B-016

クラス図によるクラス図構造検索の評価
Case Studies of Searching for Class Diagrams
Evaluation of Searching for Class Diagram by Class Diagram

長谷川 明史† 塚本 享治†
Akifumi Hasegawa Michiharu Tsukamoto

1. はじめに

UML 図はソフトウェアの設計図であり、分析から設計まで広く利用されている。UML 図から典型的な構造を見つけたり、過去に作ったものの再利用したりするとき、設計規模が大きいほど目的の構造を見つけるのは難しい。

そこで、筆者らは[1]で、ソフトウェアの静的な分析設計に利用するクラス図を検索クエリとして扱う手法を提案した。検索対象のクラス図とクエリとして用いるクラス図をそれぞれ RDF と SPARQL に変換して検索を行う。

本稿ではこの手法に従えば、典型的な構造がどの程度検索できるのか、それぞれどのような変換をすれば効率的に検索が行えるかに関して、実験と評価を行う。

2. クエリクラス図の構成

検索対象とするクラス図は、クラスと関連、汎化、依存、実現の 4 つの関係を検索対象として用いる。また、関連に関しては、集約、コンポジションや誘導可能性も併せて扱えるようにする。

クエリに用いるクラス図の記述は通常のクラス図と同様である。このとき、検索対象とする要素名には“?Class”のように疑問符をつけ変数として扱う。

例えば図 1 (a) は“ClassA”との間に関連を持つクラスを検索するためのクエリである。クエリ中の“?AssociationClass”には、(b) の“ClassB”が一致し、一方で“?associationname”には“example”が一致する。

この方法でクラス図の検索を行うと、クラス図の構造が完全に一致する箇所しか見つけられない。is-a 関係などは完全一致だけでは見つけられない。そこで、クエリクラス図にステレオタイプを記述することで、特殊な場合であることを表すことにした。

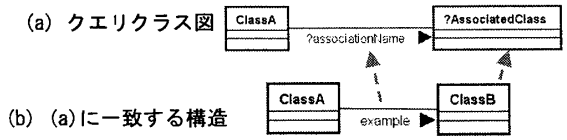

図 1 クエリクラス図(a)と検索対象(b)のマッチ

3. RDF と SPARQL によるシステムの実現

実際に検索対象のクラス図をクエリクラス図で検索するために、まず検索対象のクラス図を RDF に変換する。RDF はトリプルと呼ばれる主語、述語、目的語の組からなるシンプルな構造によって作られる。一方で検索クエリのクラス図を RDF 検索言語である SPARQL に変換を行う。これによって、RDF に変換された検索対象クラス図を検索することができる。

†東京工科大学大学院 バイオ・情報メディア研究科
Tokyo University of Technology Graduate School

次に、完全一致では見つけることのできない関係を検索するため、OWL や SWRL を使った推論規則の記述を行う。ここでは、汎化と実現をどちらも is-a 関係で扱えるようにすることと、関連の種類、誘導可能性が同じ場合に関連を推移させる記述を行った。

4. 検索実験

実現した検索システムを評価するため、実際にクラス図の検索を行った。実験環境は次のとおりである。

- Corei7 CPU 940、Windows XP x64 Edition
- Java 1.6.0_18 (–Xmx10G –Xss64M)
- Pellet 2.1.1

4.1 推論規則のクラス図検索への影響

図 2 は[2]のクラス図である。この図中には 33 のクラス、22 の関連、4 の汎化、6 の実現、7 の依存が記述されている。<<is-a>>や<<transitive>>ステレオタイプによる推論の影響を調べるため、処理するステレオタイプを変えながら図 3 のクエリクラス図を使って検索した。

左のクエリでは、<<is-a>>ステレオタイプを処理しない場合、汎化の検索になるため、4 つのパターンが得られた。一方<<is-a>>を有効にして検索を行うと、汎化と実現を合わせた 10 のパターンを得ることができた。右のクエリは<<is-a>>の推論規則の有無で検索時間は変化しなかった。また、いずれの場合も関連の推移を扱う推論規則を追加した場合、メモリが不足し結果は得られなかった。

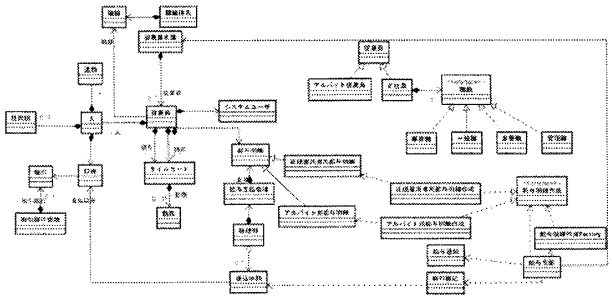

図 2 クラス図[2]

図 3 2 つのクエリクラス図

表 1 それぞれの検索結果とその時間

ステレオタイプ	検索結果	時間	ステレオタイプ	検索結果	時間
なし	4	12秒	なし	3	26秒
is-a	10	12秒	is-a	3	26秒
transitive	メモリ不足		transitive	メモリ不足	
is-a, transitive	メモリ不足		is-a, transitive	メモリ不足	

4.2 ストラテジパターンの検索

図 4 の 2 つのクエリクラス図はストラテジパターンのクエリである。 [3]のクラス図に基づいて作成したが、図 2

の職級クラスに合わせるため、左図はコンポジションを用いた。また、関連による検索への影響も併せて調べるため、右図のクエリも用意した。表 1 の結果から既に transitive で検索できないことは明らかだったため、is-a ステレオタイプのみ有効にした。

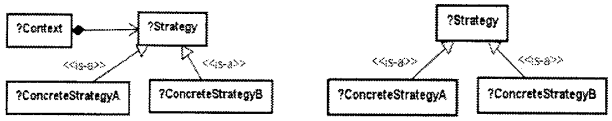

図 4　ストラテジパターン検索クエリ

表 2 ストラテジパターン検索結果

	検索結果	検索時間
左クエリクラス図	28	15分25秒
右クエリクラス図	28	20秒

どちらの場合も、28 通りの検索結果を得ることができたが、ストラテジパターンを意図した部分ではない箇所もある。“?ConcreteStrategyA”と“?ConcreteStreategyB”の両方に同じクラスが一致する場合があった。SPARQL では異なるものは、明示的に記述しなければならないためである。

4.3 コンポジットパターンの検索

図 5 は[3]のコンポジットパターンを参考に、同様の構造を検索するために作ったクエリである。これを同じく[3]のクラス図に対して検索を行った。このときの検索対象クラス図の数は 21 であり、合わせて 103 クラスに対して 5 分 4 秒で検索ができた。

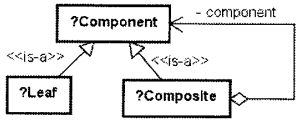

図 5　コンポジットパターンのクエリ

結果として得られたデザインパターンは、クエリの元になった Composite パターン、Interpreter パターン 、Decorator パターンであった。ほぼ同様の構造に、Chain of Responsibility パターンがあったが、集約ではなく関連だったため、結果に現れなかった。

5. 評価、考察

5.1 図による検索に対する評価、考察

クラス図によって直接クラス図を検索することは、探したい構造を直観的に記述できるが、あいまいな部分の記述方法をどのようにするかが問題になる。例えば、コンポジションは集約の特殊な場合、集約は関連の特殊な場合と考えることができる。ストラテジパターンのクエリを書くために参考にした[3]では集約が使われ、一方[2]ではコンポジションが使われていた。このような検索は、細かく区別したい場合と、そうではない場合がある。

完全一致ではない検索を行うために<<is-a>>、関連の推移には<<transitive>>ステレオタイプを用いた。他にも完全一致ではない検索を行えるようにしかければならない。

また、現在のクエリクラス図はクラスとクラスの間の関係を調べることができるが、より大きい単位での検索が必要である。実際にソフトウェアから目的のものを探す場合、あるクラスに注目することだけでなく、どのようなモジュールが組み合わされているのか、という視点で考えることも多い。このような記述の場合にどのようにクラス図で表していくかが課題である。

5.2 検索速度とメモリ使用量の改善

実験によって、小規模なクラス図を対象にした検索の場合は、数十秒で検索結果を得ることができることを確認した。しかし、検索対象が大規模な場合や推論規則の複雑な場合は、実用的ではない時間がかかったり、推論処理に膨大なメモリが必要になったりしてしまう。このため、検索不能な場合もあった。

図 2 に対しての<<transitive>>ステレオタイプを用いた検索結果やストラテジパターンの検索でのコンポジションによる検索時間の違いからわかるように、関連の検索は多くの計算時間とメモリが必要になる。

推論によるメモリ不足の問題は、推論規則が複雑すぎるためと、推移によって大量の関係が導かれているためだと予想できる。表 3 は is-a と関連の推移、それぞれの推論規則に用いた推論条件の数と、それを SWRL で記述した時のトリプルの数である。SWRL で、推論条件となるのはそのトリプルが存在するかどうかである。is-a 関係は条件 3 つなのに比べ、関連の推移は多くのチェックが必要なため、21 の推論条件が使われている。

表 3　SWRL のトリプル数と原子論理式の数

	トリプル数	推論条件
is-a	38	3
関連の推移	149	21

さらに、関連には誘導可能性や関連の種類があるため、クラス図から RDF や SPARQL に変換した時のパターンが、他の関係よりも複雑である。

これらの問題を解決するための方法として、RDF をよりシンプルに作りなおすことができる。例えば、関連の種類や誘導可能性ごとにプロパティを使い分けることによって、1 つの関連に 10 トリプル使っていたものを 4 トリプルに減らすことができる。さらに、クラス図をいくつかのまとまりに分割し、その中であらかじめ推論を行っておく方法もある。今まで推論で求めていた部分を、あらかじめ与えておくことで、推論を簡単に行えると考えられる。

6. おわりに

クラス図をクエリとして用い、検索対象のクラスの検索を行った。探したい構造を直接クエリとして記述することができるが、それは完全に一致するものの検索になってしまう。そこで継承関係の推移などはステレオタイプで指定できるようにした。今後は、is-a や関連の推移と同様に、クエリクラス図に様々な検索条件を指定できるようにしなければならない。また、推論を高速化しなければならない。

参考文献

[1]長谷川 明史,塚本 享治:クラス図をクエリとして用いるクラス図構造検索手法の提案,情報処理学会研究報告, Vol. 2010-SE-169, No.1 ,2010

[2]金澤典子：オブジェクト指向分析／設計教科書,p260,株式会社ソフト・リサーチ・センター,2008

[3]Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides：オブジェクト指向における再利用のためのデザインパターン,ソフトバンククリエイティブ,1999