特別研究報告

題目

Java クラスの利用関係を用いたソフトウェア部品のカテゴリ階層構築法

指導教官
井上 克郎 教授

報告者
宮崎 宏海

平成 19 年 2 月 20 日

大阪大学 基礎工学部 情報科学科

平成 19 年度 特別研究報告

Java クラスの利用関係を用いたソフトウェア部品のカテゴリ階層構築法

宮崎 宏海

内容梗概

ソフトウェア部品の再利用性の向上を目的としたシステムにソフトウェア部品検索がある．一般に，情報検索システムでは，システムの利用者が入力した単語をもとに検索を行うキーワード検索が用いられることが多い．キーワード検索システムの利点は，直感的で分かりやすく，検索者がキーワードを入力するため自由度が高い検索が可能な点にある．一方，あらかじめ用意された階層的な分類 (カテゴリ) を選択することで目的の文書を探す，カテゴリ検索と呼ばれる手法がある．カテゴリ検索の利用者は段階的に絞り込みを行うことで，適切なキーワードが思いつかない場合でも，目的の部品を得ることが出来る．しかし，カテゴリ検索のカテゴリ階層は，手作業で維持されるのが一般的であり，多大な時間と手間を要するという問題があった．

本研究では，ソフトウェア部品検索にカテゴリ検索を用いるためにカテゴリの階層構造を自動的に構築する手法を提案する．検索システムの利用者は段階的に絞り込みを行うため，カテゴリ階層の上位には抽象的なカテゴリが存在し，下位には具体的なカテゴリが存在しなければならない．本手法では，Java のソースコードを解析することによって単語の上位下位関係から成るシソーラスを作成し，それを用いて上記の条件に基づいたカテゴリ階層を構築する．

さらに，提案手法を実装し，実際のソフトウェアからカテゴリ階層を構築する実験を行った．その結果，カテゴリとソフトウェア部品の対応と，カテゴリ間の親子関係が適切であることが分かり，システムの有用性が確認できた．

主な用語

カテゴリ階層 (Category Tree)
シソーラス (Thesaurus)
上位下位関係 (Super-Sub Relation)
ソフトウェア部品検索 (Software Component Search)

目 次

1　まえがき

近年，高品質なソフトウェアを短時間で製作するために，ソフトウェア開発においてソフトウェア部品の再利用が頻繁に行われている．ソフトウェア部品の再利用とは，開発のコスト削減や製品の品質向上を目的として，既存のソフトウェア部品を他のソフトウェアで利用することである [2, 3, 9, 10].

一方でインターネットの普及により，SourceForge [14] などのソフトウェアに関する情報を交換するコミュニティが誕生した．SourceForge では多くのオープンソースソフトウェアを公開しており，大量のソースコードを容易に入手することが出来る．

これらの公開されているソースコードの量は膨大であるため，目的のソフトウェア部品を入手するためには検索システムが用いられることが多い．このため，目的の部品をより的確に表示する検索システムを実現することで，ソフトウェア部品の再利用を促進することが出来る．ソフトウェア部品の検索システムについて行われた研究としては，SPARS-J [8, 18] ，Koders [11]，gonzui [5]，Google Code Search [13] が挙げられる．

検索システムの代表的な検索手法として，キーワード検索とカテゴリ検索がある．キーワード検索とは，検索の問い合わせにキーワードを用いる検索システムである．キーワード検索の利点は，直感的で分かりやすく，また検索者がキーワードを決めるため自由度の高い検索が可能であることである．欠点は，目的の文書に適合しないキーワードが思いつかない場合，目的の文書が結果として出力されなかったり，不要な文書の中に埋もれてしまうことである．一方，カテゴリ検索では，検索者はあらかじめ用意された階層状のカテゴリから目的のものを探す．カテゴリ検索の利点は，検索者が適当なキーワードを思いつかない場合でも，漠然とした目的から検索できることである．

本研究ではこのカテゴリ検索によるソフトウェア部品検索に着目する．カテゴリ検索のカテゴリはその性質から手作業で作成し維持されることが多く，その作業に多大な時間と手間を要する．特に，ソフトウェア部品を対象とした場合，追加もしくは更新しようとするソフトウェア部品がどのカテゴリに属するのが適当であるかを判断するために幅広い専門知識を必要とするため，適切なカテゴリ階層を維持するのは非常に困難である．さらに，今までにない機能を持つ部品の追加やカテゴリ内の部品数の増加が起こった場合に，新しいカテゴリの追加や巨大なカテゴリの分割といった，カテゴリの再構成・維持作業が必要になる．従って，ソフトウェア部品を対象としたカテゴリ検索には，カテゴリへの分類とカテゴリの維持の自動化は不可欠だといえる．

カテゴリの階層構造について考えた場合，その構造はカテゴリ名の意味的な関係に基づいて構築されているべきである．すなわち，あるカテゴリのカテゴリ名には，その親カテゴリのカテゴリ名よりも具体的な名前が付き，その子カテゴリのカテゴリ名よりも抽象的な名前

が付いている．これは，上層のカテゴリほど抽象的なカテゴリ名が付いていれば，検索者は段階的に目的を絞り込むことができるためである．

単語間の関係を記述した辞書としてシソーラスがある．シソーラスには単語間の類義関係，反義関係，上位下位関係などが記されている．この中で，上位下位関係とは単語の概念による包含関係を意味する．

本研究では単語の上位下位関係を記述したシソーラスに着目する．そして，このシソーラスを用いてカテゴリ名による段階的な絞り込みが可能なカテゴリ階層を自動で構築する手法を提案する．しかし，既存のシソーラスはソフトウェア部品のカテゴリ階層の構築には適当ではない．なぜならば，既存のシソーラスは自然言語についての単語間の関係を記述しているが，ソフトウェア部品では自然言語とは異なる用途で単語を用いたり，複合語を用いることが多いためである．そこで，提案手法ではソフトウェア部品に適した新たなシソーラスを自動で作成し，それを用いてカテゴリ階層を構築する．

さらに，本手法を実現するシステムを作成する．そして，作成したシステムを用いて実際にソフトウェア部品の集合からカテゴリ階層を構築し，カテゴリ階層の評価を行う．

以降，2 節で既存のソフトウェア部品検索と，カテゴリ検索におけるカテゴリ階層について述べる．3 節では Java のクラス間の利用関係からシソーラスを作成し，それを利用してカテゴリ階層を構築する手法について述べる．4 節では提案手法を実装したシステムについて述べる．5 節でシステムを用いた評価実験を行い，最後に 6 節でまとめと今後の課題について述べる．

2　背景

本節では，既存のソフトウェア部品検索についてキーワード検索とカテゴリ検索の概要について述べた後，カテゴリ検索でのカテゴリ階層の構築へのシソーラスの利用と既存の関連研究について述べる．

2.1　ソフトウェア部品検索

ソフトウェア部品検索とは，部品を再利用するために大量のソフトウェア部品の集合の中から目的の部品を見つけ出す作業のことである．自然言語文書の検索システムでは，Google [6] のように検索者が目的の文書に関するキーワードを入力して検索するのが一般的であるが，ソフトウェア部品検索システムではその他に部品に関するメトリクスや記述言語，求める機能の抽象表現をキーワードに用いる場合もある．

2.1.1　キーワード検索によるソフトウェア部品検索

キーワード検索による検索システムでは，検索者が目的の部品に関するキーワードを入力すると，ソフトウェア部品の集合の中からそのキーワードと適合度の高い部品が出力される．キーワードとの適合の判定には，部品のソースコード中に指定したキーワードが出現しているか，部品の索引語としてキーワードが登録されているか，などがある．たとえば，SPARS-J では検索者が入力したキーワードが索引語として登録されている部品を出力する．

キーワード検索の利点は，直感的で分かりやすく，検索者が自身でキーワードを入力するため自由度が高い検索が可能であることである．逆に欠点としては，目的の部品に適合するキーワードを考える必要があることである．

2.1.2　カテゴリ検索によるソフトウェア部品検索

キーワードを入力しない検索手法として，カテゴリ検索がある．カテゴリ検索ではシステム側にあらかじめ用意されたカテゴリが階層的に配置されており，各カテゴリにはカテゴリ名がついている．また，カテゴリにはカテゴリ名についている単語に関連した部品が割り当てられる．検索者は階層状のカテゴリに従って段階的に目的の部品を絞り込んでいき，目的の部品に関連した単語をカテゴリ名とするカテゴリを見つければ，そのカテゴリに割り当てられた部品の中から目的の部品を探す．

カテゴリ検索の利点は，検索に用いるキーワードを検索者が考える必要がなく，漠然とした目的から検索を行うことができることである．検索者は目的の部品に関連したカテゴリを見つけるまで，検索者は目的の部品が含まれている，もしくは関連するカテゴリを子カテゴリに含んでいると思われるカテゴリを辿るだけでよく，キーワードを入力する必要がない．

カテゴリ検索のカテゴリは，内容を把握する必要があることから手作業で維持されることが多い．その多くは，その検索システムの管理者が与えたカテゴリに従って，管理者自身あるいは登録者がふさわしいカテゴリに分類する，という作業が必要となる．しかし，それには多大な時間と手間がかかる．特にソフトウェア部品を対象にすると，このコストは大きくなることが考えられる．また，管理者が与えた分類の外，あるいは中間といったものは無理矢理既存の分類に当てはめるか，カテゴリの再構成をして分類しなおす必要がある．

2.2 シソーラスとカテゴリ階層

シソーラスとは同義関係，類義関係，反義関係，上位下位関係などの単語間の関係を記した辞書である．自然言語の分野ではロジェのシソーラス [4] や WordNet [1] などが電子化されたシソーラスとして開発されている．

一方，カテゴリ階層とはカテゴリ間の親子関係によって構成される階層構造である．一般に，カテゴリ階層の親子関係には親カテゴリを 1 つしか認めない木構造と親カテゴリを複数個認める有効グラフ構造の 2 種類がある．本研究で扱うカテゴリ階層は親を複数個認める有向グラフ構造のものとする．

カテゴリ検索で段階的な絞り込みを行うためには，あるカテゴリのカテゴリ名はその子カテゴリより抽象的な名前がつき，その親カテゴリよりも具体的なカテゴリ名が付いていなくてはならない．そこで，単語間の包含関係を表す上位下位関係のシソーラスをカテゴリ階層として用いることを考える．

しかし，既存のシソーラスは自然言語について記述したものであるため，ソフトウェア部品のカテゴリ階層の構築にそのまま用いることは出来ない．これは，ソフトウェアにける単語の意味が一般的な単語の意味と異なるものがあるためである．例えば，Java には List という型が Collection の型の部分方として存在する．このため，Java のソースコード中では，List は Collection の下位であるといえる．しかし，WordNet などの一般的なシソーラスには，このような関係は記述されていない．このような意味の異なる単語が存在するため，ソフトウェア部品のカテゴリ階層を構築するためには，一般的なシソーラスではなく，ソフトウェア部品用の新たなシソーラスを作成する必要がある．

2.3 関連研究

我々の研究チームでは Java のソースコードを対象としたソフトウェア部品検索システム SPARS-J(Software Product Archive, analysis and Retrieve System for Java) を作成した．SPARS-J はキーワード検索システムであり，検索結果の順位付けには，部品に対する重みを出現箇所と TF-IDF 法から求める KR 法 (Keyword Rank 法) と，部品の重みを部品間の

利用実績から求める CR 法 (Component Rank 法) の 2 つの手法を用いている．

ソフトウェアを対象とした分類手法として，川口らの LSA に基づく手法 [16] が挙げられる．この手法では，ソフトウェアのソースコード中に出現する識別子に対して LSA[12] を適用し，その結果を用いて識別子によるソフトウェアの非排他的クラスタリングを行っている．これにより，前提知識を用いずにソフトウェアの集合を機能やライブラリといった複数の視点による自動分類を実現している．

また，ソフトウェア部品をカテゴライズした手法としては仁井谷らのソースコード中の特徴語に基づく手法 [17] がある．この手法では，ソースコード中の単語に対して出現位置による重みを付け，さらに複合語や利用関係を考慮することで，部品の特徴を表す特徴語を選出し，その特徴語をカテゴリ名とすることで自動分類を行っている．

自然言語のシソーラスの自動構築手法としては，Hearst による構文パターンのマッチングによる単語間の上位下位関係を取得する手法 [7] がある．この手法では，特定の構文パターン中の単語には上位下位関係が生じやすい事を利用している．

また，構造化文書を対象としたシソーラス構築手法として，新里らの HTML 文書の構造を利用した手法 [15] がある．新里らは，箇条書きやリストボックスなどのように文書中に繰り返して出現する要素に使用される単語は意味的に類似しており共通の上位語を持ちやすいことに着目している．さらに，単語の係り受けを調べることで共通の上位語と各下位語の類似度を調べて，精度を高めている．

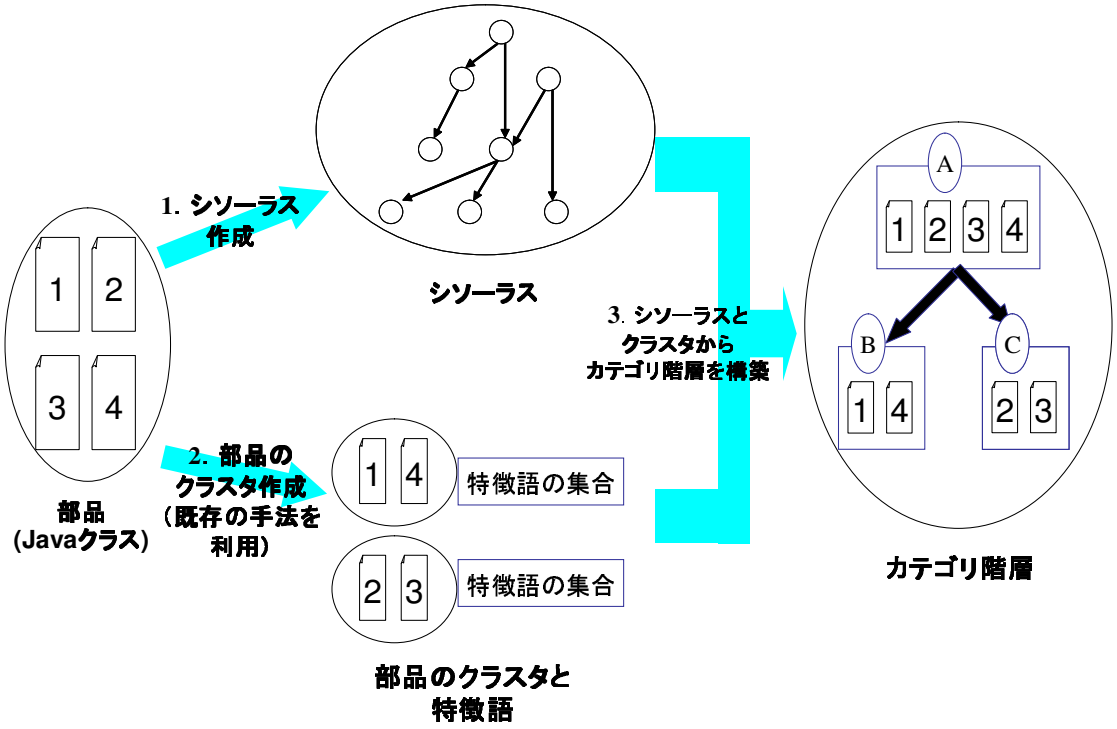


図 1: 提案手法の概要

3 提案手法

本節では，カテゴリ階層を構築する手法について述べる．まず概要を述べ，続いて手法の詳細を手順ごとに述べる．

3.1 概要

提案する手法の概要を図 1 に示す．まず Java クラスの集合を入力として，単語間の上位下位関係を持つシソーラスを作成する．次に，同じ Java クラスの集合を入力として Java クラスのクラスタリングを行い，得られたクラスタに特徴語を付ける．最後に，得られたシソーラスとクラスタからカテゴリ階層を構築する．以降，それぞれの手順について詳しく説明する．

3.2 シソーラスの作成

本研究では，シソーラスを「単語を頂点とし，単語間の上位下位関係を上位の頂点から下位の頂点に引いた有向辺で表した有向グラフ」であると定義する．そして，Java クラスの

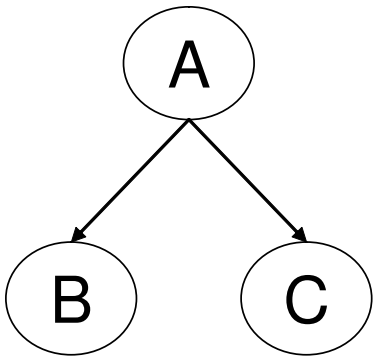


図 2: シソーラスの例

特定の利用関係に注目することで単語間の上位下位関係を取得する．

図 2 にシソーラスの例を示す．図 2 では，シソーラスには A, B, C の 3 つの単語が含まれ，A は B および C の上位であることが表されている．

3.2.1 利用関係による単語間の上位下位関係の取得

Java クラスの利用関係から，単語の組と上位下位関係を取得する．用いる利用関係を表 1 に示す．ただし，クラス名やインタフェース名は名前空間を除いたものを取得する．たとえば，java.util.List が java.util.Collection を継承していることから，Collection を上位，List を下位とする上位下位関係を得る．

次に，取得した単語の表記をキャメルケースの形式で統一する．キャメルケースとは複数の単一語で構成されている複合語に対して単一語の先頭文字を大文字で書き，先頭以外の文字を小文字で書く表記法である．具体的には，全ての単語に対して以下の処理を加える．

- 1 文字目を全て大文字に変換
- "_" (アンダーバー) は全て削除して "_" の次の 1 文字を大文字に変換する

表 1: 取得する利用関係

	上位	下位
継承関係	親クラス名	子クラス名
	親インタフェース名	子インタフェース名
実装関係	インタフェース名	実装するクラス名
フィールド変数の型と変数	変数の型名 (クラス名)	変数名

例 databaseMeta_data -> DatabaseMetaData

また，以下のいずれかの条件を満たす単語は部品の機能や特徴を表すことがほとんどなく，部品のカテゴリ階層名の構築に用いられることを前提としたシソーラスに含まれる単語として不適切である．そのため，取得した単語の組のうち，いずれかの単語が以下の条件を満たす組を削除する．

- 1 文字の単語
- 数字のみで構成されている単語

3.2.2 不適切な上位下位関係の削除

取得する利用関係は上位下位関係の期待が出来るものを選択したが，単語間に必ずしも上位下位関係が成立するとは限らない．そこで，取得した関係が上位下位関係かどうかを出現回数で判断する．出現回数が多い単語の組ほど上位下位関係の確からしさが高いと考えられる．

3.3 Java クラスのクラスタと特徴語の決定

本研究では，川口 [16] により提案されたソフトウェアの分類手法に基づく手法を用いて部品のクラスタリングを行う．この手法で得られたクラスタには 10 個の特徴語がついている．ただし，川口の手法に若干の変更を加えている．図 3 および以下にその概要を示す．

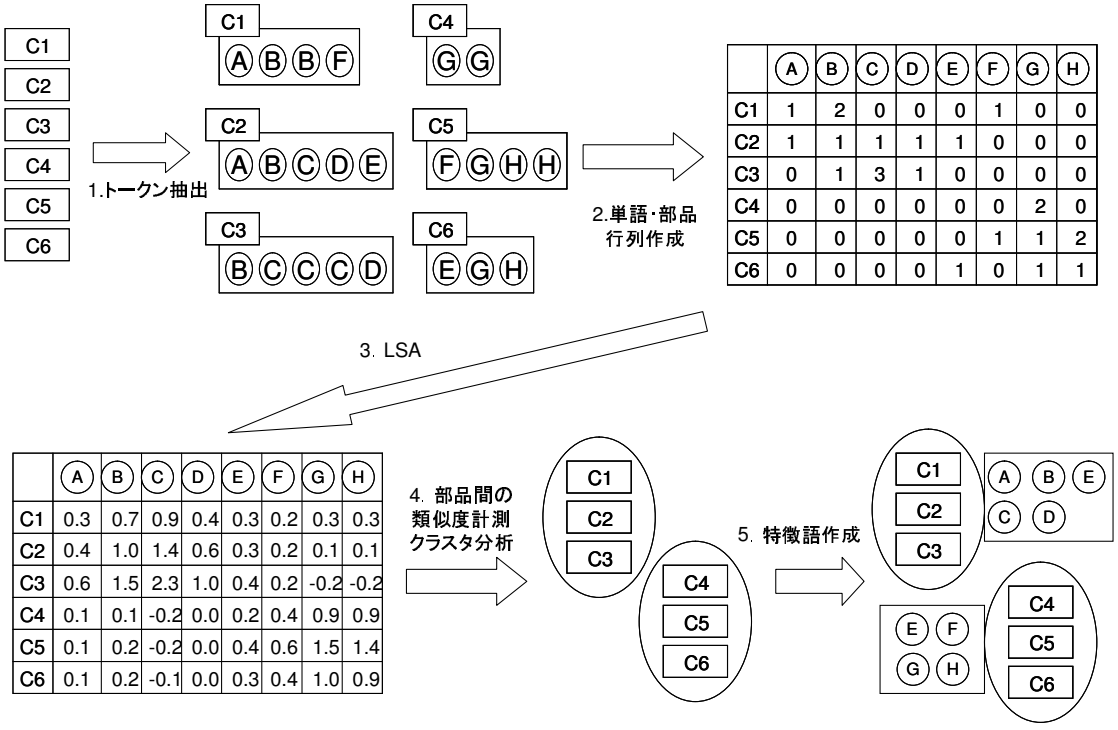
まず部品のソースコードからトークンの抽出を行い，単語の出現数を要素とする単語×部品の行列を作成する．次に，作成した行列に対して LSA を適用する．LSA の結果を用いて，部品間の類似度を計測する．その類似度を用いてクラスタ分析を行い，部品を排他的に分類する．得られた全てのクラスタに対して特徴的な単語を 10 個抽出する．

3.4 カテゴリ階層の構築

作成したシソーラスと Java クラスのクラスタリング結果からカテゴリ階層を構築する．

3.4.1 カテゴリの作成

まず，クラスタリング結果からカテゴリを作成する．クラスタの特徴語がシソーラスに登録されていれば，その特徴語をカテゴリ名とするカテゴリを作成する．図 4 では A，B，E，F のカテゴリが作成される．そして，クラスタに含まれている部品をカテゴリに割り当てる．図 4 では，1 個目のクラスタに含まれる部品はカテゴリ A および B に，2 個目のクラスタに



	A	B	C	D	E	F	G	H
C1	1	2	0	0	0	1	0	0
C2	1	1	1	1	1	0	0	0
C3	0	1	3	1	0	0	0	0
C4	0	0	0	0	0	0	2	0
C5	0	0	0	0	0	1	1	2
C6	0	0	0	0	1	0	1	1

	A	B	C	D	E	F	G	H
C1	0.3	0.7	0.9	0.4	0.3	0.2	0.3	0.3
C2	0.4	1.0	1.4	0.6	0.3	0.2	0.1	0.1
C3	0.6	1.5	2.3	1.0	0.4	0.2	-0.2	-0.2
C4	0.1	0.1	-0.2	0.0	0.2	0.4	0.9	0.9
C5	0.1	0.2	-0.2	0.0	0.4	0.6	1.5	1.4
C6	0.1	0.2	-0.1	0.0	0.3	0.4	1.0	0.9

図 3: ソフトウェア部品のクラスタリングの概要

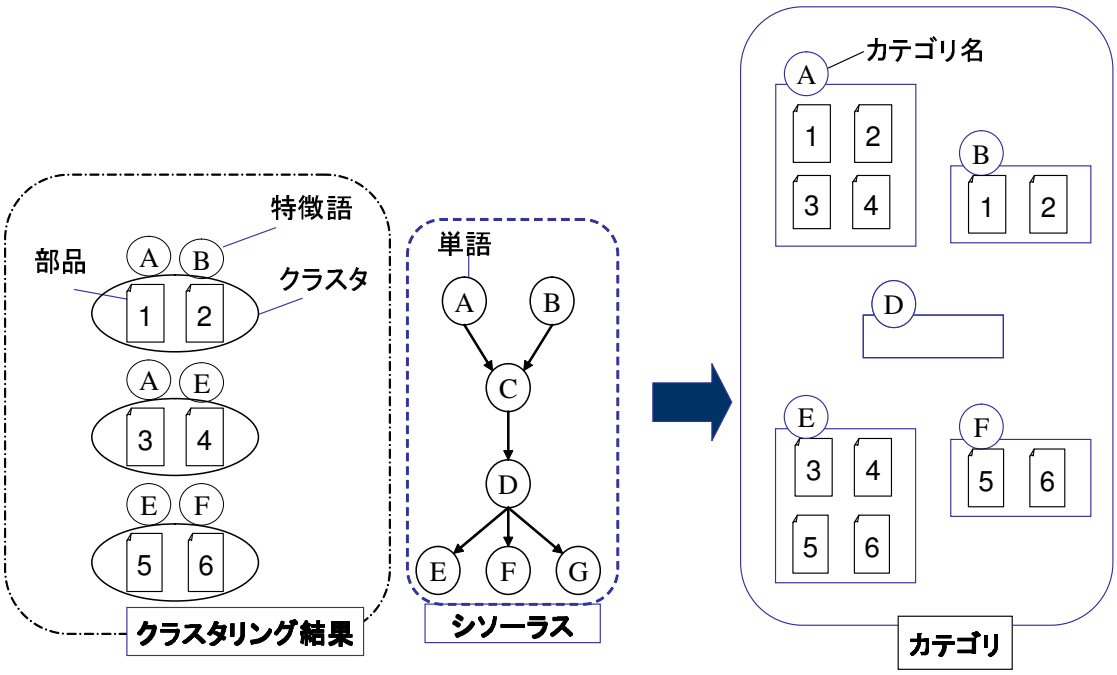


図 4: カテゴリの作成

含まれる部品はカテゴリ A および E に，3 個目のクラスタに含まれる部品はカテゴリ E および F に，それぞれに分類される．

次に，以下の条件を満たす単語をカテゴリ名とするカテゴリを作成する．

- クラスタの特徴語に含まれない
- シソーラスにおける下位の単語のうち，2 個以上の単語がクラスタの特徴語に含まれる

図 4 では D のカテゴリが作成される．この処理で作成されるカテゴリには部品が含まれないが，絞り込みには有用である．

3.4.2 カテゴリ間の親子関係の作成

作成したシソーラスを用いてカテゴリ間の親子関係を作成する．作成されたカテゴリ集合に含まれる任意の 2 カテゴリについて，それぞれのカテゴリ名の上位下位関係がシソーラスに含まれていれば，それらのカテゴリ間に親子関係を作成する．この親子関係は，上位の単語が親カテゴリ名，下位の単語が子カテゴリ名となるように作成される．

また，2 つのカテゴリが，それぞれのカテゴリ名について以下の条件を共に満たす場合，それらのカテゴリ間に親子関係を作成する．この親子関係は，条件における経路の始点の単語が親カテゴリ名，経路の終点が子カテゴリ名となるように作成される．

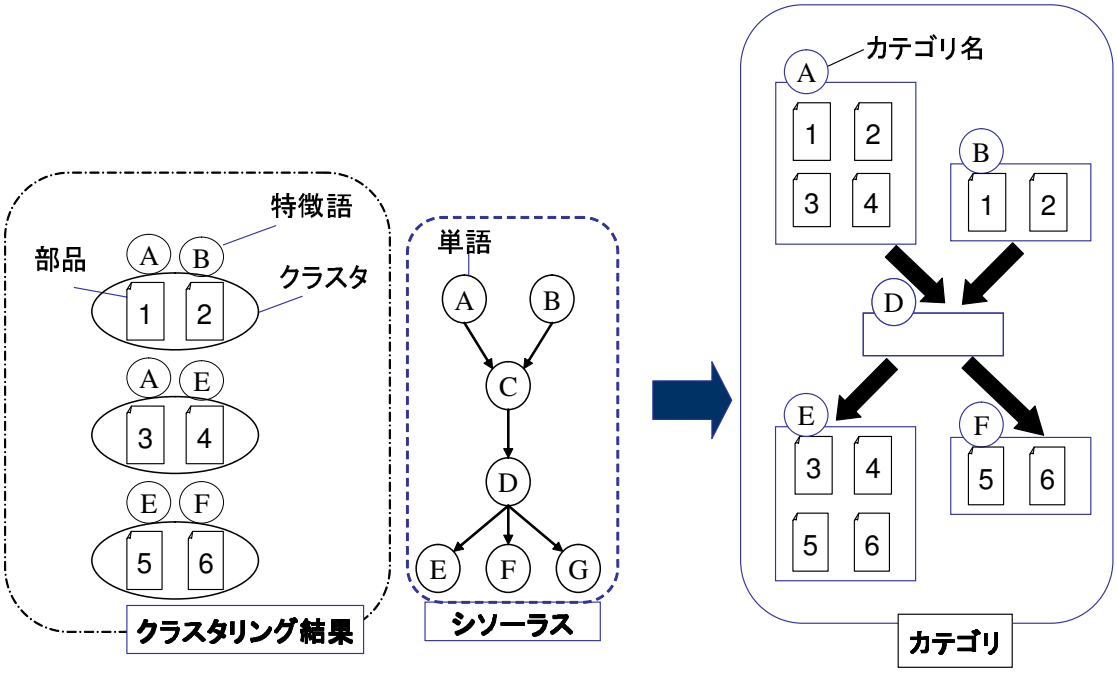


図 5: カテゴリ間の親子関係の構築

- 有向グラフ中にカテゴリ名に対応する頂点の経路が存在する

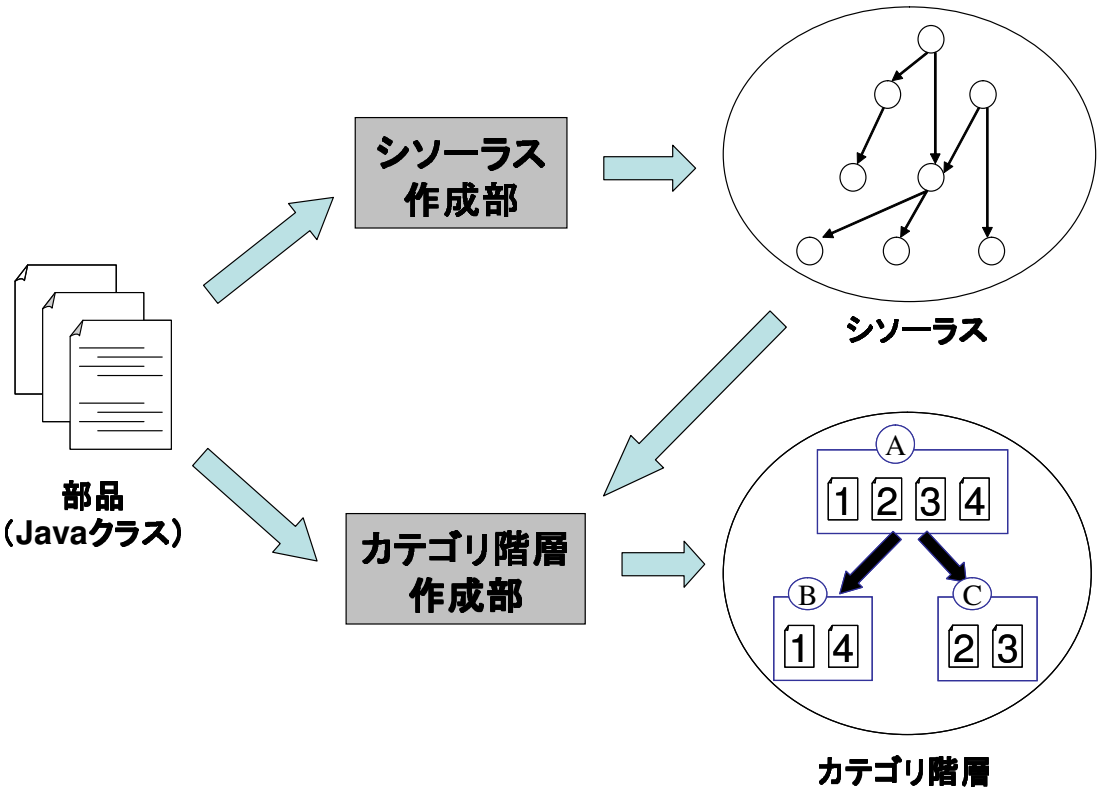


図 6: システムの概要図

4　カテゴリ階層自動構築システムの実装

本節では，作成したカテゴリ階層自動構築システムの実装について述べる．システムは，シソーラス作成部とカテゴリ階層構築部の 2 つに大別出来る．概要を図 6 に示す．シソーラス作成部には，提案手法のうち 3.2 節で述べたシソーラスを作成する手順が実装されている．また，カテゴリ階層構築部には，3.3 節で述べた部品のクラスタリングおよび 3.4 節で述べたカテゴリ階層を構築する手順が実装されている．以下，それぞれについて詳しく述べる．

4.1　シソーラス作成部

4.1.1　SPARS-J を用いた上位下位関係の取得

入力 Java ソースコードの集合

出力 上位下位関係にある単語の組 (CSV ファイル)

SPARS-J は，登録された Java ソースコードから部品を抽出し，部品間の利用関係，利用関係の種類，部品のクラス名やメソッド名，などの情報をデータベースに登録する．本研究では，入力となる Java ソースコードの集合を SPARS-J に登録し，そのデータベースから 3.2.1 節で述べた継承関係，実装関係，フィールド変数の型と変数名の関係にあたる単語の組を取得する．取得した単語の組は,「上位の単語，下位の単語，関係の種類，出現回数」を列とする CSV ファイルに出力される．

4.1.2 閾値に満たない上位下位関係の削除

入力 上位下位関係にあたる単語の組 (CSV ファイル)

出力 修正後の上位下位関係にあたる単語の組 (CSV ファイル)

入力された単語の組のうち，出現回数が利用関係ごとに設定した閾値未満である単語の組を削除する．残った単語の組を上位下位関係にあたる単語の組として,「上位の単語，下位の単語」を列とする CSV ファイルに出力する．シソーラスは，この CSV ファイルにより表現される．すなわち，CSV ファイルに含まれるすべての単語を頂点，それぞれの行で示される上位下位関係を有効辺とするグラフが，得られるシソーラスである．

4.2 カテゴリ階層構築部

4.2.1 ソフトウェア部品のクラスタリング

入力 Java ソースコードの集合

出力 クラスタリング結果

部品のクラスタリングには，報告者の所属する研究室で開発された既存のツールを用いる．このツールには，3.3 節で述べた手法が実装されている．カテゴリ階層構築部は，Java ソースコードの集合をそのツールに入力することで部品のクラスタリングを行う．クラスタリング結果として，部品のクラスタの集合，および，それぞれのクラスタの特徴語の集合が得られる．

4.2.2 カテゴリ階層の構築

入力 上位下位関係にある単語の組 (CSV ファイル)，クラスタリング結果

出力 カテゴリ (CSV ファイル)，カテゴリ間の親子関係 (CSV ファイル)

カテゴリ階層構築部は，クラスタリング結果を用いてカテゴリを作成し,「カテゴリ名，部品名」を列とする CSV ファイルとして出力する．この CSV ファイルには，3.4.1 節で述べた方法により得られた，カテゴリと部品の所属関係が記録される．なお，部品を含まないカテゴリはこの CSV ファイルには含まれず，後述するカテゴリ間の親子関係の CSV ファイルにのみ含まれる

また，カテゴリ階層構築部は，3.4.2 節で述べた方法により，カテゴリ間の親子関係を構築し，結果を「親カテゴリ名，子カテゴリ名」を列とする CSV ファイルに出力する．

5 実験

5.1 目的

提案するシステムが作成するカテゴリ階層の有用性を確認するために実験を行う．実験では，作成したシステムを用いて実際のソフトウェアに対するカテゴリ階層を構築し，カテゴリ間の親子関係の妥当性とカテゴリに割り当てられた部品の妥当性を評価する．

5.2 実験内容

5.2.1 システムへの入力

システムへの入力は JDK1.4 に含まれる全てのクラス (表 2) とした．JDK1.4 を入力とした理由は，以下の通りである．

- 規模が小さ過ぎない
- 多くの開発者に頻繁に利用される部品から成る
- クラス名や変数名などの識別子に適切な名前付けがなされていると期待できる

5.2.2 シソーラスに登録する閾値の設定

出現した識別子の組は，出現回数が表 3 に記した値以上のものをシソーラスに登録した．クラス名は外部に公開することが多いため適切な名前が期待できると判断し，継承関係と実装関係の閾値は 1 回とした．一方，フィールド変数は外部に公開しない場合もあるため，閾値を 10 回以上とした．

5.2.3 出力

システムが出力したカテゴリ階層の概要を表 4 に示す．

表 2: JDK1.4 に関する情報

総ファイル数	6024 個
総クラス数	10123 個
総行数	2066967 行

5.3 評価内容

まず，部品の割り当てが適切かを評価するために，カテゴリ名と部品の適合率を求める．次に，階層構造が適切かを評価するために，親子関係を構築しているカテゴリの親子関係の適合率を計測する．

5.3.1 カテゴリ名と部品の適合率

出力した 1109 個のカテゴリの中からランダムに 50 個のカテゴリを取得し，そのカテゴリ名と割り当てられた部品 324 個の適合率を計測した．ここで評価する適合率は以下の式で表される．

$$\frac{\text{カテゴリに割り当てられている部品のうちカテゴリ名に適合する部品の数}}{\text{カテゴリに割り当てられている部品の数}}$$

また，以下に示す 5 つの条件のうち，いずれか 1 つに該当した場合に，適合する部品であると判断した．

- カテゴリ名と部品名が同じである
- カテゴリ名が部品名の複数形や省略語である
- カテゴリ名が部品名の部分語として使われている
- カテゴリ名が部品名の類似語である
- カテゴリ名が部品の特徴の一部を表している

5.3.2 カテゴリの親子関係の適合率

出力した 2501 組の関係の中からランダムに 200 組の関係を抽出し，その親子関係の適合率を計測した．ここで評価する適合率は以下の式で表される．

$$\frac{\text{取得したカテゴリのうち親子関係が適合するカテゴリの組の数}}{\text{取得したカテゴリの組の数}}$$

表 3: 閾値の設定

利用関係	閾値 (出現数)
継承関係	1 回
実装関係	1 回
フィールド変数の型と変数名	10 回

親カテゴリ名を A，子カテゴリ名を B としたとき，B が A の一種である場合に，その親子関係が適合するとした．

5.4 結果

5.4.1 カテゴリ名と部品の適合率

適合率は 83.6 % (271 組/324 組) であった．適合条件別の適合度は表 5 のようになった．

5.4.2 カテゴリの親子関係

適合率は 64.0 % (128 組/200 組) となった．表 6 に適合例の一部を，表 7 に不適合例の一部を示す．

5.5 考察

カテゴリ名と部品の適合率は 83.6 %と高い値であったため，ソフトウェア部品のカテゴリへの割り当ては適切に行われている．

しかし，カテゴリ名と部品が 1 つも適合していないカテゴリも存在した．これは，特徴語をどのクラスタに対しても 10 個の単語としてしまったため，特徴語に部品の特徴を表す単語として不適切なものが含まれてしまったことが原因と考えられる．

一方，カテゴリの親子関係は 64.0 %とあまり高い値ではなかった．この原因としては，入力としたソフトウェア部品の集合である JDK1.4 の中で，継承や実装が正しく用いられていなかったことが原因であると考えられる．例えば，ある部品の特定の機能を使用するために継承を用いる場合があった．このような場合には，子クラスの名前が親クラスの名前の一種とならないため，シソーラスに上位下位関係として不適切な関係が抽出されてしまう．

また，フィールド変数の型と変数名の組の出現回数はすべて閾値以下であったため，フィールド変数の型と変数名の組がシソーラスに登録されることは無かった．閾値を 10 回から 2 回に変更した場合には，フィールド変数の型と名前からいくつかの関係がシソーラスに抽出

表 4: 出力

総カテゴリ数	1109 個
カテゴリ間の親子関係	2501 組
カテゴリに割り当てられた部品数	6000 個
カテゴリに割り当てられた延べ部品数	18583 個
カテゴリに含まれる部品数 (平均)	16.75 個

表 5: カテゴリ名と部品の適合率

適合の種類	適合率 (適合数/総数)	適合例 (上:カテゴリ名，下:部品名)
カテゴリ名と部品名が同じ	7.1 % (23/324)	Receiver javax.sound.midi.Receiver.java
カテゴリ名が部品名の複数形や省略語	0.3 % (1/324)	Channel java.nio.cahnnels.Channels.java
カテゴリ名が部品名の部分語	23.1 % (75/324)	Holder org.omg.CORBA.CharHolder.java
カテゴリ名が部品名の類似語	0.9 % (3/324)	PipedWriter java.io.PipedOutputStream.java
カテゴリ名が部品の特徴の一部を表す	52.2 % (169/324)	ItemEvent java.awt.Checkbox.java
合計	83.6 % (271/324)	

表 6: 親子関係の適合例

上位	下位
AbstractMap	HashMap
Cursor	CustomCursor
Member	Method

表 7: 親子関係の不適合例

上位	下位
DefaultListCellRenderer	UIResource
TableModelListener	JTable
Thread	Daemon

された．これらの抽出された関係の中には上位下位関係として適切なものも存在したが，多くはクラス名とその省略語であるなど不適切なものであった．フィールド変数の型と変数名から適切な上位下位関係を取得するためには，その出現回数を数えるだけではなく，他の手法を用いる必要があるのではないかと考えられる．

6　まとめと今後の課題

本研究では，Java クラスの利用関係と Java クラスのクラスタリング結果を用いてカテゴリ階層の構築を行う手法の提案を行った．

また，システムを実装して提案する手法によってシソーラスを用いることで意味的な絞り込みに適したカテゴリ階層を構築できることを示した．

今後の課題として，カテゴリ階層中の親カテゴリと子カテゴリの親子関係の適合率を改善することが挙げられる．具体的には，ソフトウェア部品中で用いられている識別子の中でも，その部品をその部品の外から利用するために必要な名前には適切な名前付けがなされていると考えられることから，シソーラスへの登録の閾値を private, public 等のアクセス制御の種類によって変化させることなどが考えられる．他に，カテゴリ間の関係に親子関係以外の関係を導入することでカテゴリ検索の有用性を高めることも考えられる．カテゴリ検索を行うためのユーザーインターフェースの開発も重要な課題である．

謝辞

本研究の全過程において，常に適切な御指導及び御助言を賜わ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致します.

本研究において，常に適切な御指導及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 松下誠 助教授に深く感謝致します.

本論文の作成において，常に適切な御指導及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 氏に深く感謝致します.

本論文の作成において，常に適切な御指導及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 市井誠 氏に深く感謝致します.

本論文の作成において，常に適切な御指導及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 肥後芳樹 氏に深く感謝致します.

本論文の作成において，常に適切な御指導及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 仁井谷竜介 氏に深く感謝致します.

最後に，その他様々な御指導，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 井上研究室の皆様に深く感謝致します.

参考文献

[1] A.Miller, R.Beckwith, C.Fellbaum, D.Gros, and R.Tengi. Five papers on wordnet. *Technical Report CSL Report 43*, 1990.

[2] V.R. Basili, G.Caldiera, F.McGarry, R.Pajerski, G.Page, and S.Waligora. The software engineering laboratory - an operational software experience. In *Proceedings of 14th International Conference on Software Engineering*, pp. 370–381, Melbourne, Australia, 1992.

[3] C.Braun. Reuse. In J.J. Marciniak, editor, *Encyclopedia of Software Engineering*, Vol.2, pp. 1055–1069. 1994.

[4] Chapman. L.r.roget's international thesaurus. *Thomas Y.Crowell Company Inc.*, 1977.

[5] gonzui. `http://gonzui.sourceforge.net/`.

[6] Google. `http://Google.com`.

[7] Hearst. Automatic acquistiotion of hyponyms from large text corpora. *In Proceedings of the 14th International Conference on Computational Linguistics*, pp.539–545, 1992.

[8] Katsuro Inoue, Reishi Yokomori, Tetsuo Yamamoto, Makoto Matsushita, and Shinji Kusumoto. Ranking significance of software components based on use relations. *IEEE Transactions on Software Engineering*, Vol.31, No.3, pp. 213–225, 2005.

[9] S.Isoda. Experience report on a software reuse project: Its structure, activities, and statistical results. In *Proceedings of 14th International Conference on Software Engineering*, pp.320–326, Melbourne, Australia, 1992.

[10] B.Keepence. Using patterns to model variability in product families. In *IEEE Software*, Vol.16, No.4, pp.102–108, 1999.

[11] Koders. `http://koders.com/`.

[12] T.K. Landauer, P.W. Foltz, and D.Laham. An introduction to latent semantic analysis. *Discourse Processes*, Vol.25, pp.259–284, 1998.

[13] Google Code Search. `http://www.google.com/codesearch`.

[14] SourceForge. http://sourceforge.net/.

[15] 新里圭司, 鳥澤健太郎. Html 文書からの単語間の上位下位関係の自動獲得. 自然言語処理, Vol.12, No.1, pp.125–150, 1 2005.

[16] 川口真司, 松下誠, 井上克郎. 潜在的意味解析法 LSA を利用したソフトウェア分類システムの試作. 情報処理学会研究報告, Vol.2003, No.22, pp. 55–62, 3 2003.

[17] 仁井谷竜介. ソースコードの特徴語を用いた java ソフトウェア部品の分類システム. 情報処理学会研究報告, Vol.2005, No.75, pp.49–56, 7 2005.

[18] 横森励士, 梅森文彰, 西秀雄, 山本哲男, 松下誠, 楠本真二, 井上克郎. Java ソフトウェア部品検索システム SPARS-J. 電子情報通信学会論文誌, Vol.J87-D-I, No.12, pp.1060–1068, 12 2004.