メソッドに用いられる動詞-目的語関係を収録した辞書構築手法の提案

鹿島　悠[†1]　早瀬康裕[†2]　真鍋雄貴[†1]
松下　誠[†1]　井上克郎[†1]

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を対応付ける重要な手掛りである．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．そこで本研究では，開発者による識別子への適切な命名を支援するためにオブジェクト指向プログラムで重要な役割を持つ動詞-目的語関係をソースコードから抽出し，その関係を収録した辞書の構築を行った．評価実験として 4 つの専門領域の辞書を構築し，辞書に収録された動詞-目的語関係が適当であるかどう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妥当であることを確認した．

An Approach for Building a Dictionary of Verb-Object Relations used in Methods

Yu Kashima,[†1] Yasuhiro Hayase,[†2] Yuki Manabe,[†1]
Makoto Matsushita[†1] and Katsuro Inoue[†1]

Software developers often use identifiers to link program elements with domain knowledge. If identifiers in a program have inappropriate name, developers consume much time to comprehend the program. This paper proposes a method to extract and gather verb-object (V-O) relations from source code written in object-oriented programming language. For each several domains, dictionaries consist of the V-O relations were built and evaluated by software developers. Evaluation result confirms that the relations in the dictionaries are adequate.

1. はじめに

ソフトウェア保守で大きな時間を消費するプログラム理解[2] では，識別子の名前が重要な役割を果たす[1]．作業者はプログラム理解を行う際に，関数や変数などのプログラム要素の役割を識別子から推測し，アプリケーションドメインの知識と対応付けることで，プログラムを読み進めて行く．

識別子は複数の単語から構成される場合も多く，また，プログラムでは複数の識別子を共に用いること動作を表すため，識別子に用いられる単語の間には様々な関係がある．メソッドには，メソッドの振舞いを表す動詞節を識別子として用いるのが一般的である．このとき，メソッド名には，動詞や目的語，前置詞などが表れる．さらに，メソッドを定義する際には，所属するクラスや仮引数の識別子を共に用いるし，メソッドを呼び出す際にはレシーバオブジェクトや実引数やなどの識別子が共に用いられる．これらの識別子に表れる単語の間にも，目的語などの役割がある．

プログラム中の識別子が適切な名前を持っていない場合には，作業者はプログラム要素の役割の推測や，ドメインの知識との対応付けが上手く出来ない．例えば，識別子の名前に意味の無い通し番号や，頭文字のみからなる略語が用いられている場合には，役割を正しく表す名前が用いられている場合に比べて，プログラム理解に長い時間を要してしまうことが知られている[7]．そのため，プログラムを作成したり変更したりする場合には，プログラムに表れる識別子に対して，その役割を正確に表す名前を付けることが望ましいと考えられる．

しかし，良い名前を付ける作業には様々な知識と経験が必要なため，全てのソフトウェア開発者が良い命名を行えるわけではない．プログラミング言語や，所属する開発組織，アプリケーションドメイン毎に，様々な単語やその組み合わせ方 (命名規則) が存在し，開発者はそれらを全て身につける必要がある．これらの単語や命名規則は，文書化されていない場合も多く，そのような場合には実例を多く見ることが唯一の学習方法となる．

そこで，我々の研究グループでは，良い命名の例を，蓄積されたソースコードから抽出して蓄積する研究を行なっている．過去に行なった研究では，名詞の上位下位関係と，名詞に対する修飾語の関係を抽出した辞書を作成した[11]．この名詞を中心とした辞書は，クラス

†1 大阪大学
Osaka University
†2 東洋大学
Toyo University

名や変数名などの命名支援に役立つと考えられる．

本研究では，メソッドに出現する動詞と目的語の関係に着目し，ソースコードから抽出した，動詞-目的語関係の辞書を作成する手法を提案する．具体的には，自然言語解析の技術と，我々が定義したパターンを用いることで，メソッド名から動詞を取り出し，メソッド名や引数，メソッドの所属するクラス名などから目的語を抽出する．この手順を，ドメイン毎に分類したソースコード集合に対して適用することで，ドメイン特有の動詞-目的語関係の辞書を作成する．

提案手法を 4 つのドメインに分類したソースコード集合に対して適用し，作成した辞書を評価した．その結果，作成した辞書には，ドメイン特有の動詞-目的語関係が収録されているものの，同じプログラミング言語を使用したプログラムに一般的に表れる関係も多く含まれていることが分かった．

以下，2 章ではオブジェクト指向プログラムに出現する動詞-目的語関係について述べる．3 章では，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について述べる．4 章では，辞書を作成する手法について述べる．5 章では，評価実験について述べる．6 章では，本研究の関連研究について述べる．7 章では，今後の課題について述べる．

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

本節では 1 節で述べた，オブジェクト指向プログラムに出現する動詞-目的語の関係について説明する．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．これは，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るため，動詞と目的語の関係は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，メソッド名中の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る．Fry[3] 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．

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[10] 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．動詞 bind とその目的語となっている Socket という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．

3. 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，動詞-目的語関係の抽出の際に前提とする，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．

3.1 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，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場合がある[8]．

複合語の代表的な表記法には，CamelCase と snake_case がある．CamelCase 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．

3.2 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名では，動詞または動詞句が先頭にあり，その後に名詞や形容詞が続く場合が多い[8]．また，Java 標準 API[10] の java.awt.event.ActionLister 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がある．

3.2.1 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本来あるはずの動詞が省略されている，もしくは 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．本研究の提案手法では，to や new はとして使われていると考えている．

4. 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，特定のドメインを扱うソフトウェアのソースコード集合である．出力は特定のドメインで出現する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．ただし，間接目的語は空となる場合がある．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．本手法の概要を**図 1** に示す．

本手法は以下の 3 つのステップからなる．

ステップ 1：**入力されたソースコード集合の解析**　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：**動詞-目的語関係の抽出**　ソースコード集合から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：**抽出した関係のフィルタリング**　抽出した動詞-直接目的語-間接目的語の三つ組のうち，一定数以上のソフトウェアに出現する三つ組を辞書に収録する

以降，各ステップを説明する．

4.1　入力されたソースコード集合の解析

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，ソースコード集合に含まれる各メソッドから，戻り値の型名，メソッド名，引数，メソッドが定義されているクラスの情報を抽出する．

4.2　動詞-目的語関係の抽出

ソースコード集合の解析により抽出されたメソッドの情報を基に動詞-目的語関係を表す動詞-直接目的語-間接目的語の三つ組の抽出を行う．本ステップではメソッドから抽出した情報に対して品詞情報を付与したメソッド情報と，特定の品詞情報を持つメソッド情報から動詞-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とのパターンマッチングにより三つ組を抽出する．なお，抽出パターンは事前に人手で多数定義しておく．

動詞-目的語関係の抽出処理は以下のステップからなる．

メソッド情報の獲得　各メソッドから抽出した情報に対し，品詞情報を付与し，メソッド情報を獲得する．

パターンマッチングによる動詞-目的語関係の抽出　メソッド情報と抽出パターンとの間でパターンマッチングを行い，動詞-直接目的語-間接目的語の三つ組を抽出する

以降，メソッド情報とその獲得，抽出パターン，パターンマッチングによる動詞-目的語関係の抽出を説明する．

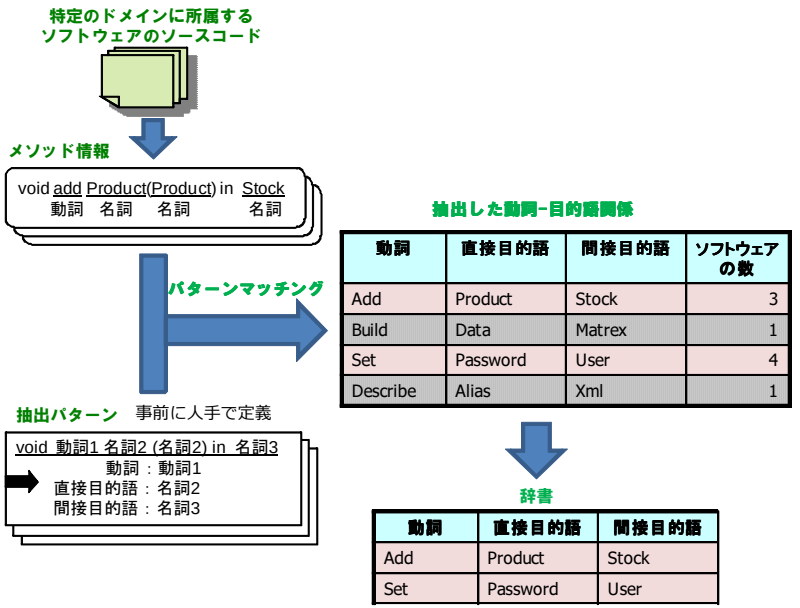

図 1　提案手法の概要

4.2.1　メソッド情報の獲得

各メソッドの，メソッドを定義しているクラスの名前，メソッド名，戻り値の型名，引数を変換しメソッド情報を獲得する．メソッド情報とは，戻り値の型名，メソッド名，引数，クラス名に含まれる各単語に対して品詞情報を与えた単語の列 4 組である．品詞情報とは，一つの単語に対し，その単語の品詞を示した情報である．以降，各メソッドの，メソッドを定義しているクラスの名前，メソッド名，戻り値の型名，引数を変換し，メソッド情報を獲得する方法について説明する．

戻り値の型名は名詞の一単語とみなし，名詞であるという品詞情報を与える．ただし，戻り値の型が void 型の場合は名詞とは見なさず，品詞情報を与えない．

引数は各々の型名や引数名を一つの単語とみなし，それらはすべて名詞と判定する．引数は 0 個以上の名詞の列と判定し，各単語に名詞であるという品詞情報を与える．

クラス名は名詞の一単語とみなし，名詞であるという品詞情報を与える．

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．そして分解により得た単語列に対し，品詞解析を行う．そして品詞解析の後，動詞と前置詞と判定された単語以外の単語に関して結合処理を行う．具体的には，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，名詞の一単語とする．そして各単語の品詞情報を与える．結合処理を行った単語は名詞であるという品詞情報を与え，それ以外の単語には品詞解析の結果得られた品詞の情報を与える．

4.2.2 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

三つ組の抽出はパターンマッチングにより行う．パターンマッチングでは，入力としてメソッド情報を与え，メソッド情報と抽出パターンを照合し，動詞-直接目的語-間接目的語の三つ組を抽出する．

抽出パターンは以下の 2 つから構成される．

- 条件 5 組
- 単語を指定する情報

前者は抽出パターンをメソッド情報に対してパターンマッチングした際に，動詞-直接目的語-間接目的語の三つ組の抽出を行う条件を，戻り値，メソッド名，引数，クラス名のそれぞれに対して指定した条件 4 組と，メソッド情報中の単語のうち同じ単語が出現する位置を指定した条件 1 組である．後者は条件 4 組が全て満たされた時にメソッド情報に出現する単語のうち，動詞，直接目的語，間接目的語として抽出する単語を指定した情報である．

抽出パターン中の条件 5 組のうち，戻り値に対する条件には，void であるか，名詞一つであるか，任意の語句の出現を許すワイルドカードであるか，のいずれかの条件が指定される．メソッド名に対する条件には，品詞の列が指定される．引数に対する条件には，名詞の列か，ワイルドカードかのいずれかの条件が指定される．クラス名に対する条件には，名詞一つであるか，ワイルドカードであるかのいずれかの条件が指定される．さらに，メソッド情報中に含まれる同じ単語が出現する位置の条件が指定される．

パターンマッチングは以下の手順からなる．

ステップ 1 メソッド情報が，抽出パターン中の条件 5 組を全て満たしているか検査する

ステップ 2 条件 5 組を全て満たしていた場合，抽出パターン中の単語を指定する情報に従い単語を抽出する

ステップ 1 では，メソッド情報の戻り値，メソッド名，引数，クラス名が抽出パターン中の条件 5 組の戻り値に対する条件，メソッド名に対する条件，引数に対する条件，クラス名

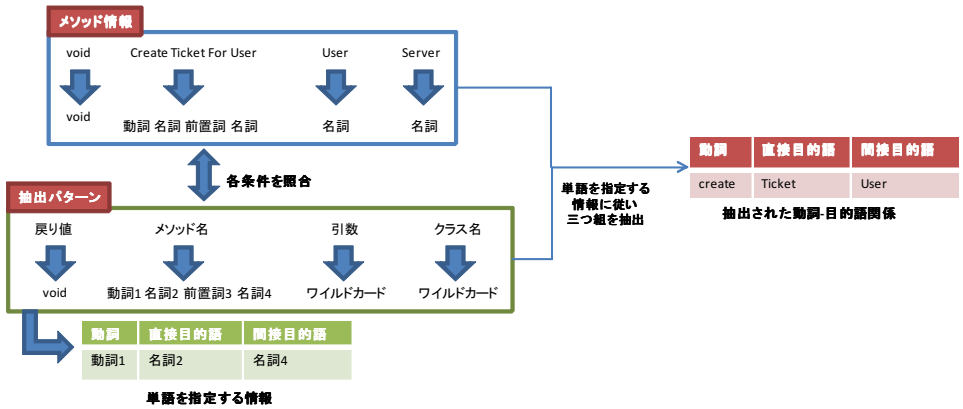

図 2 パターンマッチング

に対する条件，同じ単語が出現する位置の条件を全て満たしているか検査する．戻り値に対する条件，引数に対する条件，クラス名に対する条件で，ワイルドカードが指定された場合はメソッド情報の対応する箇所に任意の語句の出現を許す．戻り値に対する条件で，void が指定された場合は，戻り値の型が void である場合のみを許す．戻り値に対する条件，メソッド名に対する条件，引数に対する条件，クラス名に対する条件で品詞が指定された場合は，メソッド情報の対応する箇所の単語に与えられた品詞情報と条件中の品詞が一致する場合のみを許す．

ステップ 1 で条件 5 組をメソッド情報が全て満たしていた場合ステップ 2 に移る．ステップ 2 では，抽出パターン中の単語の指定に従い，動詞-直接目的語-間接目的語の三つ組の抽出を行う．

以上説明したパターンマッチングを，事前に定義した全ての抽出パターンと獲得されたメソッド情報との間で行う．メソッド情報に複数のパターンがマッチした場合，一つのメソッドから複数の三つ組が抽出される．

4.3 抽出した動詞-直接目的語-間接目的語のフィルタリング

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よく見られる動詞- 目的語関係を表わす組を辞書に収録する．提案手法では，人間が指定した数以上のソフトウェアで出現する三つ組を辞書に収録する．

5. 評価実験

提案手法により作成される辞書の妥当性を評価するため，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．

5.1 実験対象

抽出パターンを事前に 31 個定義し，WebApplication，XML，Database，DesktopApplication の 4 つのドメインを実験対象とし，各ドメインに属するソフトウェアを収集した．各ドメインのソフトウェアの一覧を**表 1** に示す．以降の表において，ドメイン WebApplication，XML，Database，GUI をそれぞれ，Web，XML，DB，GUI と略記する．

表 1 ドメインと解析したソフトウェア

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	
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		
GUI		
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 2 各ドメインの辞書作成に用いたメソッド数と三つ組の数

	$\|N\|$	$\|N_t\|$	$\|T\|$
Web	74707	67276	67429
XML	55812	46885	49926
DB	74127	60326	63087
GUI	298696	247918	273202

5.2 実験方法

初めに，本手法を用いて実験対象の各ドメインを対象とした動詞-目的語関係辞書を作成する．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，そのうち三つ組の抽出が行えたメソッドの数と，抽出した三つ組の数を**表 2** に示す．表 2 において，N をメソッドの集合，N_t を三つ組を抽出できたメソッドの集合，T を三つ組の集合とする．各ドメインにおいて，三つ組を抽出できたメソッドの全メソッドに対する割合は Web Application で 90，XML で 84%，Database で 81%，GUI では 83%であった．いくつかのメソッドでは，一つのメソッドから複数の三つ組を抽出したため，抽出を行ったメソッドの数よりも抽出した三つ組の数の方が大きくなっている．

抽出した三つ組を同じ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**表 3** に示す．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．各辞書に収録された三つ組の数は，WebApplication の辞書が 282 組，XML の辞書が 547 組，Database の辞書が 672 組，GUI の辞書が 407 組である．

次に，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を以下の観点で評価する．

(1) 対象ドメインでよく見られる組か．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．
(2) 動詞，直接目的語，間接目的語はそれぞれ適当か
(3) 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．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．

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 3 つ以上のソフトウェアで出現した三つ組と 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．被験者はみな Java での開発経験がある．また，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が回答するように選択した．

表 3 各三つ組が出現したアプリケーション数の度数分布表

	1	**2**	**3**	**4**	**5**	**6**
Web	67147	258	18	4	2	0
XML	49379	465	63	13	5	1
DB	62415	609	28	1	32	2
GUI	272795	339	38	23	5	2

実験で使用したアンケートの内容を下記に示す．実験目的の観点 1 を評価するために以下の 3 つの設問を用意した．

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 (A) そう思う，(B) どちらかと言えばそう思う，(C) どちらかと言えば違う，(D) 違う，(E) 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 (A) そう思う，(B) どちらかと言えばそう思う，(C) どちらかと言えば違う，(D) 違う，(E) わからないの 5 つから選択する．

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．

実験目的の観点 2 を評価するために次の設問を用意した．

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．

実験目的の観点 3 を評価するために以下の 3 つの設問を用意した．

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，(A) 適当な組だと思う，(B) どちらかと言えば適当な組である，(C) どちらかと言えば不適当な組だと思う，(D) 不適当な組だと思う，(E) 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，A) 適当な組だと思う，(B) どちらかと言えば適当な組である，(C) どちらかと言えば不適当な組だと思う，(D) 不適当な組だと思う，(E) 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回答する．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する．被験者は一つの辞書に対し，3 つ以上のソフトウェアで出現した三つ組と 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．三つ組は提案手法を用いて作成した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，ドメイン 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．

5.3 実験結果

観点 1 についてのアンケート結果を以下に示す．設問 1 の回答，設問 2 の回答を**表 4** に示す．観点 1 の各設問に対し (A) そう思う，(B) どちらかと言えばそう思うと回答した場合を肯定的評価と見なした場合，設問 1 の結果より，対象ドメインで見られると肯定的評価を与えられた各辞書の三つ組の割合は，Web Application で 62%，XML で 68%，Database で 75%，GUI で 75%であった．また，設問 2 の結果より各辞書の三つ組のうち Java プログラムで一般的に見られると肯定的評価を与えられた組の割合は Web Application で 50%，XML で 48%，Database で 38%，GUI で 76%であった．

次に，設問 3 の回答について以下に記述する．Web Application の辞書ではデータベースのドメインに属すると判断された組が 14 組，Web Application に限らず一般的に出現すると判断された組が 2 組，入出力のドメインに属すると判断されたが 6 組であった．XML の辞書ではデータ解析のドメインに属すると判断された組が 2 組，GUI，構文解析器，リソース管理，木構造，グラフ解析のドメインに属すると判断された組がそれぞれ 1 組ずつあった．Database の辞書では，GUI のドメインに属すると判断された組が 5 組，Web Application のドメインに属すると判断された組が 1 組であった．GUI の辞書では，GUI に限らず他のドメインでも一般的であると判断された組が 4 組，データベース，ネットワーク関係，文字列処理，テストケース，アーカイバのドメインに属すると判断された組がそれぞれ 1 組ずづあった．

観点 2 についてのアンケート結果を以下に示す．設問 4 の回答を**表 6** に示す．回答結果より，収録された動詞-直接目的語-間接目的語の三つ組のうち，品詞の判定が間違っていると判定された三つ組の割合は，Web Application の辞書で 6%，XML の辞書で 13%，Database の辞書で 12%，GUI の辞書で 10%であった．

観点 3 についてのアンケート結果を以下に示す．設問 5,6 の回答を**表 7** に示す．(A) 適当な組だと思う，(B) どちらかと言えば適当な組だと思う”と回答した場合を肯定的評価と見なした場合，設問 5 の結果より各辞書の三つ組のうち対象ドメインの命名支援のための辞書に収録してもよいと肯定的評価を与えられた組の割合は，Web Application で 56%であり，XML で 53%であり，Database で 71%であり，GUI で 64%であった．また，設問 6 の結果より各辞書の三つ組のうち Java プログラムの命名支援のための辞書に収録してもよいと肯定的評価を与えられた組の割合は，Web Application で 35%，XML で 30%，Database

で 48%，GUI で 61%であった．

設問 7 の回答について以下に記述する．XML の辞書では GUI，構文解析器，リソース管理，木構造，グラフ解析，データ解析の辞書に収録してもよいと判断された組がそれぞれ 1 組ずつあった．Database の辞書では，GUI，Web Application の辞書に収録してもよいと判断された組がそれぞれ 1 組ずつあった．Web Application と GUI の辞書では他のドメインの辞書に収録してもよいと判断された組は無かった．

5.3.1 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当と判断した

表 4 各辞書に対する設問 1，2 の回答

	設問 1 の回答					設問 2 の回答				
	(A)	(B)	(C)	(D)	(E)	(A)	(B)	(C)	(D)	(E)
Web	21	35	7	3	24	16	29	10	30	5
XML	44	18	5	7	16	15	11	17	42	5
DB	32	36	4	9	9	22	13	32	17	6
GUI	42	26	9	6	7	32	37	9	6	6

表 5 各辞書に対する設問 3 の回答，括弧内の数字は組の数を表す．

Web	データベース (16)， Web Appication に限らず他のドメインでも一般的 (2)，入出力 (6)
XML	GUI(1)，データ解析 (2)，構文解析器 (1)，リソース管理 (1)， 木構造 (1)，グラフ解析 (1)
DB	GUI(5)，Web Appication(1)，文字列処理 (1)
GUI	GUI に限らず他のドメインでも一般的にありそう (4)，DB(1)， ネットワーク関係 (1)，テストケース (1)，アーカイバ (1)

表 6 各辞書に対する設問 4 の回答

	動詞が間違っている	直接目的語が間違っている	間接目的語が間違っている	いずれかが間違っている
Web	3	1	3	6
XML	5	7	1	12
DB	1	6	5	11
GUI	8	1	0	9

5.4 考　　察

設問 1 の結果より各辞書には辞書に含まれる三つ組のうち 60%から 70%程度ドメイン固有の三つ組が収録されていることから，ドメイン固有の関係を入力されたソフトウェアから抽出することには成功していると考えられる．しかし，設問 2 や設問 3 の結果より，各辞書には対象ドメイン以外で一般的な組が多く混入していることが分かる．この原因として以下の 2 つが考えられる．

- 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
- 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

前者の原因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．

また，観点 1 を評価する設問の結果と観点 3 を評価する設問の結果を比較すると，各ド

表 7 各辞書に対する設問 5，6 の回答

	設問 5 の回答					設問 6 の回答				
	(A)	(B)	(C)	(D)	(E)	(A)	(B)	(C)	(D)	(E)
Web	19	32	11	4	24	13	19	23	30	5
XML	33	15	10	16	16	14	13	12	46	5
DB	35	29	10	10	6	31	13	24	16	6
GUI	28	30	13	11	8	29	26	15	13	7

表 8 各辞書に対する設問 7 の回答，括弧内の数字は組の数を表す

XML	GUI(1)，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)
DB	GUI(5)，Web Appication(1)

メインの辞書に収録してもよいと判断された三つ組の数は各ドメインで一般的に見られると判断された組の数よりも少ないことが分かる．この原因は，5.3.1 節で，品詞の判定が間違っている，省略語が含まれている等の理由が挙げられている．品詞の判定については設問 1 の結果より 10%程度間違っていることが分かる．品詞の判定精度を挙げることや，ドメインで見られる組ではあるが辞書に収録すべきでは無いと判断される組を辞書に収録しないようにする方法の開発が今後の課題である．

5.5 妥 当 性

被験者に示した三つ組は，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．被験者は著者が所属する研究室に所属する学生である．そのため被験者のランダムな抽出は行えていない．

6. 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[3),9)] 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 や Aspect Mining に適用する手法を提案した．Hill ら[4)] 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 や Aspect Mining に適用している．Høst[5),6)] 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．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．

以上挙げた研究に対し，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，さらに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．これによりドメイン固有の関係を収録した命名支援のための辞書の作成を行っている点が他の研究と異なっている点である．

7. 今後の課題

今後の課題として，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挙げられる．メソッドのクラス名や，フィールド変数名など，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 Eclipse Plugin の実装を検討している．さらに，辞書中の動詞-目的語関係を視覚化することによるプログラム理解支援を検討している．

参 考 文 献

1) Corbi, T. A.: Program understanding: challenge for the 1990's, *IBM Syst. J.*, Vol.28, No.2, pp.294–306 (1989).
2) Fjeldstad, R. K. and Hamlen, W. T.: Application Program Maintenance Study: Report to Our Respondents, *Proceedings GUIDE 48* (1983).
3) 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).
4) Hill, E., Pollock, L. and Vijay-Shanker, K.: Automatically capturing source code context of NL-queries for software maintenance and reuse, *ICSE '09: Proceedings of the 2009 IEEE 31st International Conference on Software Engineering*, Washington, DC, USA, IEEE Computer Society, pp.232–242 (2009).
5) Høst, E.W. and Østvold, B.M.: The Programmer's Lexicon, Volume I: The Verbs, *SCAM '07: Proceedings of the Seventh IEEE International Working Conference on Source Code Analysis and Manipulation*, Washington, DC, USA, IEEE Computer Society, pp.193–202 (2007).
6) Høst, E.W. and Østvold, B.M.: Debugging Method Names, *Genoa: Proceedings of the 23rd European Conference on ECOOP 2009 — Object-Oriented Programming*, Berlin, Heidelberg, Springer-Verlag, pp.294–317 (2009).
7) Lawrie, D., Morrell, C., Feild, H. and Binkley, D.: What's in a Name? A Study of Identifiers, *ICPC '06: Proceedings of the 14th IEEE International Conference on Program Comprehension*, Washington, DC, USA, IEEE Computer Society, pp.3–12 (2006).
8) Oracle Corporation: The Java Tutorials. `http://java.sun.com/docs/books/tutorial/java/javaOO/index.html`.
9) Shepherd, D., Pollock, L. and Vijay-Shanker, K.: Towards supporting on-demand virtual remodularization using program graphs, *AOSD '06: Proceedings of the 5th international conference on Aspect-oriented software development*, New York, NY, USA, ACM, pp.3–14 (2006).
10) Sun Microsystems: Java Platform SE6 API 仕様. `http://java.sun.com/javase/ja/6/docs/ja/api`.
11) 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).