ソフトウェア理解支援を目的とした辞書の作成法

早瀬康裕[†1]　市井誠[†1]　井上克郎[†1]

本研究では，ソフトウェア理解を支援するために，ソフトウェア中で用いられる単語の辞書を自動的に作成する手法をする．作成する辞書には，識別子に用いられる名詞と修飾語，そしてその間の関係が記録される．

A Novel Approach for Building a Thesaurus for Program Comprehension

Yasuhiro Hayase,[†1] Makoto Ichii[†1] and Katsuro Inoue [†1]

We propose a method for building a thesaurus that helps program comprehension. The thesaurus conatins nominal or adjective words and relations between the words.

1. はじめに

近年，ソフトウェア保守にかかるコストの増大が問題となっている．ソフトウェアを変更するためには対象ソフトウェアを詳細に理解する必要があるが，ソフトウェアに付随するべきドキュメントは，ソフトウェア保守の過程で適切に変更されないことが多い[1] ため，ソフトウェアを理解するにはソースコードを読み，理解しなければならない[2]．

ソースコードを理解する際に，保守作業者は様々な手がかりを利用するが，ソースコード中に出現する識別子の名前は非常に重要な手がかりの 1 つであると言われている．具体的には，保守作業者がソースコードを理解する時には，そのプログラムの使われる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[3][4]．このため，識別子に適切な名前が付けられていない場合には，ソースコードを理解するために莫大な時間を必要としてしまう[5]．

また，識別子に適切な名前が付けられていたとしても，作業者が保守対象のソフトウェアに不慣れな場合には，プログラム要素と知識との対応付けが上手く出来ない．このことが，対象ソフトウェアに熟練している者に比べて，保守作業に長い時間がかかってしまう原因の 1 つになっていると考えられる．

そこで，本研究ではソースコード中に出現する単語とその間の関係から，単語間の関係を記した辞書（シソーラス）を作成する手法を提案する．このシソーラスを用いることで，対象ソフトウェアを効率的に理解することが出来るようになると期待される．

2. 提案手法

提案手法の概要を図 1 に示す．

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ースコードであり，出力は名詞の上位下位関係と，名詞と修飾語の関係を記録したグラフ構造の辞書である．

提案手法は，まずソースコード集合を構文解析し，識別子名とそれらの間の is-a 関係 (一方がもう一つの具体例になっている関係) を取得する．一般に，識別子名は複数の単語を連結したもの (複合語) であることが多い．そこで，複合語を切り分け，単語の列を作成する．is-a 関係の具体例としては，クラスやインタフェースの継承や実装関係や，変数とその型の関係がある．

次に，単語列の間の is-a 関係を解析する．解析結果は，以下の 3 つに分類される．

一方の単語列がもう一方に含まれる場合　長い単語列から短い単語列を除いた単語列を修飾語として抽出する．

共通して現れる部分単語列がある場合　共通する部分を除いた 2 つの単語列を，名詞の上位下位関係として抽出する．また，共通する部分を修飾語として抽出する．

それ以外の場合　2 つの単語列を，名詞の上位下位関係として抽出する．

†1 大阪大学 大学院情報科学研究科
Osaka University

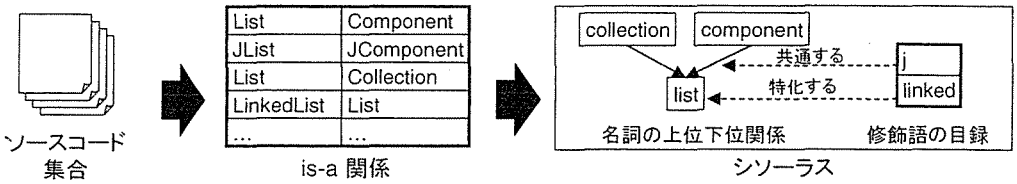

図 1 手法の概要

抽出した関係や単語が，予め定めた閾値以上の回数出現したならば，辞書に登録する．

3. 関連研究

対象ソフトウェアに不慣れな保守作業者であっても効率的な保守作業を可能にするために，保守作業者に適切な知識を与える研究が行われている．

Caprile ら[6] は，C 言語の関数名の構造を定義し，関数名に用いられる語彙の辞書を作成した．さらに，その辞書を用いて関数名の再構築を支援する手法を提案した．この手法では，対象とするソフトウェア毎に，関数名の正規文法を手動で作成しなければならないという問題がある．Host ら[7] は，メソッドの命名作業を支援することを目的として，既存の Java ソースコードを解析し，メソッド名に用いられる動詞の意味を説明した辞書を作成した．Antoniol ら[8] は，プログラム中の識別子に用いられている語彙が，開発の進行に伴ってどう変化するかについて調査した．

4. 将来の課題

ソースコード集合を解析し，シソーラスを作成するシステムは実装済みである．

次に，作成したシソーラスに，自然言語とは異なる単語や関係が存在するかについて調査するために，作成したシソーラスと WordNet などの自然言語のシソーラスとを比較することを計画している．

さらに，シソーラスから不適切な単語や関係を除去するために，より適切な閾値の設定方法を考案する．一般に，ソースコードの外部から利用される識別子は熟慮して名付けられるため適切な名前が付けられることが多いが，内部のみで使用される識別子の名前にはあまり注意が払われない．そこで，この性質を利用し，ソースコード内からのみ参照可能な識別子や，外部からの参照が少ない識別子から取得した単語間の関係は，同じ関係が複数のソースコードで出現した時のみシソーラスに登録するようにすれば良いのではないかと考えている．外部からの参照の多さの基準としては，fan-in や Component Rank 値 (実績に基く再利用性の尺度) を用いることを考えている．

以上に加え，作成したシソーラスを効果的に提示するユーザインタフェースを作成することも必要である．

参考文献

1) LaToza, T. D., Venolia, G. and DeLine, R.: Maintaining mental models: a study of developer work habits, *ICSE '06*, New York, NY, USA, ACM, pp.492–501 (2006).
2) LaToza, T.D., Garlan, D., Herbsleb, J.D. and Myers, B.A.: Program comprehension as fact finding, *ESEC-FSE '07*, New York, NY, USA, ACM, pp.361–370 (2007).
3) von Mayrhauser, A. and Vans, A.M.: Identification of Dynamic Comprehension Processes During Large Scale Maintenance, *IEEE Trans. Softw. Eng.*, Vol.22, No.6, pp.424–437 (1996).
4) Pennington, N.: Comprehension strategies in programming, pp.100–113 (1987).
5) Lawrie, D., Morrell, C., Feild, H. and Binkley, D.: What's in a Name? A Study of Identifiers, *ICPC '06*, Washington, DC, USA, IEEE Computer Society, pp.3–12 (2006).
6) Caprile, B. and Tonella, P.: Restructuring Program Identifier Names, *ICSM '00*, Washington, DC, USA, IEEE Computer Society, p.97 (2000).
7) Host, E.W. and Ostvold, B.M.: The Programmer's Lexicon, Volume I: The Verbs, *SCAM '07*, Washington, DC, USA, IEEE Computer Society, pp.193–202 (2007).
8) Antoniol, G., Gueheneuc, Y.-G., Merlo, E. and Tonella, P.: Mining the Lexicon Used by Programmers during Sofware Evolution, *ICSM '07*, Washington, DC, USA, IEEE Computer Society, pp.14–23 (2007).