セマンティックウェブ構築のためのハイブリッドモデル型 OWLオントロジープロセッサ

野崎広志†

日本アイ・ビー・エム(株) ソフトウエア開発研究所†

1．はじめに

セマンティックウェブ(以下、SW と略記)はウェブに広がる膨大な情報を人間以外にも機械が解釈し利用可能にする次世代インターネット技術として Tim Berners-Lee[1]の提唱以来、W3C の推進も手伝い注目を浴びている。

1.1 オントロジーとオントロジープロセッサ

オントロジーはそうした流れの中でメタデータとしてセマンティックスを表現する役割を担い、コンセプト(ボキャブラリ)、その間の関係や制約を体系化して形式的に表現したもので、今後の知的な機械処理実現のキーテクノロジーとなりつつある。オントロジー記述言語は、主に RDF(Resource Description Framework)とそのスキーマ RDFS (RDF Schema)[2] ,DAML(DARPA Agent Markup Language)+OIL(Ontology Inference Layer)[3][7],OWL(Web Ontology Language)[4]と発展してきており、オントロジープロセッサ(以下、OP と略記)にはこれらオントロジーとアプリケーション(以下、App と略記)の間に介在した様々なレベルでのサービスの提供が求められる(以下、OWL 処理を中心とした OP を OWLP と略)。

1.2 関連研究

こうした観点から、処理レベルの違いはあるが表１に示すオープンソースも OP として分類できるであろう。

表1. オントロジープロセッサ関連のオープンソースの比較

	操作 API	推論	Query し易さ	OWL 特化処理	日本語	XML 標準技術
ARP[8]	×	×	×	△	△	△
F-OWL [5]	△	○	◎	○	?	×
JTP[6]	△	○	◎	△(DAML)	×	×
Jena1.6[9]	○	×	○	△(DAML)	△	×
Jena2 [10]	◎	○	○	○	△	×
本モデル	◎	○	◎	○	○	○
OWL モデル	◎	×	○	○	○	×
RDF モデル	○	△	○	○	○	×
Stmt モデル	○	○	◎	○	○	×
DOM モデル	○	×	○	×	○	○

1.3 従来の問題点

表１に、操作 API の使い易さやクエリの容易さ等の観点からの主観的評価を示すが、大まかな問題点として：

1)F-OWL[5]や JTP[6]は推論ルールの変更で OWL や DAML に対応可能であるがステートメント(言明)単位の操作 API が貧弱。

2)Jena1.6/Jena2[10]は必要な機能が揃っており API も豊富であるが試行錯誤的なクエリがしにくく、App の Rapid Prototyping 的用途には不向きと思われる。

3) Wonder Web[11]でのような OWL オブジェクト操作の API だけではデータ操作は容易だが、推論的機能を必要に応じ実装する必要があり、複雑な知的処理には不向きである.

2．ハイブリッドモデル型 OWL プロセッサ

SW はまだ有効性の模索段階であり、現段階では試行錯誤的に様々なアイデアを試せる１パッケージが望ましいと考え、既存技術にある１つのモデルだけではなく、App から見た場合、オブジェクトモデル(OWL モデル)、グラフモデル(RDF モデル)、ステートメントモデル(Stmt モデル)等を用途に応じて同時に扱える構成とし(①～④の API)、全体として前述の問題点を克服するよう、RDF パーザーや推論エンジンを部品として組合せ、OWLP を実装した。図１にその構成を示す。

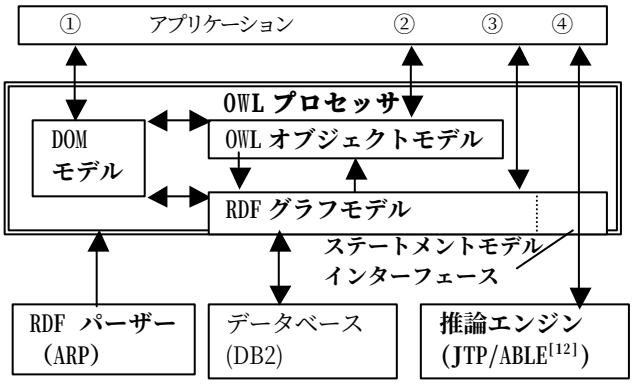

図1. 本 OWL プロセッサの構成図

3．各モデルとその特徴

3.1 OWL オブジェクトモデル(OWL モデル)

OWL 言語要素(allValuesFrom 等)をオブジェクト化しクラス、プロパティやインスタンスとして扱い可能にしたモデルでオブジェクト指向的な操作 API（図１の②の OWL API）を提供する。高級言語的な処理が可能。

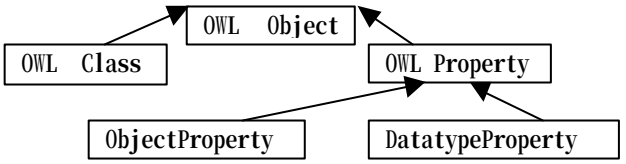

図2. OWL オブジェクトモデル模式図（実装クラス階層の一部）

3.2 RDF グラフモデル(RDF モデル)

主語、述語、目的語の３つ組(ステートメント)をグラフのノードとアークで表現するモデルである。OWL や DAML のオントロジーも RDF グラフに還元されるので、基本モデルでもある。実装ではノードとステートメントの対応関係も保持するよう工夫しており、今後の柔軟な処理を考慮したモデルである。図１の③の API(RDF API)でアクセスする。

3.3 ステートメントモデル（Stmt モデル）

ステートメント単位でしか操作ができないモデルで、推論

Hybrid OWL Ontology Processor composed of several models for feasibility study of Semantic Web Applications

†Hiroshi Nozaki, Yamato Software Dev. Laboratory, IBM Japan

エンジンや DB エンジンとのインターフェースをとるモデル。グラフモデルの１種でもあるが別として扱う。図１の④の API で推論エンジンへアクセス。試行錯誤的なクエリの繰り返しには、このモデルが最適である。

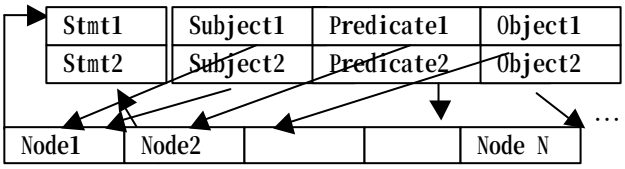

図3. RDF グラフモデル(上部はステートメントモデル)模式図

3.4 DOM モデル（意味表現としての DOM 木）

DOM[13]は XML 文書の DOM 木をアクセスする標準 API として広く普及しているが、本研究ではステートメントデータのコレクションとして DOM 木を構成することで、App の標準的な XML 関連手法によるデータアクセスを可能にするために DOM モデルも提供している。オントロジーの記述には同じ意味でも異なる記述の仕方ができるが、パージングに DOM を使用するのではなく、意味表現としてのステートメントから DOM 木を構築することで記述の違いを吸収し、同じ意味表現を App に提供可能にする。また XML の標準的手法が利用可能。

４．本 OWL プロセッサの実行例

実装した OWLP の有効性を示すために、クエリを例にとり、その実行例を示す(図４)。OWL モデルでオントロジーを読み込み、Stmt モデルで KIF[14]ライクなクエリを発行し、owl:disjointWith による推論を行なっている例である。Stmt モデルでは、図５に示すような推論ルールの修正やクエリの変更がパターン記述風に可能で、更に OWL モデルで簡易なオブジェクト操作が可能である。実装した OWLP は、クエリ言語はサポートしていないが、表２のように各モデルに応じたクエリ形式をサポートし、クエリを分かりやすく容易なものにしている。なお、本研究では、JTP による述語を変数にしてのクエリも可能にするために holds をクエリ(表２)や推論規則(図５)に付加している。

５．おわりに

各モデルそれぞれ OWLP の実現は可能であるが、これらを複合させて同時に利用出来るハイブリッド型の OWL プロセッサを作成し実行結果を例に有効性を示した。用途に応じた異なるモデルの同時利用も可能なため幅広い App に対応可能である。しかし、内部的には各モデルでのデータの変更に伴うモデル間の整合性維持や各モデルでの機能の充実(特に推論)や連携処理等が今後の課題である。

謝辞

本研究は２００３年度総務省の委託研究『ＩＴを活用した企業間連携・起業支援プラットフォームに係わる要素技術に関する調査研究』の一部をより発展させたものである。研究の機会を与えて戴き、ここに感謝の意を表する。

参考文献

[1]Tim Berners-Lee, ”Semantic Web Road map ” (http://www.w3.org/DesignIssues/Semantic.html)
[2]RDF Schemas (http://www.w3.org/TR/rdf-schema/)
[3]DAML+OIL,(http://www.w3.org/TR/daml+oil-reference)
[4]OWL (http://www.w3.org/TR/owl-ref/)
[5]F-OWL: An OWL Inference Engine in Flora-2,Department of Computer Science & Electrical engineering, UMBC
[6]JTP: An Object-Oriented Modular Reasoning System,Knowledge Systems Laboratory ,Stanford University
[7]DAML DARPA Agent Markup Language (DAML) , (http://www.daml.org/)
[8]ARP: Another RDF Parser (http://www-uk.hpl.hp.com/people/jjc/arp/)
[9]Jena, HP Labs Semantic Web Research, (http://www.hpl.hp.com/semweb/index.html)
[10]Jena2 , (http://www.hpl.hp.com/semweb/jena2.htm)
[11]WonderWeb:Ontology Infrastructure for the Semantic Web(http://wonderweb.semanticweb.org/owl/)
[12]Agent Building and Learning Environment(ABLE), (http://www.alphaworks.ibm.com/tech/able)
[13]Document Object Model(DOM)
[14]Knowledge Interchange Format(KIF), (http://logic.stanford.edu/kif/dpans.html)

表2. 各モデルとQuery の形式

	推論	例	主モデル
KIF 形式	○	(holds rdf:type ?x Animal)	Stmt モデル
RDQL 風パターン記述API	×	QGraph QG;… QG.setSELECT(“?x ”); QG.addWHERE(“(?x rdf:type Animal) ”); QG.execQueryPath();	RDF モデル
OWL Query パターン	×	<owl:Class rdf:nodeID="_VarX"> <rdfs:subClassOf rdf:resource="#Animal" /> </owl:Class>	OWL/RDF モデル(nodeID と匿名ノードパターン)
DOM	×	XQuery 等の標準的手法(略)	DOM モデル

```
<owl:Class rdf:ID="動物" />
<owl:Class rdf:ID="人間">
  <rdfs:subClassOf  rdf:resource="#動物" />
</owl:Class>
<owl:Class rdf:ID="男性">
  <rdfs:subClassOf  rdf:resource="#人間" />
  <owl:disjointWith  rdf:resource="#女性"/>
</owl:Class>
<owl:Class rdf:ID="女性">
  <rdfs:subClassOf  rdf:resource="#人間" />
  <owl:disjointWith  rdf:resource="#男性"/>
</owl:Class>
<人間  rdf:ID="太郎" >
  <owl:complementOf rdf:resource="#男性"/>
</人間>
```

```
Query：
(holds rdf:type 太郎 ?x)
```

```
Answer:
 ?x:  人間
 ?x:  動物
 ?x:  女性
```

図４サンプルオントロジー(一部)とQuery の結果

```
(=> (and (holds rdfs:subClassOf ?csub ?csuper) (holds rdf:type ?x ?csub)) (holds rdf:type ?x ?csuper))
(=> (and (holds owl:disjointWith ?c1 ?c2) (holds rdf:type ?c1 owl:Class) (holds rdf:type ?c2 owl:Class) (holds owl:complementOf ?x1 ?c1))
```

図５ 推論ルール(１番目は、下位クラスのインスタンスは上位クラスのインスタンスでもあることを示す前向き推論規則)