B-006

機能に着目したソフトウェア部品の抽象化による特徴抽出

Feature extraction by abstraction of software component that pay attention to function

茅野 良洋‡ Yoshihiro Kayano
織田 健† Takeshi Oda

1 はじめに

近年、ソフトウェア開発の大規模化に伴い、品質低下、コストの増大が問題となっている。これらの問題を解決するために、ソフトウェアの再利用が有効とされ、ソフトウェアリポジトリを利用したソフトウェア部品の再利用が研究されている。その再利用のためのソフトウェアリポジトリには、部品を適切に探し出すための部品検索機構が必要である。ソフトウェア検索の分野ではソースコードのクラス名等とキーワードに論理演算子を用いて記述した検索キーとで字面の一致を取る手法が広く用いられている [1]。しかし、キーワードでは部品の機能を表現するには不十分である。本稿では部品の機能による検索を目指し、特徴抽出を用いた部品検索を提案する。

2 背景と目標

2.1 ソフトウェア部品検索

現在、ソフトウェア部品検索で一般的な手法は検索キーとしてキーワードを用いるものであるが、キーワードでは文脈を表現できず部品の機能を表現するには不十分である。また、検索キーとして形式的仕様を用いる手法が提案されているが、抽象的な表現ができないため、検索キーの記述に柔軟性を持たせた検索ができない。部品の機能による部品検索では、検索キーが機能を表現できるほどの記述力をもち、かつ、抽象的な表現が必要である。

2.2 過去の研究

部品の機能による検索を目指すために、我々は関数型計算モデルに着目し、プログラム演算の特性を基に形式的仕様言語 CafeOBJ[2] で記述された仕様からの特徴抽出手法を提案した [3]。この手法では、キーワードを用いた手法より形式的で部品の機能を表現できていたが、データ構造が抽象化されておらず、仕様の全ての式を用いて計算していたために抽象度が低く、複雑なシステムの仕様を表現するのには適していなかった。

2.3 研究の目標

本研究では、部品の機能による部品検索を目標としている。より抽象度を高めるために、部品の仕様から余分な式をふるい落としながら要約し特徴を抽出する手法を提案する。この手法では、図 1 のように部品の形式的仕様から部品を抽象化して部品の機能を要約して抽出し、それを特徴として検索対象とする。また、検索キーは特徴と同じ形式で入力し、特徴と検索キーの文字列一致によって目的の部品を検索する。

†電気通信大学 情報通信工学科
‡電気通信大学 電気通信学研究科 情報通信工学専攻

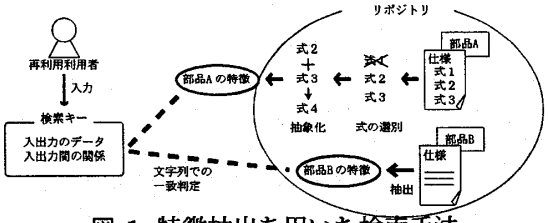

図 1:特徴抽出を用いた検索手法

3 特徴抽出を用いた部品検索

3.1 着目点

部品の機能を検索対象にするには、部品という厳密な定義を要約し抽象的な表現にする必要がある。検索利用者としては部品がどのような処理をしているかよりもその部品からどのような出力結果が得られるかという部分が大事である点に着目し、部品の機能が部品の入出力とその関係で抽象的に表現できるようにしたい。そこで、部品の特徴は実行可能なコードからデータ構造の抽象化とアルゴリズムの除去をして入出力とその関係で表現される機能を抽出し、その機能を要約することで抽出する。これにより、キーワード検索では表現できなかった文脈を特徴においては部品の機能を要約することで表現する。

3.2 特徴の定義

部品の特徴は、以下の性質を満たし、入出力の値と入出力間の関係で表現されるように抽出する。

1. 部品の最も代表的な機能を表現している
2. データ構造を抽象化している
3. アルゴリズムを含まない

テキストの自動要約の概念を取り入れることで代表的な機能を抽出する。Edmundson はテキストの要約をする際に各文に重み付けをする事で重要な文を抽出しており、特に複数の重み付けの方法を組み合わせることでより要約の精度があがるとしている [4]。要約のため文に重みをつける式は以下の通りである。

$$W(s) = \alpha C(s) + \beta K(s) + \gamma L(s) + \delta T(s)$$

このとき、$W(s)$ は文 s の総合的な重み、$C(s)$ は手がかり語、$K(s)$ はキーワード、$L(s)$ は位置情報、$T(s)$ はタイトルをもとにした文 s の重みである。

3.3 特徴抽出手順

3.3.1 重み付け

まず、仕様の各式に対して重みを付けていく。このとき、式 e の重み $W(e)$ は以下の式で計算される。

$$W(e) = \alpha C(e) + \beta T(e,t) + \gamma A(e) \quad (1)$$

$C(e)$ は手がかり表現を基に計算される。重要な予約語や変数名、関数名、演算子などには大きい値を、重要でないものには負の値を設定しておき、式内のそれらの値の合計値が $C(e)$ の値となる。次に、$T(e,t)$ は部品名 t を基に計算する。部品名 t は CafeOBJ のモジュール名に当り、部品名と変数名や関数名との類似度判定をし、類似度が高い変数名や関数名には大きい値をつけて式 e に対するそれらの出現度を計算していく。$A(e)$ は構造を基に計算する。式内に再帰構造やその他の関数を使用している式には大きい値をつけて計算する。α、β、γ の値はそれぞれの計算方法に対する係数となる。

3.3.2 重み上位の式の抽出

仕様内の各式 e に対して重み $W(e)$ が上位の式を抽出する。ただし、上位のものを持ってくるだけでは機能を表現するのに必要な式を全て含んでいるとは限らないため、重みの分布等も考慮する。重みの上位何割の式を取ってくるかや他の抽出法については検討中である。

3.3.3 抽出した式への関連付け

3.3.2 で抽出した式が単体で機能を表現しているとは限らず、その抽出した式が機能を表現するために必要不可欠な式が他の式に存在する可能性がある。そこで、その抽出した式を抽象化するために必要な情報をその式に関連付けする。具体的には、3.3.2 で抽出した式に対して、その式内に出てくる関数を含む式を全て関連付けし、また、その式で展開できる式があれば展開しておく。

例えば、マップ関数 map と要素を 2 倍する関数 f が定義されるとき、これを CafeOBJ で記述すると公理は以下のようになる。

$$eq \quad map(nil) = nil\,. \tag{2}$$
$$eq \quad map(I.L) = f(I).map(L)\,. \tag{3}$$
$$eq \quad f(I) = 2 * I\,. \tag{4}$$

nil は空のリストを示し、$I.L$ はリスト L の要素が整数値 I であることを示している。このとき、(3) 式が抽出された式である場合、(3) 式は、(2) 式が関連付けされ、(4) 式により以下のように展開される。

$$map(I.L) = (2 * I).map(L)\,.$$

3.3.4 入出力値間の関係の抽象化

3.3.2 で抽出した式をアルゴリズムを含まないレベルまで抽象化する。この際、3.3.3 で関連付けされた情報があれば用いる。抽象化の方法としては、ある構造やアルゴリズムへ対応した抽象化のルールを用意しておき、それらが式内に見つかればルールを適用していくことで抽象化する。入出力に関係のある構造やアルゴリズムがルールの対象の中心となる。

上述の map 関数の例に抽象化を適用すると、(2) 式と (3) 式から再帰構造が見つかり、入出力間の関係としては次の式が導き出される。

$$I * 2 \rightarrow I'\{I \in L, I' \in L'\}$$

この式では左辺は入力値を表し、右辺は出力値を表す。また、{} はこの式に対する条件を表す。この式により、リスト L の全ての要素に対して、要素を 2 倍するという部分が適用されることを表現している。

3.3.5 入出力値の抽象化

データ構造の抽象化を行うため、各式の入出力の値を抽象化する。抽象化した値は本稿ではデータと呼ぶ。データは BNF で表すと以下の通りとなる。

データ = 数値 | 文字 | 論理値 | 任意の値
| データの集まり | データの列

数値は数値演算が可能な値、文字は文字型等、論理値は Bool 代数に基づいた真偽値となり、その他の単一の値は任意の値とする。

データの集まりには 2 種類あり、集合などのデータが同じデータの集まりのものと、レコードなどの異なるデータの集まりのものとなる。また、データの列は同一のデータの集まりに順序関係を持つもので、リストや配列などが該当する。

3.4 特徴の形式

3 式を基に抽出した特徴は以下の通りである。

signature

(数値 : I) の列 : $L \rightarrow$ (数値 : I') の列 : L'

constraint

$I * 2 \rightarrow I'\{I \in L, I' \in L'\}$

signature では入出力の値を表現し $\rightarrow$ の左辺に入力を右辺に出力を記述する。constraint では入出力値間の関係を表現している。

4 考察

特徴抽出手順において、特に重要な部分は重み付けの部分である。この重みによってその仕様の機能に関連する式を抽出しなければならず、(1) 式の係数 α、β、γ や各計算での値は、特徴が代表的な機能を表現できるように規定する必要がある。また、入出力値間の関係の抽象化におけるルールの規定も重要であり、詳細なルールの規定だけでなく、適用の順番等も考慮する必要がある。様々な仕様に対して検証を行いこれらの最適な値を設定する必要がある。

5 おわりに

本稿では、部品の機能による検索を目指し、仕様からの特徴抽出手法を提案した。今後は特徴抽出手順における詳細なルールの規定をして、様々な仕様に対して特徴抽出の検証を行い、特徴抽出手順の重み付けの設定や抽象化の設定を規定する必要がある。

参考文献

[1] R.C. Seacord, S.A. Hissam, and K.C Wallnau. Agora a search engine for software components. *IEEE Internet Computing*, pp. 62–70, Nov/Dec 1998.

[2] 中川中, 谷津弘一, 本間毅寛. Cafeobj への誘い. *CafeOBJ HomePage*, 1996.

[3] 足立智隆, 織田健. 形式的なソフトウェア部品検索のための仕様からの特徴抽出. 第 70 回情報処理学会全国大会講演論文集, Vol. 1, pp. 349–350, 2008.

[4] H.P.Edmundson. New methods in automatic abstracting. *Journal of ACM*, Vol. 16, No. 2, pp. 264–285, 1969.