには適切に単語を組み合せる必要がある。そのため、識別子の命名支援のためには単語間の関係を例示することが有効であると考えられる。

本節では、オブジェクト指向プログラム中で出現する単語間の関係とオブジェクト指向プログラムにおける識別子の名前の特徴について述べる。

2.1 オブジェクト指向プログラム中の単語間の関係

本節では、オブジェクト指向プログラム中に出現する単語間の関係について述べる。まず、オブジェクト指向プログラムの特徴について述べ、その後オブジェクト指向プログラム中出现する単語間の関係の具体例について述べる。

2.1.1 オブジェクト指向プログラムのクラスとメソッド

多くのオブジェクト指向プログラムには、クラスという概念が存在する。クラスはインスタンスの生成器であり、データメンバとメソッドを保持する。

メソッドは、C 言語等における関数とほぼ同等の機能を持ち、オブジェクトに対する操作が規定される場合が多い。ソースコード中でのメソッドを呼び出しは、(クラス名またはインスタンス名).(メソッド名)(引数) という表記をされる場合が多い。以降の節では、この一連の表記に含まれるメソッドを定義しているクラスやメソッドを呼び出しているインスタンス、メソッドの戻り値、メソッド名、メソッドの引数の集合を指してメソッドの呼び出しに関連する部分と呼ぶ。

2.1.2 オブジェクト指向プログラムに出現する動詞-目的語の関係

オブジェクト指向プログラムでは、あるオブジェクト A を対象に操作 B を実行する、という処理が多く出現する。ここで、B を動詞、A を目的語と考えると、この処理は「A に対して B する」という表現でき、A と B の関係は自然言語に出現する動詞と目的語の関係に類似している。オブジェクトを対象に操作を行うことを示す表現が多く出現するのは、メソッドの呼び出しに関連する部分であり、動詞と目的語の関係もメソッド呼び出しに多く出現していると考えられる。実際、メソッド名には動詞が出現することが多く、メソッド名中動詞の後に目的語が続く場合や、引数やメソッドを定義しているクラスの名前に目的語が出現することがしばしばあり、Fry[2] らは動詞をメソッド名から、直接目的語をメソッド名中の動詞の後の語や引数やクラス名から抽出する方法を提案している。

ここで、オブジェクト指向プログラムで出現する動詞と目的語の関係の特徴として、自然言語で出現する動詞と目的語の関係と異なっている場合が多いという点が挙げられる。例えば、Java 標準 API[1] の `java.net.Socket` クラスには `bind(SocketAddress)` というメソッドがあるが、これは `SocketAddress` を `Socket` に `bind` するという意味を示している。このような動詞と目的語の関係は自然言語には滅多に出現しない。

また、ソースコード中での動詞と目的語の関係はプログラムが対象としているドメインごとに異なっている場合も多い。なぜならば、ドメインが異なっているプログラムのソースコードの間では、出現する単語そのものや、単語の意味や用法が異なっている場合が多いからである。例えば、GUI を持つプログラムのソースコード中では、`Button` や `Toolbar` と言った単語が多く出現するが、CUI しか持たないプログラムのソースコード中ではそのような単語はあまり出現しない。

2.2 オブジェクト指向プログラム中で用いられる識別子の命名規則

本節では、オブジェクト指向プログラム中の識別子の命名に頻出する規則について説明する。

2.2.1 ソースコード中での複合語の表記法

ソースコード中では複合語の単語境界に空白文字を使用することができないため、一般的にソースコード中の識別子に出現する複合語はひと綴りの単語として表記されることが多い。

複合語の代表的な表記法には、`Camel Case` と `Snake Case` がある。`Camel Case` は要素語の先頭文字を大文字で記す表記法である。`Snake Case` はアンダースコアを用いて単語境界を表わす表記法である。`Camel Case` の例を図 1 に、`Snake Case` の例を図 2 に示した。

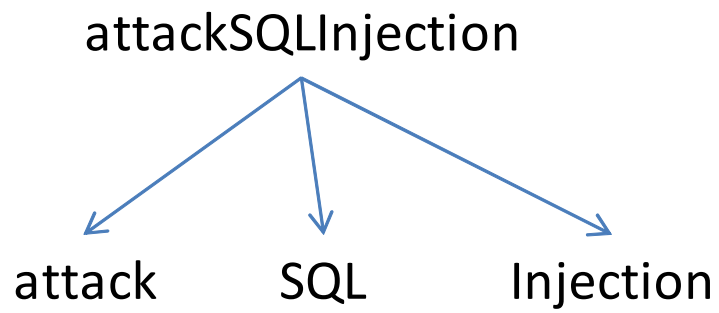


図 1: Camel Case の例

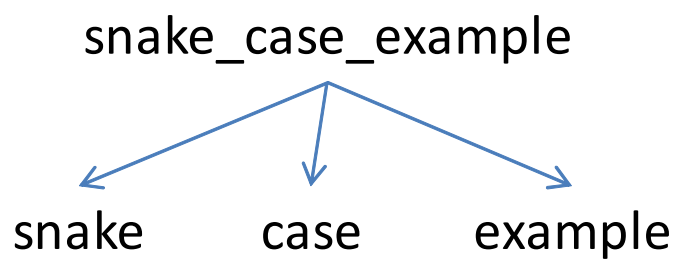


図 2: Snake Case の例

2.2.2 オブジェクト指向プログラムのメソッドの命名規則

オブジェクト指向プログラムのメソッドは Sun の The Java Tutorials[13] で示されているように、動詞または動詞句をメソッド名の先頭に置き、その後に名詞や形容詞が続く場合が多い。また、Java 標準 API[1] の `java.awt.event.ActionListener` インターフェース中のメソッド `actionPerformed(ActionEvent e)` のように、名詞や形容詞がメソッド名の先頭に置かれ、メソッド名の末尾に過去分詞の動詞が置かれる場合も多い。

2.2.3 オブジェクト指向プログラムの特殊なメソッド名

`java.lang.Object` クラスの `toString()` メソッドや、`java.lang.Class` クラスの `newInstance()` メソッドには動詞は出現しない。これらのメソッドは、(1) 本来あるはずの動詞が省略されている (2) `to` や `new` が動詞として使われている、のどちらかのと考えられる。

3 提案手法

本節では、オブジェクト指向プログラムのソースコード中に出現する動詞-目的語関係を抽出し、その関係を収録した辞書を作成する手法について述べる。本手法では、動詞-目的語関係を動詞-直接目的語-間接目的語の三つ組で表現する。ただし、間接目的語は空となる場合がある。動詞-直接目的語-間接目的語の三つ組の抽出については、動詞はメソッド名中の動詞から、直接目的語と間接目的語はメソッド名中の名詞、仮引数の型名と名前、メソッドを定義しているクラスの名前から抽出する。

3.1 概要

本手法の概要を図3で示す。本手法の入力はオブジェクト指向プログラミング言語で書かれたソフトウェアのソースコード集合であり、出力はソースコード中の動詞-目的語関係を表わす動詞-直接目的語-間接目的語の三つ組を収録した辞書である。実装では、Javaで書かれたソースコード集合を入力として与える。

本手法は具体的には以下のステップからなる。

ステップ1 ソースコード集合の解析を行い、全メソッドから動詞-直接目的語-間接目的語の三つ組の抽出に必要な各メソッドの情報を抽出する

ステップ2 ステップ1で抽出した情報を基に、各メソッドから動詞-直接目的語-間接目的語の三つ組を抽出する

ステップ3 抽出した動詞-直接目的語-間接目的語の三つ組に対してフィルタリングを行い、入力となったソースコード集合中で複数のソフトウェアに出現する動詞-直接目的語-間接目的語の三つ組を辞書に収録する

以降各ステップを詳説する。

3.2 ソースコード集合の解析

本手法では、入力として渡されたソースコード集合に対して字句解析、構文解析、意味解析を行い、全メソッドについて図4で示す情報を抽出する。実装では、ソースコード集合の解析にMASU[9]を用いる。

3.3 各メソッドから動詞-直接目的語-間接目的語の三つ組を抽出

図4に示した情報を用いて、各メソッドから動詞-直接目的語-間接目的語の三つ組の抽出を行う。処理の概要を図5に示す。三つ組の抽出処理は以下のステップに分解される。



Javaソースコード集合



(void) setPassword(Password p) in class User
(ElemInfo) getElemInfor() in class List
.....etc



動詞	直接目的語	間接目的語
Set	Password	User
Get	ElementInfo	List
...etc		

動詞-直接目的語-間接目的語の三つ組



動詞-目的語関係辞書

図 3: 提案手法の概要

- メソッド名
- 戻り値の型名
- 各仮引数の型名と名前
- メソッドを定義しているクラスの名前

図 4: メソッド情報の要素

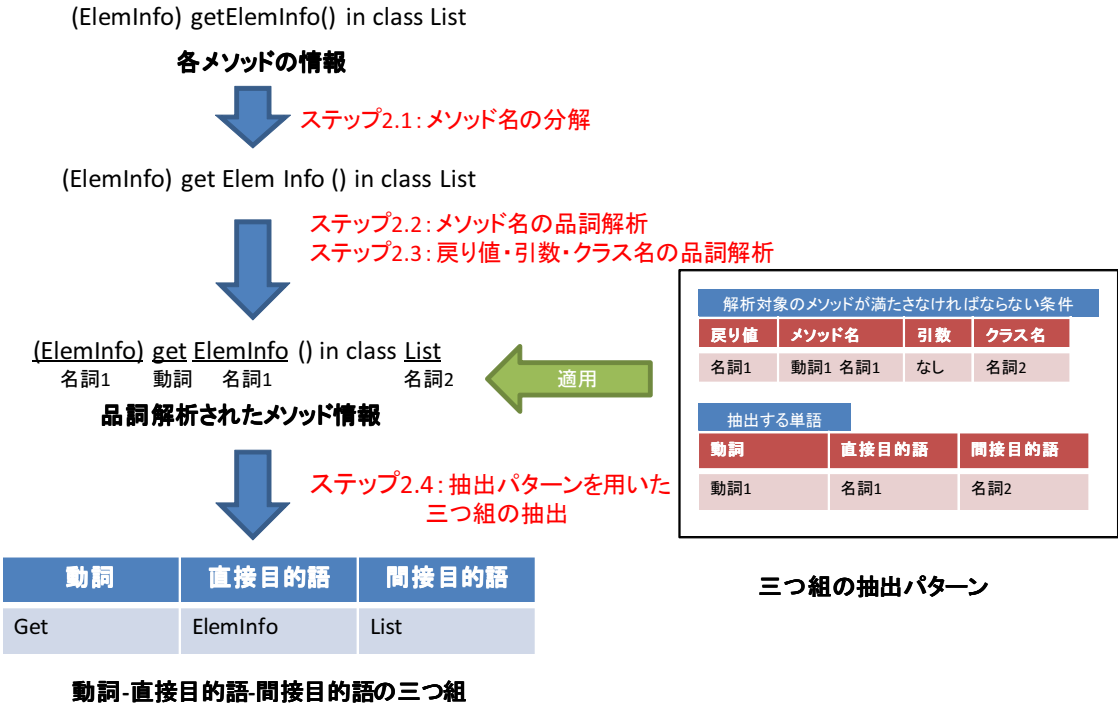


図 5: 動詞-直接目的語-間接目的語の三つ組抽出処理の概要

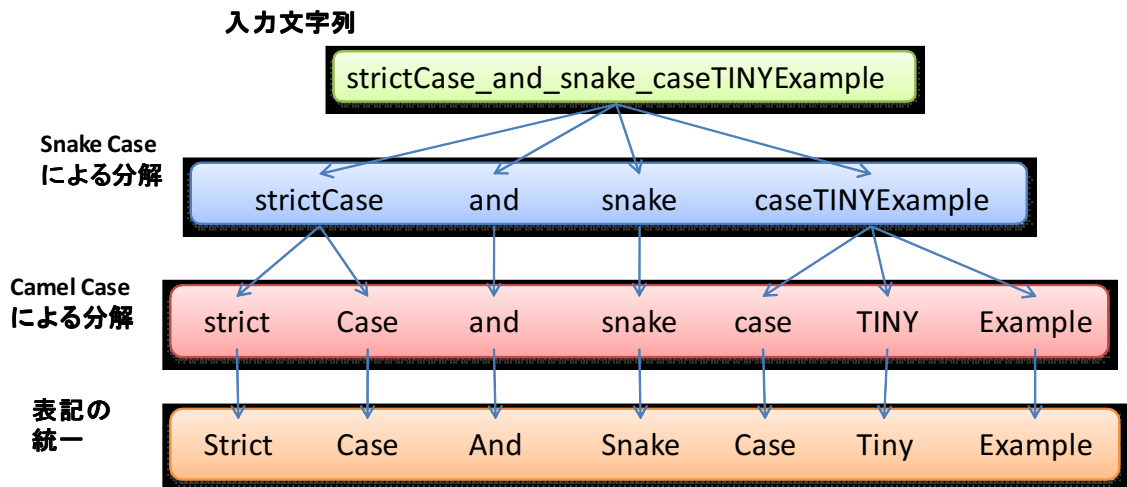


図 6: 複合語の分解処理

ステップ 2.1 メソッド名の分解

ステップ 2.2 メソッド名の品詞解析

ステップ 2.3 戻り値の型名, 引数, クラス名の品詞解析

ステップ 2.4 抽出パターンを用いた動詞-直接目的語-間接目的語の三つ組の抽出

以降これらの処理を詳説する.

3.3.1 メソッド名の分解

メソッド名は一般に複合語となっている場合が多いため, これを単語列に分解する. 具体的には, まず, 複合語が Snake Case に従って表記されていると仮定し, アンダースコアを単語境界とみなし, 分解する. 次に分解により生成した単語列中の各単語に対し, 各単語が Camel Case の表記法に従って表記されていると仮定して小文字と大文字の間を単語境界とみなし, 分解を行う. 最後に, 単語列中の各単語に対し, 大文字と小文字の表記を揃える処理を行う. 具体的には, 各要素語の先頭文字を大文字にして, 残りの文字を小文字にする.

「strictCase_and_snake_caseTINYExample」という複合語を説明した方法を用いて要素語に分解する例を図 6 に示した. この複合語は最終的に, 「Strict」「Case」「And」「Snake」「Case」「Tiny」「Example」という単語に分解される.

3.3.2 メソッド名の品詞解析

メソッド名の分解により得た単語列に対し、品詞解析を行う。まず自然言語用の品詞解析器に単語列となったメソッド名を入力し、品詞解析を行う。ただし、自然言語用の品詞解析器ではメソッド名を上手く品詞解析できないので、修正処理を行う。なお、実装では自然言語用の品詞解析器として OpenNLP[10] を用いた。

最初に、メソッド名の先頭で出現する `to` と `new` は動詞として判定する。これらは、自然言語用の品詞解析器ではそれぞれ前置詞、形容詞として判定するが、2.2.3 節で述べた通り、この `to` や `new` は動詞であるとも考えられるため、本手法ではメソッド名の先頭の `to` や `new` は動詞と判断する。

次に、事前に人手で定義しておいたメソッド名中で動詞として頻出する単語と、名詞として頻出する単語については、品詞解析器の判定に関わらず、それぞれ動詞と名詞として判定する。動詞として判定する語は

- `save`
- `hover`

である。名詞として判定する語は

- `offset`
- `string`
- `action`

である。

ここまでの処理の結果単語列の先頭にある単語が動詞と判定されておらず、単語列の末尾にある単語が過去形と判断されていないとき、自然言語用の辞書を用いて先頭語が動詞として使われる単語であるかどうかを判定し、動詞としての意味を持つ単語であった場合は動詞と判定する。実装では自然言語用の辞書として WordNet[15] を使用した。

最後に、単語列に含まれる単語のうち、動詞と前置詞以外の単語に関して結合処理を行う。動詞と前置詞以外の語が連続して出現している場合それらを結合し、一つの名詞と判定する。ただし、前置詞の中でも `of` は連体修飾語として頻出するため、`of` は名詞の一部であると考え、他の語との結合を行う。

3.3.3 戻り値の型名・引数・クラス名の品詞解析

戻り値の型名、引数、クラス名に出現する識別子の品詞解析を行う。型名の変換の例を図 7 で示す。戻り値の型名は上記の通り、パッケージ名とシグネチャとジェネクリクスの宣言

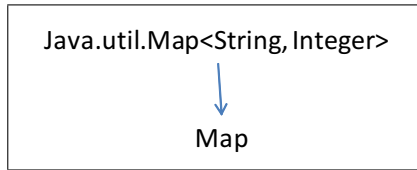


図 7: 型名の変換

解析対象のメソッドが満たさなければならない条件			
戻り値	メソッド名	引数	クラス名
名詞1	動詞1 名詞1	なし	名詞2

抽出する単語		
動詞	直接目的語	間接目的語
動詞1	名詞1	名詞2

図 8: 動詞-直接目的語-間接目的語の三つ組の抽出パターン

を取り除いた後、名詞の一単語と判定する。次に、クラス名は、クラス名全体を名詞の一単語と判定する。引数の型名は戻り値と同じくパッケージ名とシグネチャを取り除き、名詞の一単語と判定する。仮引数名は仮引数名全体を名詞の一単語と判定する。

3.3.4 抽出パターンを用いた動詞-直接目的語-間接目的語の三つ組の抽出

メソッドから動詞-直接目的語-間接目的語の三つ組を抽出するため、解析対象のメソッドがある特定の条件を満たす場合に動詞-直接目的語-間接目的語の三つ組を抽出するパターンを事前に人手で多数定義し、解析対象の各メソッドと抽出パターンとの照合を行い動詞-直接目的語-間接目的語の三つ組を抽出する。

抽出パターンは、

要素 1 この抽出パターンで動詞-直接目的語-間接目的語の三つ組の抽出を行うために解析対象のメソッドが満たしていなければならない条件

要素 2 解析対象のメソッド中のどの単語を動詞、直接目的語、間接目的語として抽出するかを定めたパターン

の2つの要素で構成される。

各々の要素について詳しく述べる。抽出パターン中の要素1は以下の条件に分解される。

戻り値が満たさなければならない条件 この条件には以下の3つの条件のいずれかが代入される。

- void型を含め、どのような戻り値でもかまわない
- void型である
- void型以外であり、戻り値の型名の品詞がこの条件で規定された品詞と一致する

メソッド名が満たさなければならない条件 この条件では、

- メソッド名の単語数
- メソッド名の各単語が持つ品詞

の2つが規定され、両方の条件を満たさなければならない

引数が満たさなければならない条件 この条件には以下2つの条件のいずれかが代入される。

- 引数がない場合を含め、どのような引数でもかまわない
- 引数がない
- 引数があり、以下の規定された条件を満たす
 - － 引数の個数
 - － 引数の各単語が持つ品詞

クラス名が満たさなければならない条件 この条件には以下の2つの条件のいずれかが代入される。

- どのようなクラス名でもかまわない
- クラス名の品詞がこの条件で規定された品詞と一致する

同じ単語がどこに出現するか この条件では、解析対象のメソッド中の特定の要素を構成する単語のうち、同じ品詞を持ち同じ綴りである単語が出現するか否か、出現する場合はどの単語どうしが同じ単語であるかが規定される。解析対象のメソッド中の特定の要素とはメソッド名、戻り値、引数、クラス名である。ただし、戻り値の条件が「void型を含め、どのような戻り値でもかまわない」のときには特定の要素に戻り値は含まれない。同様に、引数の条件が「引数がない場合を含め、どのような引数でもかまわない」のときは特定の要素に引数は含まれない。さらに、クラス名の条件が「どのようなクラス名でもかまわない」のときは特定の要素にクラス名は含まれない。

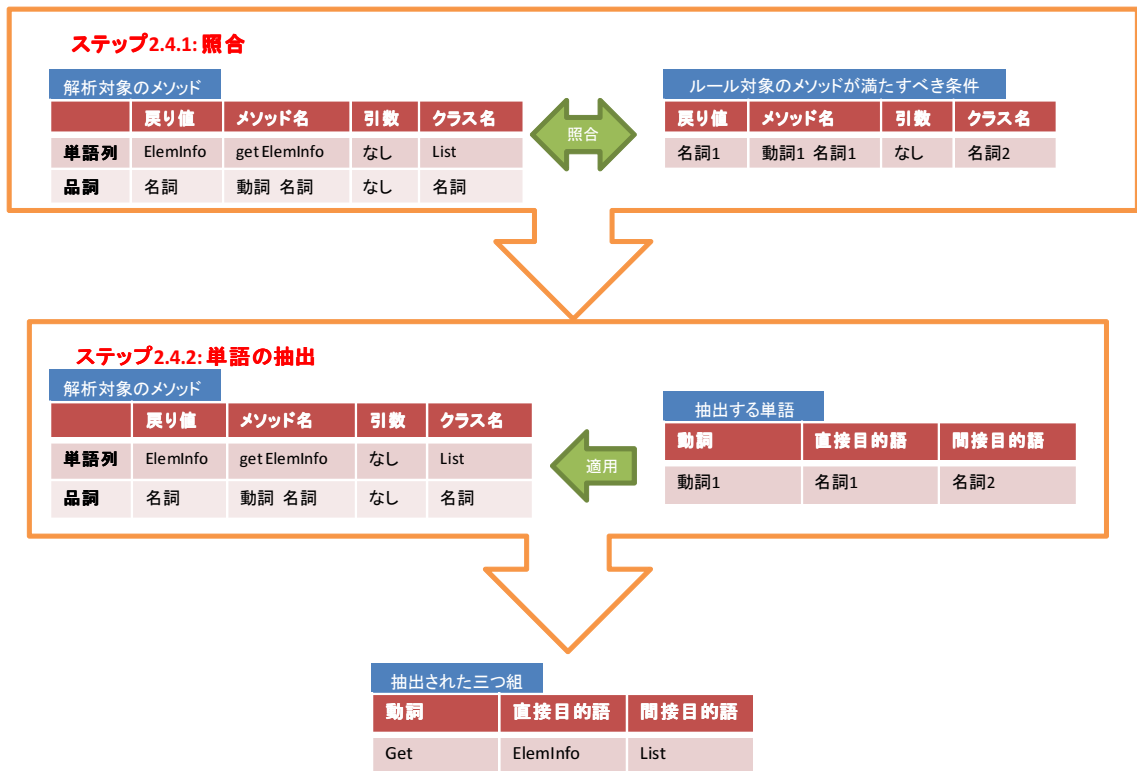


図 9: 動詞-直接目的語-間接目的語の三つ組の抽出処理

抽出パターン中の要素 2 は以下の要素で構成される。

動詞の抽出ルール メソッド中のどの要素のどの位置に出現する単語を動詞として抽出するかを定めたルール

直接目的語の抽出ルール メソッド中のどの要素のどの位置に出現する単語を直接目的語として抽出するかを定めたルール

間接目的語の抽出ルール メソッド中のどの要素のどの位置に出現する単語を間接目的語として抽出するかを定めたルール

抽出パターンの例を図 8 に示した。図中で、同じ単語の出現条件は各条件中の品詞に番号を付与して表わしており、同じ品詞で同じ番号を持つ場合同じ単語である。

次に、各メソッドから抽出パターンを用いて動詞-直接目的語-間接目的語の三つ組を抽出する処理について述べる。この処理の例を図 9 で示した。この処理は照合と単語の抽出からなる。

ステップ 2.4.1：照合 抽出パターン中の要素 1 の，解析対象のメソッドが満たさなければならない条件を，解析対象のメソッドが満たしているかどうかを判定する．満たしている場合はステップ 2.4.2 に移る．満たしていない場合はこのパターンでは動詞-直接目的語-間接目的語の三つ組の抽出は行わない．

ステップ 2.4.2：単語の抽出 抽出パターン中の要素 2 に従い，動詞，直接目的語，間接目的語となる単語を抽出する

なお，定義したパターンの一覧は表 1 節に示した．表中の*は任意の語句とマッチすることを示す値である．また表中の void は void 型を示す値である．

3.4 抽出した動詞-直接目的語-間接目的語のフィルタリング

ステップ 2 の処理で抽出された動詞-直接目的語-間接目的語の三つ組のうち入力となったソフトウェアの対象ドメインにおいて一般的な動詞-目的語関係を表わす組を辞書に収録する．提案手法では，複数のソフトウェアで出現する三つ組を，対象ドメインにおいて一般的な関係であるとみなし，辞書に収録する．実際にいくつのソフトウェアで出現した三つ組を辞書に登録するかは，人間が適宜判断し決定する．

表 1: 抽出パターン一覧

戻り値	メソッド名	引数	クラス名	動詞	直接目的語	間接目的語
*	動詞 1 名詞 1 前置詞 1 名詞 2	*	*	動詞 1	名詞 1	名詞 2
*	動詞 1 前置詞 1 名詞 1	*	名詞 2	動詞 1	名詞 2	名詞 1
*	動詞 1 名詞 1	*	名詞 2	動詞 1	名詞 1	名詞 2
*	動詞 1 前置詞 1 名詞 1	名詞 2	*	動詞 1	名詞 2	名詞 1
*	動詞 1 名詞 1 前置詞 1	名詞 2	*	動詞 1	名詞 1	名詞 2
void	動詞 1	なし	名詞 1	動詞 1	名詞 1	なし
void	動詞 1 前置詞 1 名詞 1	なし	名詞 2	動詞 1	名詞 2	名詞 1
void	動詞 1 名詞 1	名詞 1	名詞 2	動詞 1	名詞 1	名詞 2
void	動詞 1 名詞 1 前置詞 1 名詞 2	*	名詞 3	動詞 1	名詞 1	名詞 3
void	動詞 1	名詞 1	名詞 2	動詞 1	名詞 1	名詞 2
void	動詞 1 名詞 1	名詞 2	名詞 3	動詞 1	名詞 2	名詞 3
void	動詞 1 名詞 1	名詞 2	名詞 1	動詞 1	名詞 1	名詞 2
void	動詞 1 名詞 1	なし	名詞 1	動詞 1	名詞 1	なし
void	動詞 1 名詞 1	なし	名詞 1	動詞 1	名詞 1	なし
void	名詞 1 動詞 1	名詞 2	名詞 3	動詞 1	名詞 1	名詞 2
void	名詞 1 動詞 1	名詞 2	名詞 3	動詞 1	名詞 1	名詞 3
void	名詞 1 動詞 1	名詞 1	名詞 2	動詞 1	名詞 1	名詞 2
名詞 1	動詞 1 名詞 1	名詞 2	名詞 1	動詞 1	名詞 1	名詞 2
名詞 1	動詞 1 名詞 1 前置詞 1 名詞 2	名詞 2	名詞 3	動詞 1	名詞 1	名詞 2
名詞 1	動詞 1 名詞 2 前置詞 1 名詞 3	なし	名詞 4	動詞 1	名詞 2	名詞 3
名詞 1	動詞 1 前置詞 1 名詞 2	名詞 3	名詞 4	動詞 1	名詞 4	名詞 2
名詞 1	動詞 1 名詞 1	なし	名詞 2	動詞 1	名詞 1	名詞 2
名詞 1	動詞 1	名詞 2	名詞 3	動詞 1	名詞 3	名詞 2
名詞 1	動詞 1 前置詞 1	名詞 2	名詞 3	動詞 1	名詞 3	名詞 2
名詞 1	動詞 1 前置詞 1 名詞 2	*	*	動詞 1	名詞 1	名詞 2
名詞 1	動詞 1 前置詞 1 名詞 2	なし	名詞 1	動詞 1	名詞 1	名詞 2
名詞 1	動詞 1 前置詞 1	名詞 2	名詞 2	動詞 1	名詞 2	名詞 2
名詞 1	動詞 1 名詞 2	なし	名詞 1	動詞 1	名詞 2	名詞 1
名詞 1	動詞 1 名詞 1	なし	名詞 2	動詞 1	名詞 2	なし
名詞 1	動詞 1 名詞 2	なし	名詞 2	動詞 1	名詞 2	名詞 1
名詞 1	動詞 1 名詞 1	なし	名詞 1	動詞 1	名詞 1	なし

4 適用実験

特定のドメインを対象とした辞書の作成を行い，作成された動詞-目的語関係辞書に対する評価実験を行った．本節ではその内容及び結果と結果に対する考察を述べる．

4.1 実験目的

本手法により作成された，特定のドメインを対象とした動詞-目的語関係辞書に収録された動詞-直接目的語-間接目的語の三つ組が対象ドメインで一般的に見られる組であり，命名支援に役立つ組であるかどうかを検証する．具体的には辞書に収録された各三つ組に対し，以下の項目についてを検証を行う．

1. 収録された三つ組の動詞，直接目的語，間接目的語はそれぞれ適当かどうか
2. 収録された三つ組は対象ドメインでよく見られる組かどうか．また，対象ドメインを問わず Java プログラム一般で見られる組や対象ドメイン以外のドメインでよく見られる組かどうか．
3. 収録された三つ組は，対象ドメイン用の命名支援の辞書に収録するのに適当な組かどうか．また，Java プログラム一般用の命名支援の辞書や対象ドメイン以外のドメイン用の辞書に収録するのに適当な組かどうか．

4.2 実験対象

WebApplication, XML, Database, DesktopApplication の4つのドメインのソフトウェアを集め，各ドメインを対象とした辞書の作成を行った．各ドメインの辞書を作る際に解析対象としたソフトウェアの一覧を表2と表3に示す．なお，辞書には抽出された組のうち，2つ以上のソフトウェアで出現した三つ組を収録した．

また，各ドメインの辞書を作成した際に，解析対象のソフトウェアに含まれていたメソッド数の合計と，そのうちいくつのメソッドから三つ組の抽出が行えたか，そして抽出した三つ組の数はいくつであったかを示す．さらに，抽出した三つ組を，三つ組が出現したソフトウェアの数で分類した際の，三つ組の数の度数分布表を示す．Web Application の辞書作成時の情報は表4に，度数分布表は5に示した．XML の辞書作成時の情報は表6に，度数分布表は7に示した．Database の辞書作成時の情報は表8に，度数分布表は9に示した．GUI の辞書作成時の情報は表10に，度数分布表は11に示した．

表 2: ドメインと解析したソフトウェア

ドメイン	ソフトウェア
Web Application	BBS-CS8.0.3
	JForum 2.1.8
	JGossip 1.1.0.005
	mvnForum 1.2.1
	Yazd Discussion Forum Software 3.0
	Order Portal 1.2.4
	Arianne RPG 0.80
	JBoss Wiki Beta2
	JSP Wiki 2.8.3
	SnipSnap 1.0b3
XML	Castor 1.3
	DOM4J 1.6.1
	JDOM 1.1.1
	Piccolo 1.04
	Saxon-HE 9.2.0.5
	Xalan-J 2.7.1
	Xbeans 2.0.0
	Xerces-J 2.9.0
	XOM 1.2.4
	XPP3 1.1.4
	Xstream 1.3.1

表 3: ドメインと解析したソフトウェア (続き)

ドメイン	ソフトウェア
Database	Axion 1.0 Milestone 2
	Apache Derby 10.5.3
	H2 1.2.128
	HSQLDB 1.8.1.1
	Berkeley DB Java Edition 4.0.92
	Mckoi 1.0.3
	MyOODB 4.0.0
	NeoDatis 1.9.22.674
	OZONE 1.1
	tinySQL 2.26
Desktop Application	ArgoUML 0.28.1
	BlueJ 2.5.3
	Eclipse Classic 3.5.1
	jEdit 4.3.1
	NetBeans 6.8
	vuze 4.3.1.2
	LimeWire 5.4

表 4: WebApplication の辞書作成時の情報

メソッド数	74707
三つ組を抽出できたメソッドの数	67276
抽出した三つ組の数	67429

表 5: Web Application の三つ組の度数分布表

ソフトウェア数	三つ組の数
1	67147
2	258
3	18
4	4
5	2

表 6: XML の辞書作成時の情報

メソッド数	55812
三つ組を抽出できたメソッドの数	46885
抽出した三つ組の数	49926

表 7: XML の三つ組の度数分布表

ソフトウェア数	三つ組の数
1	49379
2	465
3	63
4	13
5	5
6	1

表 8: Database の辞書作成時の情報

メソッド数	74127
三つ組を抽出できたメソッドの数	60326
抽出した三つ組の数	63087

表 9: Database の三つ組の度数分布表

ソフトウェア数	三つ組の数
1	62415
2	609
3	28
4	1
5	32
6	2

表 10: GUI の辞書作成時の情報

メソッド数	298696
三つ組を抽出できたメソッドの数	247918
抽出した三つ組の数	273202

表 11: GUI の三つ組の度数分布表

ソフトウェア数	三つ組の数
1	272795
2	339
3	38
4	23
5	5
6	2

4.3 実験方法

6人の被験者に対し、抽出された動詞-直接目的語-間接目的語の各三つ組のうち(1)3つ以上のソフトウェアで出現した三つ組と(2)2つのソフトウェアでのみ出現した三つ組に対する評価をアンケート方式で行った。アンケートの質問項目を以下に示す。

設問 1 この動詞-直接目的語-間接目的語の三つ組は、対象としているドメインで一般的に見られる組だと思うか。回答は(1) そう思う (2) どちらかと言えばそう思う (3) どちらかと言えば違う (4) 違う (5) わからない、の5つから選択する。

設問 2 この動詞-直接目的語-間接目的語の三つ組は、対象ドメインを問わず、Java プログラムで一般的に見られる組だと思うか回答は、(1) そう思う (2) どちらかと言えばそう思う (3) どちらかと言えば違う (4) 違う (5) わからない、の5つから選択せよ。

設問 3 この動詞-直接目的語-間接目的語の三つ組は、対象ドメインとは別のドメインで出現する組だと思うか。そうならばドメイン名を答えよ。

設問 4 動詞、直接目的語、間接目的語それぞれに対し、判定が間違っていると思うか。違っていると思う場合はどの語が間違っていると思うか答えよ。

設問 5 この動詞-直接目的語-間接目的語の三つ組を、対象ドメインのプログラム開発を行う際の命名支援のための辞書に収録するのに適当な組だと思うか。回答は、(1) 適当な組だと思う (2) どちらかと言えば適当な組である (3) どちらかと言えば不適当な組だと思う (4) 不適当な組だと思う (5) 分からない、の5つから選択せよ。

設問 6 この動詞-直接目的語-間接目的語の三つ組を、Java プログラム一般の開発を行う際の命名支援のための辞書に収録するのに適当な組だと思うか。回答は、(1) 適当な組だと思う (2) どちらかと言えば適当な組である (3) どちらかと言えば不適当な組だと思う (4) 不適当な組だと思う (5) 分からない、の5つから選択せよ。

設問 7 この動詞-直接目的語-間接目的語の三つ組を、対象ドメインとは別のドメインのプログラム開発を行う際の命名支援のための辞書に収録し、開発者に例示してもよいと思うか。よいと思う場合はドメイン名を答えよ。

被験者は一人あたり2つのドメインの辞書に関してアンケートに答えて頂く。そして、一つの辞書あたり3人の被験者にアンケートに回答して頂く。被験者は一つの辞書に対し、(1)3つ以上のソフトウェアで出現した三つ組と(2)2つのソフトウェアでのみ出現した三つ組、それぞれ15組づつ合計30組に対し評価を行う。三つ組は辞書からランダムに抽出したものであるが、各被験者が同じ三つ組に対する評価を行わないように抽出する。ただし、Web

Application で3つ以上のソフトウェアで出現した三つ組は24組しかいないため、これについては違う被験者の間で同じ三つ組に対して評価している場合がある。

4.4 実験結果

本節では実験結果について述べる。それぞれのドメインの辞書に対するアンケートの結果を示す。また、辞書に収録するのに適当な組であると評価された三つ組と、不適当な組であると評価された三つ組の例をいくつか示す。また、アンケートの結果から、Java プログラム全般ではあまり一般的な三つ組ではないと評価されたが対象ドメインでは一般的な三つ組と評価された組の数と、その逆で対象ドメインでは一般的ではないが Java プログラム全般では一般的と評価された組の数を示し、いくつか例をあげる。Java プログラム全般ではあまり一般的な三つ組ではないと評価されたが対象ドメインでは一般的な三つ組と評価された組とは具体的には、設問2の回答がどちらかと言えば違う、または違うであり、設問1の回答がそう思う、またはどちらかと言えばそう思うになっている場合を指す。同様に対象ドメインでは一般的ではないが Java プログラム全般では一般的と評価された組とは設問1の回答がどちらかと言えば違う、または違うであり、設問2の回答がそう思う、またはどちらかと言えばそう思うであった組を指す。

表 12: 辞書に収録するのに適当と評価された組と不適当と評価された組の例

	動詞	直接目的語	間接目的語
適当な組	get	Password	SmtptAuthenticator
	destroy	Session	HttpSessionEvent
不適当な組	set	order	OrderBy
	set	Date	User

4.4.1 Web Application

本節では Web Application の辞書に対する実験の結果を示す。3つ以上のソフトウェアで出現した組に対するアンケートの回答とその回答が与えられた三つ組の数を表 13 に、2つのソフトウェアでのみ出現した組に対するアンケートの回答とその回答が与えられた三つ組の数を表 14 に示した。また、辞書に収録するのに適当であると評価された組と不適当と評価された組の例を表 12 に示した。さらに、Java プログラム全般ではあまり一般的な三つ組ではないと評価されたが対象ドメインでは一般的な三つ組と対象ドメインでは一般的ではないが Java プログラム全般では一般的と評価された組の数を表 15 に、組の例を表 16 に示した。

表 13: 3つ以上のソフトウェアに出現した三つ組に対するアンケートの回答

設問	回答	回答された組の数
設問 1	そう思う	16
	どちらかと言えばそう思う	16
	どちらかと言えば違う	1
	違う	7
	わからない	5
設問 2	そう思う	14
	どちらかと言えばそう思う	4
	どちらかと言えば違う	16
	違う	9
	わからない	2
設問 3	データベース	14
設問 4	動詞が間違っている	1
	直接目的語が間違っている	3
	間接目的語が間違っている	5
	いずれかが間違っている	8
設問 5	適当な組だと思う	17
	どちらかと言えば適当な組だと思う	13
	どちらかと言えば不適当な組だと思う	6
	不適当な組だと思う	7
	わからない	2
設問 6	適当な組だと思う	18
	どちらかと言えば適当な組だと思う	2
	どちらかと言えば不適当な組だと思う	13
	不適当な組だと思う	10
	わからない	2

表 14: 2つのソフトウェアに出現した三つ組に対するアンケートの回答

設問	回答	回答された組の数
設問 1	そう思う	16
	どちらかと言えばそう思う	20
	どちらかと言えば違う	3
	違う	2
	わからない	4
設問 2	そう思う	8
	どちらかと言えばそう思う	9
	どちらかと言えば違う	16
	違う	8
	わからない	4
設問 3	Web Application に限らず他のドメインでも一般的	2
	データベース	2
	入出力	6
設問 4	動詞が間違っている	0
	直接目的語が間違っている	3
	間接目的語が間違っている	0
	いずれかが間違っている	3
設問 5	適当な組だと思う	18
	どちらかと言えば適当な組だと思う	16
	どちらかと言えば不適当な組だと思う	4
	不適当な組だと思う	3
	わからない	4
設問 6	適当な組だと思う	13
	どちらかと言えば適当な組だと思う	11
	どちらかと言えば不適当な組だと思う	11
	不適当な組だと思う	6
	わからない	4

表 15: 一般的なドメインと三つ組の数

対象ドメインで一般的	18
Java プログラムで一般的	6

表 16: 一般的なドメインと三つ組の例

	動詞	直接目的語	間接目的語
対象ドメインで一般的	initialize	Context	Event
	delete	page	WikiEngine
Java プログラムで一般的	set	Factory	FileUploadBase
	flush	GzipResponseStream	

4.4.2 XML

本節では XML の辞書に対する実験の結果を示す。3 つ以上のソフトウェアで出現した組に対するアンケートの回答とその回答が与えられた三つ組の数を表 18 に、2 つのソフトウェアでのみ出現した組に対するアンケートの回答とその回答が与えられた三つ組の数を表 19 に示した。また、辞書に収録するのに適当であると評価された組と不適当と評価された組の例を表 17 に示した。さらに、Java プログラム全般ではあまり一般的な三つ組ではないと評価されたが対象ドメインでは一般的な三つ組と対象ドメインでは一般的ではないが Java プログラム全般では一般的と評価された組の数を表 20 に、組の例を表 21 に示した。

表 17: 辞書に収録するのに適当と評価された組と不適当と評価された組の例

	動詞	直接目的語	間接目的語
適当な組	end	Name	XmlWriter
	declare	prefix	NamespaceSupport
不適当な組	To	String	Entry
	set	Result	TransformerHandlerImpl

表 18: 3つ以上のソフトウェアに出現した三つ組に対するアンケートの回答

設問	回答	回答された組の数
設問 1	そう思う	24
	どちらかと言えばそう思う	10
	どちらかと言えば違う	4
	違う	3
	わからない	4
設問 2	そう思う	11
	どちらかと言えばそう思う	5
	どちらかと言えば違う	7
	違う	21
	わからない	1
設問 3	GUI	1
	データ解析	1
設問 4	動詞が間違っている	4
	直接目的語が間違っている	3
	間接目的語が間違っている	0
	いずれかが間違っている	6
設問 5	適当な組だと思う	21
	どちらかと言えば適当な組だと思う	7
	どちらかと言えば不適当な組だと思う	7
	不適当な組だと思う	6
	わからない	4
設問 6	適当な組だと思う	11
	どちらかと言えば適当な組だと思う	5
	どちらかと言えば不適当な組だと思う	8
	不適当な組だと思う	20
	わからない	1
設問 7	GUI	1

表 19: 2つのソフトウェアに出現した三つ組に対するアンケートの回答

設問	回答	回答された組の数
設問 1	そう思う	20
	どちらかと言えばそう思う	8
	どちらかと言えば違う	1
	違う	4
	わからない	12
設問 2	そう思う	4
	どちらかと言えばそう思う	6
	どちらかと言えば違う	10
	違う	21
	わからない	4
設問 3	構文解析器	1
	リソース管理	1
	木構造	1
	グラフ解析	1
	データ解析	1
設問 4	動詞が間違っている	1
	直接目的語が間違っている	4
	間接目的語が間違っている	1
	いずれかが間違っている	6
設問 5	適当な組だと思う	12
	どちらかと言えば適当な組だと思う	8
	どちらかと言えば不適当な組だと思う	3
	不適当な組だと思う	10
	わからない	12
設問 6	適当な組だと思う	3
	どちらかと言えば適当な組だと思う	8
	どちらかと言えば不適当な組だと思う	4
	不適当な組だと思う	26
	わからない	4
設問 7	構文解析器	1
	リソース管理	1
	木構造	1
	グラフ解析	1
	データ解析	1

表 20: 一般的なドメインと三つ組の数

対象ドメインで一般的	41
Java プログラムで一般的	5

表 21: 一般的なドメインと三つ組の例

	動詞	直接目的語	間接目的語
対象ドメインで一般的	end	Document	XmlWriter
	get	Document	DomAdapter
Java プログラムで一般的	peek	IntStack	
	To	String	Type

4.4.3 Database

本節では Database の辞書に対する実験の結果を示す. 3つ以上のソフトウェアで出現した組に対するアンケートの回答とその回答が与えられた三つ組の数を表 23 に, 2つのソフトウェアでのみ出現した組に対するアンケートの回答とその回答が与えられた三つ組の数を表 24 に示した. また, 辞書に収録するのに適当であると評価された組と不適当と評価された組の例を表 22 に示した. さらに, Java プログラム全般ではあまり一般的な三つ組ではないと評価されたが対象ドメインでは一般的な三つ組と対象ドメインでは一般的ではないが Java プログラム全般では一般的と評価された組の数を表 25 に, 組の例を表 26 に示した.

表 22: 辞書に収録するのに適当と評価された組と不適当と評価された組の例

	動詞	直接目的語	間接目的語
適当な組	add	Index	Table
	be	Admin	User
不適当な組	release	Mouse	MouseEvent
	add	All	HashSet

表 23: 3つ以上のソフトウェアに出現した三つ組に対するアンケートの回答

設問	回答	回答された組の数
設問 1	そう思う	16
	どちらかと言えばそう思う	16
	どちらかと言えば違う	1
	違う	7
	わからない	5
設問 2	そう思う	14
	どちらかと言えばそう思う	4
	どちらかと言えば違う	16
	違う	9
	わからない	2
設問 3	GUI	4
	Web Application	1
設問 4	動詞が間違っている	1
	直接目的語が間違っている	3
	間接目的語が間違っている	5
	いずれかが間違っている	8
設問 5	適当な組だと思う	17
	どちらかと言えば適当な組だと思う	13
	どちらかと言えば不適当な組だと思う	6
	不適当な組だと思う	7
	わからない	2
設問 6	適当な組だと思う	18
	どちらかと言えば適当な組だと思う	2
	どちらかと言えば不適当な組だと思う	13
	不適当な組だと思う	10
	わからない	2
設問 7	GUI	4
	Web Application	1

表 24: 2つのソフトウェアに出現した三つ組に対するアンケートの回答

設問	回答	回答された組の数
設問 1	そう思う	16
	どちらかと言えばそう思う	20
	どちらかと言えば違う	3
	違う	2
	わからない	4
設問 2	そう思う	8
	どちらかと言えばそう思う	9
	どちらかと言えば違う	16
	違う	8
	わからない	4
設問 3	GUI	1
設問 4	動詞が間違っている	0
	直接目的語が間違っている	3
	間接目的語が間違っている	0
	いずれかが間違っている	3
設問 5	適当な組だと思う	18
	どちらかと言えば適当な組だと思う	16
	どちらかと言えば不適当な組だと思う	4
	不適当な組だと思う	3
	わからない	4
設問 6	適当な組だと思う	13
	どちらかと言えば適当な組だと思う	11
	どちらかと言えば不適当な組だと思う	11
	不適当な組だと思う	6
	わからない	4
設問 7	GUI	1

表 25: 一般的なドメインと三つ組の数

対象ドメインで一般的	38
Java プログラムで一般的	4

表 26: 一般的なドメインと三つ組の例

	動詞	直接目的語	間接目的語
対象ドメインで一般的	support	Catalogs	ProcedureCalls
	get	Index	Column
Java プログラムで一般的	click	Mouse	MouseEvent
	remove	HashMap	Key

4.4.4 GUI

本節では GUI の辞書に対する実験の結果を示す。3 つ以上のソフトウェアで出現した組に対するアンケートの回答とその回答が与えられた三つ組の数を表 28 に、2 つのソフトウェアでのみ出現した組に対するアンケートの回答とその回答が与えられた三つ組の数を表 29 に示した。また、辞書に収録するのに適当であると評価された組と不適当と評価された組の例を表 27 に示した。さらに、Java プログラム全般ではあまり一般的な三つ組ではないと評価されたが対象ドメインでは一般的な三つ組と対象ドメインでは一般的ではないが Java プログラム全般では一般的と評価された組の数を表 30 に、組の例を表 31 に示した。

表 27: 辞書に収録するのに適当と評価された組と不適当と評価された組の例

	動詞	直接目的語	間接目的語
適当な組	close	Connection	
	click	Mouse	MouseEvent
不適当な組	refresh	Diffence	Sleak
	class	Class	Name

表 28: 3つ以上のソフトウェアに出現した三つ組に対するアンケートの回答

設問	回答	回答された組の数
設問 1	そう思う	23
	どちらかと言えばそう思う	12
	どちらかと言えば違う	5
	違う	3
	わからない	2
設問 2	そう思う	15
	どちらかと言えばそう思う	24
	どちらかと言えば違う	3
	違う	1
	わからない	2
設問 3	文字列処理	1
	GUIに限らず他のドメインでも一般的にありそう	2
	DB	1
	ネットワーク関係	1
設問 4	動詞が間違っている	6
	直接目的語が間違っている	0
	間接目的語が間違っている	0
	いずれかが間違っている	6
設問 5	適当な組だと思う	13
	どちらかと言えば適当な組だと思う	16
	どちらかと言えば不適当な組だと思う	5
	不適当な組だと思う	8
	わからない	3
設問 6	適当な組だと思う	13
	どちらかと言えば適当な組だと思う	15
	どちらかと言えば不適当な組だと思う	7
	不適当な組だと思う	7
	わからない	3

表 29: 2つのソフトウェアに出現した三つ組に対するアンケートの回答

設問	回答	回答された組の数
設問 1	そう思う	19
	どちらかと言えばそう思う	14
	どちらかと言えば違う	4
	違う	3
	わからない	5
設問 2	そう思う	17
	どちらかと言えばそう思う	13
	どちらかと言えば違う	6
	違う	5
	わからない	4
設問 3	テストケース	1
	アーカイバ	1
	GUIに限らず一般的にありそう	2
設問 4	動詞が間違っている	2
	直接目的語が間違っている	1
	間接目的語が間違っている	0
	いずれかが間違っている	3
設問 5	適当な組だと思う	15
	どちらかと言えば適当な組だと思う	14
	どちらかと言えば不適当な組だと思う	8
	不適当な組だと思う	3
	わからない	5
設問 6	適当な組だと思う	16
	どちらかと言えば適当な組だと思う	11
	どちらかと言えば不適当な組だと思う	8
	不適当な組だと思う	6
	わからない	4

表 30: 一般的なドメインと三つ組の数

対象ドメインで一般的	13
Java プログラムで一般的	13

表 31: 一般的なドメインと三つ組の例

	動詞	直接目的語	間接目的語
対象ドメインで一般的	scroll	Rect	Visible
	dispose	Widget	DisposeEvent
Java プログラムで一般的	get	Descriptor	Type
	open	Connection	Handler

4.5 妥当性

回答者に回答して頂いた三つ組は，辞書中に収録された三つ組からランダムに抽出した。ただし，違う被験者が同じ三つ組に対して回答しないように抽出を行った。

被験者は私の所属する井上研究室に所属する学生に協力して頂いた。そのため被験者のランダムな抽出は行えていない。しかし，被験者はみな Java での開発経験があり，各被験者の回答対象のドメインは，そのドメインの知識がある被験者に回答して頂けるように選択した。

実験結果を，今回作成していないドメインの辞書に対して適用するためには，辞書を作成するための解析対象のソフトウェアが扱うドメインを十分に絞り，多くのソフトウェアを解析対象とする必要がある。

4.6 辞書に収録するのに適当でない理由

いくつかの三つ組は対象ドメインや Java で一般的に見られる関係であるが，辞書に収録するべきではないという評価がなされた。この理由について，被験者に対し追跡調査を行った。以下被験者が挙げた理由を示す。

- 動詞，直接目的語，間接目的語の判定が間違っている
- 一部直接目的語や間接目的語に何らかの語の省略語と思われる語が出現していた。具体的には Fe や Ae などが挙げられる。それらは辞書に収録しても命名支援に役立たないと判断した

- あまりにも一般的すぎる関係だった
- 特定のライブラリに依存して出現する関係であり，ドメインの辞書に入れるには不適當と判断した

4.7 その他被験者から頂いたコメント

アンケートの際に設問以外にもいくつかコメントを頂いた。そのうち特に参考になったコメントを挙げる。

- 同じ意味を表現する際に，いくつかある同義語のうちどの語を用いるかは表記揺れを無くすために重要であり，このような辞書が参考になるかもしれない
- 変数等に使われている省略名と完全名との対応関係を示して欲しい
- 動詞とその目的語だけ示されても，ドメインの知識が無い人間には意味がわからないのではないか

4.8 考察

本実験では，ドメイン特有の関係のみを抽出した辞書の作成には成功していない。この原因として以下の2つが考えられる。

1. 辞書を作成する際に解析対象として入力したソフトウェアの数が少なすぎたため，ドメイン特有の関係を一般的な関係と判定して辞書に収録することができなかった
2. 一般にソフトウェアは複数のドメインを扱うことが多く，単純に三つ組が出現したソフトウェアの数でフィルタリングを行っても，対象ドメインのみに含まれている関係だけを残すことができない

前者は解析対象として入力するソフトウェアを増やすことで解決できる。後者は，まず様々な種類のソフトウェアを集め辞書を作成し，プログラム全般における一般的な関係を収録した辞書を作成し，その後対象ドメインを扱うソフトウェアから作成した辞書からプログラム全般で出現した一般的な関係を取り除くという方法で解決を考えている。

5 関連研究

ソースコードを静的解析し、動詞や目的語を抽出していくつかの問題に適用する手法が提案されている。本節ではそれらの関連研究について述べる。

Shepherd や Fry ら [2, 12] はオブジェクト指向プログラムのソースコード中のメソッドやコメントから動詞と直接目的語の組を抽出し、Feature Location 問題や Aspect Mining に適用する手法を提案した。対して本研究では、動詞と直接目的語と間接目的語の三つ組を抽出している。さらに、複数のソフトウェアのソースコードから取り出した三つ組のうち、複数のソフトウェアで出現した三つ組を辞書に収録している。

また、Hill ら [3] はオブジェクト指向プログラムのソースコード中のメソッドから動詞と直接目的語と間接目的語の三つ組を抽出し、Feature Location 問題や Aspect Mining に適用している。対して本研究では、複数のソフトウェアのソースコードから抽出した三つ組のうち、複数のソフトウェアで出現した三つ組を辞書に収録している。

Høst [4, 5] らはオブジェクト指向プログラムのソースコード中のメソッドに出現する動詞に着目し、動詞の辞書の作成を行った。また、後にメソッド名の命名パターンとメソッドのメトリクスとの対応関係を収集し、メソッドの命名規約違反を検出する手法を提案している。対して本研究では、動詞と直接目的語と間接目的語の三つ組を抽出対象としている。

6 まとめと今後の課題

本研究では、オブジェクト指向プログラムにおける識別子の命名を支援するため、ソースコード中に出現する動詞-目的語関係を、動詞-直接目的語-間接目的語の三つ組として抽出し、収録した辞書の構築を行った。

今後の課題としては、構築した辞書を用いて命名支援を行うプログラムの作成が必要である。命名しようとしているメソッドのクラスの名前や、フィールド変数の名前など、既に記述されている識別子を基に動詞-目的語関係の推薦を行うプログラムを Eclipse Plugin として実装することを検討している。さらに、構築した辞書中の動詞-目的語関係を視覚化し、保守作業者に提示することによるプログラム理解支援も期待できる。

謝辞

本研究において、常に適切な御指導および御助言を賜りました大阪大学大学院情報科学研究科コンピュータサイエンス専攻 井上 克郎 教授に心より深く感謝いたします。

本研究において、適切な御指導および御助言を賜りました大阪大学大学院情報科学研究科コンピュータサイエンス専攻 松下 誠 准教授に深く感謝いたします。

本研究において、逐次適切な御指導および御助言を頂きました大阪大学大学院情報科学研究科コンピュータサイエンス専攻 石尾 隆 助教に深く感謝いたします。

本研究において、常時適切な御指導および御助言を頂きました大阪大学大学院情報科学研究科コンピュータサイエンス専攻 早瀬 康裕 特任助教に深く感謝いたします。

最後に、その他様々な御指導、御助言等を頂いた大阪大学大学院情報科学研究科コンピュータサイエンス専攻井上研究室の皆様に深く感謝いたします。

参考文献

- [1] Java Platform SE6API 仕様. <http://java.sun.com/javase/ja/6/docs/ja/api>.
- [2] Z. P. Fry, D. Shepherd, E. Hill, L. Pollock, and K. Vijay-Shanker. 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.
- [3] Emily Hill, Lori Pollock, and K. Vijay-Shanker. Automatically capturing source code context of nl-queries for software maintenance and reuse. In *ICSE '09: Proceedings of the 2009 IEEE 31st International Conference on Software Engineering*, pp. 232–242, Washington, DC, USA, 2009. IEEE Computer Society.
- [4] Einar W. Høst and Bjarte M. Østvold. The programmer’s lexicon, volume i: The verbs. In *SCAM '07: Proceedings of the Seventh IEEE International Working Conference on Source Code Analysis and Manipulation*, pp. 193–202, Washington, DC, USA, 2007. IEEE Computer Society.
- [5] Einar W. Høst and Bjarte M. Østvold. Debugging method names. In *Genoa: Proceedings of the 23rd European Conference on ECOOP 2009 — Object-Oriented Programming*, pp. 294–317, Berlin, Heidelberg, 2009. Springer-Verlag.
- [6] Thomas D. LaToza, David Garlan, James D. Herbsleb, and Brad A. Myers. Program comprehension as fact finding. In *ESEC-FSE '07: Proceedings of the the 6th joint meeting of the European software engineering conference and the ACM SIGSOFT symposium on The foundations of software engineering*, pp. 361–370, New York, NY, USA, 2007. ACM.
- [7] Thomas D. LaToza, Gina Venolia, and Robert DeLine. Maintaining mental models: a study of developer work habits. In *ICSE '06: Proceedings of the 28th international conference on Software engineering*, pp. 492–501, New York, NY, USA, 2006. ACM.
- [8] Dawn Lawrie, Christopher Morrell, Henry Feild, and David Binkley. What’s in a name? a study of identifiers. In *ICPC '06: Proceedings of the 14th IEEE International Conference on Program Comprehension*, pp. 3–12, Washington, DC, USA, 2006. IEEE Computer Society.
- [9] MASU. <http://sourceforge.net/projects/masu/>.

- [10] OpenNLP. <http://opennlp.sourceforge.net/>.
- [11] Nancy Pennington. Comprehension strategies in programming. pp. 100–113, 1987.
- [12] David Shepherd, Lori Pollock, and K. Vijay-Shanker. Towards supporting on-demand virtual remodularization using program graphs. In *AOSD '06: Proceedings of the 5th international conference on Aspect-oriented software development*, pp. 3–14, New York, NY, USA, 2006. ACM.
- [13] The Java Tutorials. <http://java.sun.com/docs/books/tutorial/java/java00/index.html>.
- [14] A. von Mayrhauser and A. M. Vans. Identification of dynamic comprehension processes during large scale maintenance. *IEEE Trans. Softw. Eng.*, Vol. 22, No. 6, pp. 424–437, 1996.
- [15] WordNet. <http://wordnet.princeton.edu/>.
- [16] 早瀬康裕, 市井誠, 井上克郎. ソフトウェア理解支援を目的とした辞書の作成法. 情報処理学会シンポジウム論文集, No. 3, pp. 33–34, 2008.