3L-8

クラス図とシーケンス図からの セマンティック Web 技術を用いたソフトウェアパターン検索

長谷川　明史　　塚本　享治

東京工科大学大学院　バイオ・情報メディア研究科

1. はじめに

UML(Unified Modeling Language)はソフトウェアの設計図として用いられている。この UML をセマンティック Web のデータ構造である RDF に変換することによって、ソフトウェア開発の場面にセマンティック Web 技術を持ちこみ、活用することができる。

本稿では、筆者らが過去にそれぞれ独立して行ったクラス図の検索[1]とシーケンス図の検索[2]の手法を組み合わせ、ソフトウェアパターンなどの典型的な設計に対して検索を行った。

2. アプローチ

2.1. クラス図とシーケンス図の対応関係

シーケンス図はオブジェクト間のメッセージのやり取りを時系列で表現する図であり、クラス図はクラスの関係を静的にあらわす図である。この２つの図は密接に関係している。例えばシーケンス図中のライフラインはクラスのインスタンスと、シーケンス図中のメッセージはクラスが持つ操作と対応する。

これを表したのが図１である。図中のライフラインＡ、ライフラインＢはそれぞれクラスＡ、クラスＢのインスタンスである。また、ライフラインＡからライフラインＢへ送信されるメッセージ operation1 はクラスＢが持つ操作である。

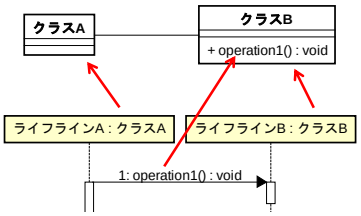

図１　クラス図とシーケンス図の対応

2.2. クラス図の検索とシーケンス図の検索方法

[1][2]ではそれぞれのダイアグラムを RDF という形式に変換した。さらに、RDF 検索クエリである SPARQL を用いることで目的の箇所の検索を行った。

RDF は主語、述語、目的語からなるトリプルを単位としたラベル付き有向グラフの構造である。

[1]ではクラス図の検索に時間がかかりすぎるという問題があり、関連を表す RDF が複雑であるのが原因の一つであった(図２の a)。

Searching Software's Pattern from Class Diagram and Sequence Diagram by Semantic Web Tequnology.
Akifumi HASEGAWA, Michiharu TSUKAMOTO
Tokyo University of Technology, School of Media Science

そこで、本稿ではクラス図中の関連表現を改めた。具体的には、クラス間の関連は uml:associate を述語として用いて表現した(図２の b)。

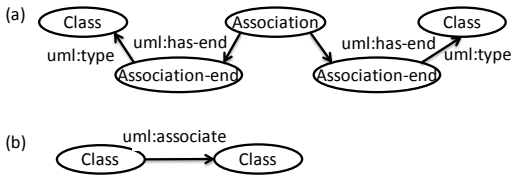

図２　関連の表現

2.3. RDF の統合方法

図３がクラス図とシーケンス図を統合した RDF の構造の例である。点線より上がクラス図の RDF であり、下がシーケンス図の RDF である。ライフラインのオブジェクトが属するクラスを uml:lifelineType という述語で、メッセージに対応するクラスの操作を uml:operation という述語で表す。

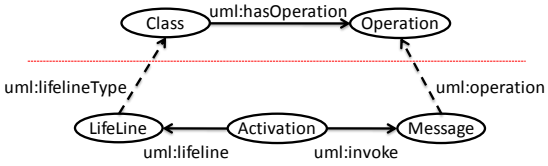

図３　クラス図とシーケンス図の統合

この２つの図を結び付けるトリプルは、シーケンス図を RDF 化する際に付け加える。インスタンスは自分の属するクラスを知っているが、クラスは自身のインスタンスを知っているとは限らない。

3. SPARQL による検索実験

3.1. 検索実験準備

本稿では RDF の検索クエリとして、直接 SPARQL を記述することにした。

表１対象パターン

パターン	クラス数
AbstractFactory	9
AbstractFactory(MazeFactory)	12
ChainOfResponsibility	4
ChainOfResponsibility(Wigit)	6
Composite	4
Composit(Equipment)	8
Decorator	6
Decorator(VisualComponent)	7
State	4
State(TCPState)	6
Strategy	6
Strategy(Composition)	6
InterceptingFilter (CustomFilter:Decorator)	5
InterceptingFilter (CustomFilter:Nodecorator)	8
InterceptingFilter(StanderdFilter)	8
InterceptingFilter(TemplateFilter)	8

表 2RDF のトリプル数

	トリプル数
OWL(語彙定義)	110
クラス図RDF	1063
シーケンス図RDF	1760
推論前	2933
推論後	4856
推論増加	1923

検索対象は、[3]から 6 つのパターン、[4]から Intercepting Filter パターンの４つの戦略を取り

上げ、文献中のクラス図とシーケンス図、ソースコードを参考に作図を行った。これらの詳細は表１、表２に示す。

3.2. Decorator パターンの検索

クラス図とシーケンス図を組み合わせることによる効果を確かめるため、クラス図だけで検索する場合とシーケンス図も併せて検索する場合で比較を行う。リスト１はクラス図の構造だけを検索するクエリであり、リスト２はリスト１にシーケンス図部分の指定を加えたクエリである。リスト１によって 17 件の結果が得られ、そのうちの 7 件がリスト２の結果にあらわれた（表 3、表 4）。

リスト１　クラス図検索のクエリ

```
PREFIX uml:<http://www.teu.ac.jp/g3109018/uml/>
SELECT DISTINCT ?DecoratorName ?ConcreteName
WHERE{
    ?Decorator uml:is-a ?Component.
    ?Decorator uml:associate ?Component.
    ?ConcreteDecoratorA uml:is-a ?Decorator.
    ?Decorator uml:name ?DecoratorName.
    ?ConcreteDecoratorA uml:name ?ConcreteName.
}
```

リスト２　クラス図シーケンス図検索のクエリ

```
PREFIX uml:<http://www.teu.ac.jp/g3109018/uml/>
SELECT DISTINCT ?DecoratorName ?ConcreteName
WHERE{
    ?Decorator uml:is-a ?Component.
    ?Decorator uml:name ?DecoratorName.
    ?Decorator uml:associate ?Component.
    ?ConcreteDecoratorA uml:is-a ?Decorator.
    ?ConcreteDecoratorA uml:name ?ConcreteName.

    ?Component uml:hasOperation ?op.
    ?ClientActivation uml:implicit ?message1.
    ?message1 uml:activate ?DecoratorActivation.
    ?DecoratorActivation uml:implicit ?message2.
    ?DecoratorActivation uml:lifeline ?lifeline.
    ?lifeline uml:lifelineType ?ConcreteDecoratorA.
    ?message1 uml:operation ?op.
    ?message2 uml:operation ?op.
}
```

表３　リスト１の結果

DecoratorName	ConcreteName
AuthenticationFilter	AuthenticationFilter
Handler	Handler
Handler	ConcreteHandler2
Handler	ConcreteHandler1
CompositeEquipment	Cabinet
CompositeEquipment	CpmpositeEquipment
CompositeEquipment	Bus
Composite	Composite
Decorator	Decorator
Decorator	BorderDecorator
Decorator	ScrollDecorator
HelpHandler	HelpHandler
HelpHandler	Widget
HelpHandler	Application
Decorator	ConcreteDecoratorB
Decorator	ConcreteDecoratorA
DebuggingFilter	DebuggingFilter

表４　リスト２の結果

DecoratorName	ConcreteName
AuthenticationFilter	AuthenticationFilter
Handler	ConcreteHandler1
CompositeEquipment	Chassis
CompositeEquipment	Bus
Decorator	ScrollDecorator
Decorator	ConcreteDecoratorB
DebuggingFilter	DebuggingFilter

表 4 の Authentication Filter と Debugging Filter は Intercepting Filter パターンのカスタムフィルタの Decorator 実装、Decorator は GoF の Decorator パターンのものである。Handler は Chain Of Responsibility パターンのものであり、Composite Equipment は Composite パターンのモデルから見つかった。

3.3. Strategy パターンと State パターンの検索

次に Strategy パターンの検索を行う。この 2 つのパターンの目的は異なっているが構造も振舞いも共通であるため、クエリを区別することができない。したがって Strategy パターンを検索するために記述したクエリ（リスト３）は、そのまま State パターンの構造と振舞いを検索するためのクエリになる。

リスト３　ストラテジーパターンのクエリ

```
PREFIX uml:<http://www.teu.ac.jp/g3109018/uml/>
SELECT DISTINCT ?StrategyName ?ConcreteName
WHERE{
    ?Client uml:associate ?Strategy.
    ?ConcreteStrategy uml:is-a ?Strategy.
    ?Strategy uml:hasOperation ?AlgorithmInterface.

    ?Aclivation uml:invoke ?message.
    ?Aclivation uml:lifeline ?lifeline.
    ?lifeline uml:lifelineType ?Client.
    ?message uml:operation ?AlgorithmInterface.
    ?ConcreteStrategy uml:name ?ConcreteName.
    ?Strategy uml:name ?StrategyName.
}
```

Strategy パターンや State パターンはそれ単体では非常にシンプルであるため、構造と振舞いが一致してしまう部分が多く、63 か所の一致が見られた。

4. 考察

Composite パターンと Decorator パターン、Strategy パターンと State パターンは構造も振舞いも類似している。しかし、これらが異なるパターンとして分類されるのは、その構造や振舞いを使う目的が違うためである。例えば、decorator は component を１つしか持たない Composite パターンとみなすことができるが、オブジェクトの集約を目的としているわけではないと[3]で述べられている。

これらパターンの目的や意図の違いは、設計者が UML 図にその意図を記述することで扱うことができるようになる。例えば、Composite は集約が目的だが、Decorator は機能追加を目的として component に委譲を行う。このような設計の意図を rdfs:subPropertyOf などを用いて、uml:associate の下により意味のあるプロパティを定義することで、RDF 中に設計の意図を入れたモデルを構築することができる。

5. おわりに

クラス図とシーケンス図をそれぞれ RDF グラフに変換し、クラスとライフライン、操作とメッセージを対応付けることによって、構造と振舞いの 2 つの視点からの検索を行うことができた。しかし、ソフトウェアには構造も振舞いも近いが意図が異なる場合があるため、uml:associate で表されている「関連がある」という意味よりも、踏み込んだ関係を利用した検索が行えるように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]長谷川明史，塚本享治，“シーケンス図をクエリとして用いるシーケンス図の振舞い検索の提案”，信学技報，KBSE2010-43，2011

[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

[4]Deepak Alur，John Crupi，Dan Malks： J2EE パターン，日経 BP 社,p156-171,2005