

セマンティック Web とオントロジー

～ マンリーダブルからマシンリーダブルへ～

山元 佑輝, 佐野 僚

Yuki Yamamoto, Ryo Sano

1 はじめに

近年, インターネットを利用した正確な情報の入手が求められており, その技術の一つとして Semantic Web がある. Semantic Web をグローバルで誰もが利用可能なものにするための技術に Ontology がある. 本発表では Ontology について述べる.

2 Semantic Web

Semantic Web とは, 「URI で識別できるメタデータをエージェントが自動処理するウェブ」である. HTML ベースのウェブが「人間が読んで理解する」に主眼を置いていたのに対し, Semantic Web は「マシン同士が理解可能なウェブ」を目指し, 段階的に技術の開発が進んでいる (Fig.1). 現在では Ontology 階層まで開発が進んでいる.

レイヤー	役割
Trust	コンテキスト, プルーフ, 暗号化と電子署名により, エージェントが示した結果の信頼性を判断
Proof	エージェントの処理の履歴, 処理理由など, 結果を導いた根拠を示す
Logic	一階述語論理などを用いた知識の記述と, それに基づくエージェントの処理
Rule	問い合わせ, フィルタリングを可能にする共通基盤としての論理の定義
Ontology	より精密な語彙の定義と, 複数のスキーマの関係づけ・融合を可能にする推論
RDF Schema	語彙 (クラス, プロパティ) を定義する手段の提供
RDF MS	機械処理可能なメタデータの表現 (データモデル)
XML/namespace	処理が容易な記述言語 (XML) と複数語彙の区別・混在を可能にするメカニズム (名前空間)
URI/Unicode	リソースのグローバルな識別 (URI) とグローバルなデータ表現 (Unicode)

Fig. 1 Semantic Web の技術レイヤー

3 Ontology

3.1 概要

Ontology は, Heidegger らのいう「存在論」を指す哲学用語として知られている. しかし, その意味が拡張されて, 対象とする世界に存在するものごとを体系的に分類し, その関係を記述するものとして, 言語学や人工知能研究に取り込まれてきた. ウェブでの Ontology とは,

Berners-Lee らによると, 「分類体系」と「推論ルール集」である.

3.2 OWL (Web Ontology Language)

Ontology の概念が用いられた Web 言語が OWL (Web Ontology Language) である. OWL は, 共有性, 発展性, 相互運用性, 矛盾の検出といった要件を満たすものとして, W3C のワーキンググループで開発されている. つまり, 「用語・語彙とそこに含まれる各要素の関連の明確な表現」を目的としている.

3.3 Web Ontology の特徴

- 語彙セットと推論機構
- 階層関係, 部分全体関係
- 多重継承, 時間順序の記述, 異なる文書のインターフェイス
- 一貫性チェックの制約条件, クラスとインスタンス
- サービスの発見, 複数の Ontology の変換と統合利用

推論というと複雑なイメージが湧くが, OWL での推論とは, 同一のものか, どちらかに含まれるものか, 互いに素であるか, を調べるに過ぎず, これらは数学的に処理が可能である.

3.4 OWL の基本構成

OWL は, 語彙を定義する手段である RDF の集合で構成されている. OWL 言語仕様では, どんな RDF が OWL の語彙を構成し, それによって何を意味するかを定義する. OWL は一般に RDF の XML 構文によって記述され, 次のような構成要素を含む.

1. バージョン情報と他のオントロジーのインポートを記述するヘッダ要素
2. オブジェクトのクラスを定義するクラス要素
3. プロパティを定義するプロパティ要素
4. クラスおよびプロパティのインスタンス

OWL の仕様はまだ草案段階で不確定要素が多く, 今後変更される可能性がある. そのことを踏まえた上で, OWL の一例 (Fig.2) を示す. Fig.2 は「雄」と

いうクラスは「動物」のサブクラスで「雌」とは互いに素である」という事を定義している。rdfs:subClassOf, owl:disjointWith は OWL で定義されているクラス要素である。rdf:resource="# ~"の部分は RDF での定義である。

```
< owl:Class rdf:ID="Male" >
< rdfs:subClassOf rdf:resource="#Animal" />
< owl:disjointWith rdf:resource="#Female" />
</owl:Class >
```

Fig. 2 OWL の一例

3.5 問題点

普通のユーザにメタデータを記述することは困難である。そのためメタデータを記述するためのツールの普及が問題となるが、そのためにはデータベース、正確なメタデータが提供される必要がある。どのようにして、コンテンツ作者の負担にならない形でメタデータを容易に確実に記述するか、といった問題がある。

3.6 XSLT

一つの方法として XSLT を使うというものがある。この方法のポイントは、XSLT を定義すれば、XHTML を書くだけでメタデータが提供できる。コンテンツに手を加えた時に、別途メタデータを編集する必要もない。また、これらの class 属性は、CSS と組み合わせてページの視覚表現にも使えるので一石二鳥である。もちろん、XSLT は HTML には適用できないので、コンテンツは XHTML で書かれている必要がある。XHTML への移行のメリットが見えないとよく言われるが、Semantic Web への第一歩は、XHTML 化によって踏み出すことができる。

しかし、この方法もユーザ依存である。なぜなら、一般ユーザが XHTML を使うことを前提にした話だからだ。一般ユーザの中であまり浸透していない XHTML が、どこまで広がっていくかが、これからの鍵になる。

次に一例を示す。次のような XHTML (Fig.3) があつたとする。XSLT のテンプレート (Fig.4) を適用してみる。この結果メタデータ (Fig.5) が得られた。

```
< p id="c01" class="Concert" >
< span class="title" >
シュトゥットガルト放送響来日公演
</span >が
< span class="date" > 2001-11-16 </span >
に行われた。
</p >
```

Fig. 3 XHTML

```
< xsl:template match="h:p" >
< xsl:element name="wn:@class" >
< xsl:attribute name="rdf:ID" >
< xsl:value-of select="@id" />
</xsl:attribute >
< xsl:apply-templates select="h:span[@class]" />
</xsl:element >
</xsl:template >
- - - - -
< xsl:template match="h:span[@class]" >
< xsl:element name="dc:@class" >
< xsl:value-of select="." />
</xsl:element >
</xsl:template >
```

Fig. 4 XSLT のテンプレート

```
< wn:Concert rdf:ID="c01" >
< dc:title >
シュトゥットガルト放送響来日公演
</dc:title >
< dc:date > 2001-11-16 </dc:date >
</wn:Concert >
```

Fig. 5 検出されたメタデータ

4 終わりに

「Semantic Web の実現は不可能である」という意見がある。Semantic Web の目的は、「意味を標準化する」ではなく、「意味を表現する概念やプロパティに一意の識別子 (URI) を与えて、計算機が確実に認識・処理できるようにする」を目指すことである。

現在、HTML 主流だが、完全無欠な世界ではなく、Semantic Web も同様である。また、一度に全てが完成するようなものでもない。ただ不可能と決め付けるだけではなく、実現へむけての新たな問題提起をすることが重要である。

Semantic Web が普及すれば、個々の Web がデータベースの一部となることができる。その結果、広範囲における正確な情報検索が可能となり、人間の負担を軽減することができるだろう。

参考文献

- 1) メタ情報とセマンティック・ウェブ,
http://www.kanzaki.com/docs/sw/
- 2) The Web Ontology Language (OWL),
http://lists.w3.org/Archives/Public/www-webont-wg/2002Jan/att-0061/01-swol.text