

## 特集3

# セマンティック Web と オントロジ記述言語

清野 正樹<sup>†1</sup> 松下電器産業 (株)

来間 啓伸<sup>†2</sup> (株) 日立製作所

今村 誠<sup>†3</sup> 三菱電機 (株)

## ■セマンティック Webの枠組とオントロジ

オントロジ (ontology) は、元来は存在論と呼ばれ、アリストテレス以来、事物の存在の意義付けを議論する哲学的な研究領域であったが、近年の知識工学や自然言語処理などの技術分野においては、それぞれの知識 (あるいは、語彙、概念等) が、知識全体の体系の中で、どこに位置付けられるかを明らかにする研究分野ということが出来る。W3C (World Wide Web Consortium) が提案するセマンティック Web の場合、その階層構造の中で、オントロジ層を XML 層と RDF 層の上に位置付けている (本特集「セマンティック Web とは」の図-1 参照)。

人間が Web ドキュメントを閲覧することだけを目的とした HTML と比べ、XML はドキュメントの構造とタグを自由に規定することにより、表現の柔軟性や拡張性を持たせることができる。また、RDF (Resource Description Framework) は XML ドキュメントの特性を

メタデータとして自由に記述することができる。したがって、Web 上の個々の記述は、一般には表現方法や使用する語彙 (vocabulary) が異なるため、以下の図-1 に示すように、オントロジによって概念間の相互関係を定義しておくことによって、はじめて、それぞれの記述の意味的な関係付け (たとえば、上下関係、同義関係等) を行うことができる。

本稿では、はじめに、こうしたセマンティック Web におけるオントロジの役割について電子商取引を例に挙げてさらに詳しく説明したのち、W3C のセマンティック Web 活動の一部としてスタートした Web-Ontology Working Group における、新たなオントロジ記述言語 OWL (Ontology Web Language) の策定に関する最新状況を報告する。

## ■オントロジの役割

### XML ドキュメント交換の課題

XML を用いた電子商取引では、取引の自動化を促進するために、企業間で交換される XML 文書中のタグ名やタグの内容のデータ型などを標準化する必要がある。なぜなら、商品カタログをメーカー横断で検索したり、受発注伝票を交換したりするためには、商品の分類体系や、各々の商品分類カテゴリごとに商品属性を規定する辞書を企業間で共有する必要があるからである。実際、たとえば、電子機器部品の業界では、まだ XML が W3C の勧告になる以前から、SGML (Standard Generalized Markup Language) 形式の商品カタログを作る

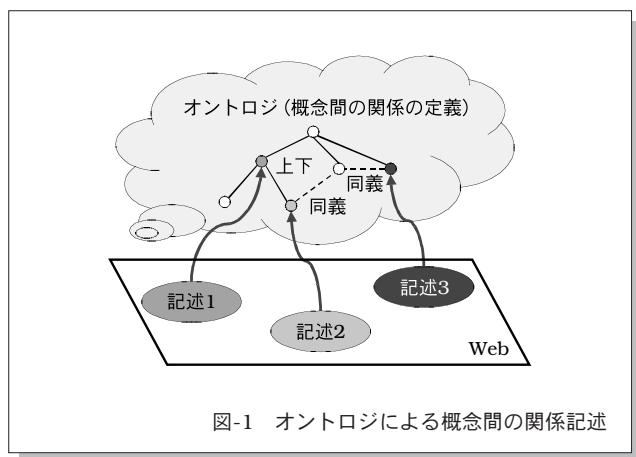


図-1 オントロジによる概念間の関係記述

<sup>†1</sup> E-mail:kiyono@trl.mei.co.jp

<sup>†2</sup> E-mail:kuruma@sdl.hitachi.co.jp

<sup>†3</sup> E-mail:imamura@isl.melco.co.jp

ための辞書整備が進められており、パソコンなどの情報機器と電子部品（半導体）のサプライチェーンの効率化をめざすRosettaNet (<http://www.rosettanet.org/>)に受け継がれている。この辞書では、「周波数シンセサイザやミキサは、チューナ的一种である」といった商品の分類体系や、「チューナは、入力オフセット電圧や電源電圧変動除去比という属性を持つ」といった知識が記述されている。

しかし、オープンで自由なインターネットの世界では、同一業界内でいくつかのコンソーシアムが組織されて、複数の辞書標準ができることがある。また、たとえある業界内での標準化が達成されたとしても、異なる業界では、同じ概念を別の辞書体系で表現されることはどうしても避けられない。そこで、現実のインターネットの世界では、異なる辞書体系を持つもの同士での情報交換を実現する仕組みが必要になってくる。

ところが、現在のXMLの枠組みだけでは、複数の辞書体系が並存する世界をうまく扱うことはなかなか難しい。というのは、XMLを用いた商品カタログ記述では、入れ子構造の深さ、タグの名称、および属性の使用法の違いにより、情報の表現方法に任意性があるので(図-2に同じ半導体部品仕様の2つの表現例を示す)、同じ意味の情報をいくつかの異なった形式で記述されたとしても、機械処理によりその同一性を認識することは難しいからである。そこで、異なった形式のXML文書を相互に活用できるようにするXML変換機能が必要となるが、現在のXMLの枠組では、DTD(Document Type Definition)ごとに個別に変換プログラムを開発するしかないという問題点があった。

### オントロジへの期待

このXML変換プログラム開発を容易にするための要素技術として期待されているのが、オントロジである。セマンティックWebでは、異なる辞書体系を持つXML文書間の変換が容易になるように、概念間の階層関係や概念定義間の整合性などを自動計算できるようなオントロジ記述言語を提供しようとしている(図-3参照)。また、オントロジ記述言語の提供により、オントロジの機械処理が促進されるので、複数Webサイトのコンテンツを連携させるような自動サービスの構築が容易になるという効果も期待されている。

さて、オントロジにはいくつかの定義があるが、Fenselらは、彼らのOn-To-Knowledgeプロジェクト(<http://www.ontoknowledge.org/>)での主張に合致したオントロジの定義として、Gruberの定義「共有される概念化(conceptualization)の形式的(formal)かつ明示

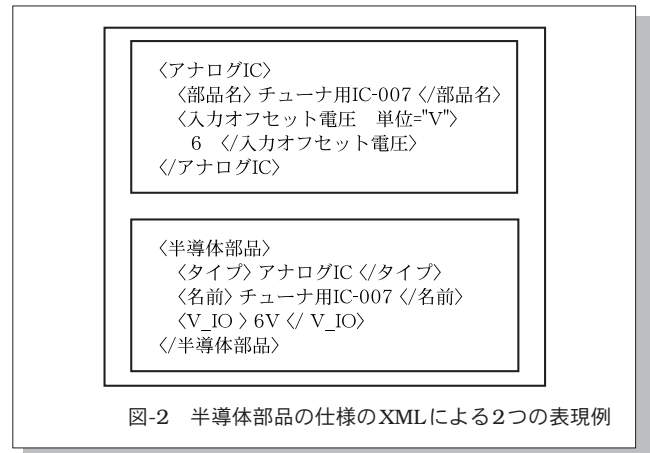


図-2 半導体部品の仕様のXMLによる2つの表現例

的(explicit)な仕様]をとりあげている<sup>1)</sup>。ここで、概念化とは、対象とする現象の抽象的なモデルのことであり、現象中で興味を持つ概念と、それらの概念間の関係が表現される。形式的とは、オントロジは機械に理解されるものでなければならないということである。そして、明示的とは、概念のタイプと、概念間の制約関係が明示的に定義されていなければならないということである。

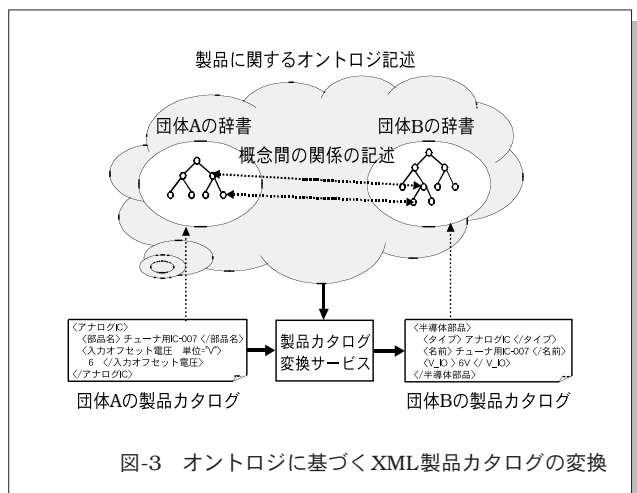
現状のXMLを用いた電子商取引における辞書も一種のオントロジといえるが、異なる辞書体系を持つ企業間の情報交換を促進するには、XML変換プログラムの開発生産性を向上させるために、Gruberの定義でいう「形式性」や「明示性」を高めていくことが重要になってくる。そこで、セマンティックWebでは、数学的な基盤を持ったオントロジ記述言語を提供しようとしている。

### ■オントロジ記述言語に対する要求事項

#### WebオントロジWGの活動

W3Cでは、セマンティックWeb活動の中で、メリーランド大学のHendlerを中心に、新しいオントロジ記述言語の構築を目的とした作業グループWeb-Ontology Working Group<sup>2)</sup>(以後、WOWGと記載)を2001年11月にスタートさせた。WOWGでは、セマンティックWebおよびオントロジのユースケースの分析に基づき、新しいオントロジ記述言語OWLに対する要求事項を2002年3月8日にWorking Draft化<sup>3)</sup>するとともに、これを満足するOWLの言語仕様を現在検討中である。

オントロジ記述言語には、WOWGの活動以前から、DAML+OIL<sup>4)</sup>をはじめとして、いくつかの提案がなされている。DAML+OILは、HendlerらによるDARPAのDAML(DARPA Agent Markup Language)プロジェ



クト (<http://www.daml.org/>) で開発されていたオントロジ記述言語 DAML-ONT と、On-to-Knowledge プロジェクトで開発されていた OIL (Ontology Inference Layer) を統合した言語で、現時点 (2002/5/20) での最新版は、2001年3月の Ver.4.2 である。DAML+OIL は、WOWG においても、OWL 策定の議論の出発点と位置付けられている。

### Webオントロジのユースケース

WOWG では Web オントロジの代表的なユースケースとして次の6つの事例を挙げている。

#### (1) Webポータル

Webポータルは特定のトピックに関する情報を集めた Web サイトで、単純な情報インデックスとキーワード検索によって構成されているケースが多い。オントロジは、こうしたサイトに対し、情報を探索するための語彙セットと、推論機構を提供することができる。たとえば、“学术论文は、1人以上の著者(すなわち“people”)によって書かれ、“people”は氏名と所属(すなわち“organization”)を持つ”といった知識をオントロジに記述しておけば、推論機構を使って知的な文献検索を行うことができる。

#### (2) マルチメディアコレクション

オントロジは、映像、画像、写真などのマルチメディア情報のコレクション(集合体)において、各情報に意味的なアノテーションを付与するのに利用することができる。こうしたオントロジには、次のような記述力が求められる。

- 階層関係(検索語の抽象化・具象化に利用)
- 部分全体関係(検索語の展開に利用)
- 定義的な知識の記述(たとえば、アンティークな家具に対し“Late Georgian”といえ、1760年から1811年の間に作られた英国の家具であるといった知識)

- デフォルト知識の記述(たとえば、“Late Georgianの整理だんす”は通常マホガニー製であるといった知識。常に真とは限らない)

#### (3) 企業のWebサイトマネジメント

大企業の Web サイトには、会社説明、プレスリリース、製品情報、経営情報、仕様書など大量のページが含まれている。こうした Web サイトに対して、オントロジは情報のインデキシングと効率的な検索手段を提供することができる。その際、オントロジには、次のような記述力が求められる。

- 階層構造における多重継承 (Multiple Inheritance)
- 部分全体関係(たとえば、“このプロジェクトは次のサブプロジェクトから構成される”)
- 時間的な順序関係(たとえば、“プロジェクトAはプロジェクトBの後に実施される”)
- 企業活動における他のXMLドキュメントとのインタフェース
- 言語ニュートラルな表現(グローバル企業にとって、英語での検索への対応)

#### (4) 設計情報のドキュメンテーション

製造分野では、製品の設計情報のドキュメンテーションが、セマンティック Web およびオントロジの適切なユースケースと考えられている。たとえば、航空機分野では、機体の設計ドキュメント、製造工程ドキュメント、テスト工程ドキュメント、メンテナンスドキュメントなど相互に関連する多数のドキュメントから構成される。こうしたオントロジには、次のような記述力が求められる。

- 一貫性チェックのための制約条件の記述(たとえば、複葉機 (biplane) の翼の数は2であるといった、以下のような記述)

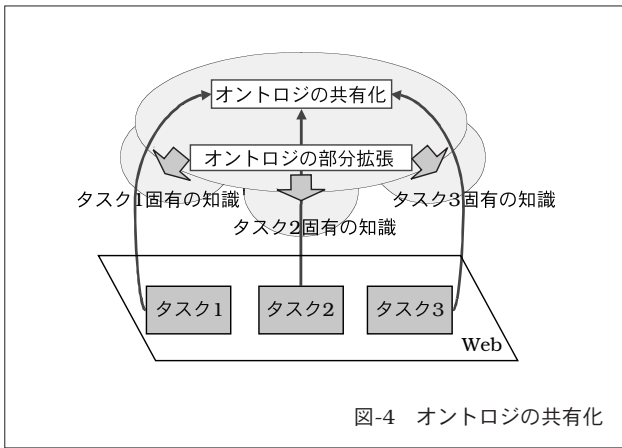
(aircraft.type = biplane) =>

(CardinalityOf (InstancesOf (Class = Wing))) = 2)

- 言語ニュートラルな表現
- クラスとインスタンスの区別(たとえば、一般的な“供給業者 (supplier)”という表現と個々の企業との区別)
- N項関係の記述
- 他のXMLドキュメントとのインタフェース

#### (5) 知的エージェント

セマンティック Web は多様な情報リソースを統合利用する知的エージェントにも利用することができる。特に、利用者の特性や嗜好に基づいて情報を選択するタイプのエージェントにとっては、対象ドメインの知識を記述したオントロジが不可欠である。こうしたオントロジは、次のような課題を含んでいる。



- オントロジの場所の特定
- 複数の独立したオントロジの統合利用
- ドメインやサービスごとに異なるオントロジ間での変換や相互参照
- 記述と利用を容易にする単純な表現形式

#### (6) ユビキタスコンピューティング

ユビキタスコンピューティングは、モバイル機器を使い、いつでもどこでもネットワークサービスを利用できる環境である。その技術的な課題は、任意の場所で、必要なネットワークサービスに ad hoc に接続することにあるが、現状では、場所に対して固定的なサービスしか提供できない。これに対し、オントロジは、ユビキタス環境下であっても、その場所からアクセス可能なサービスの中から、ad hoc に適切なサービスの発見 (discovery) を可能にしてくれる。

#### オントロジ記述言語に対する要求事項

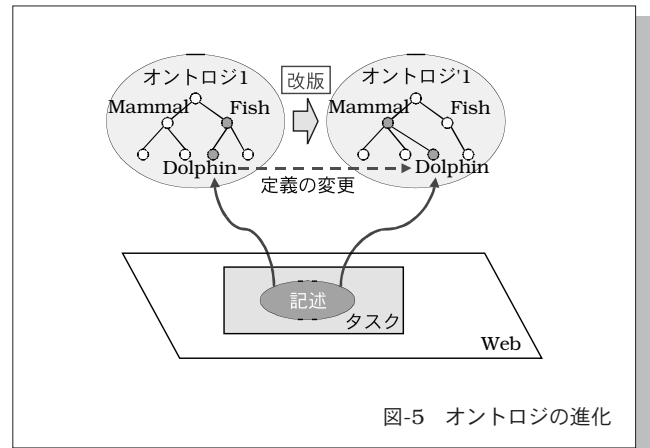
本節では、WOWG が提示した、OWL 1.0 の言語仕様が目指すべき8つの設計ゴールについて説明する。

##### (1) オントロジの共有化 (Shared ontologies)

分散したデータソースが共通の用語を使用するようなタスクにおいて、オントロジは、異なるデータソースに共通の意味付けを与えるために、共有化されなければならない。また、既存の知識だけでは不十分なことが多いため、オントロジは、個々のタスクに必要な新しい定義を追加できるよう、拡張性を持たなければならない (図-4 参照)。

実現のアプローチとしては、XML Schema と RDF だけでは、用語の意味的定義を提供することができないため、オントロジ記述言語で次の情報を規定する。

- 1) オントロジを定義するための表現形式
- 2) ドキュメントを1つあるいはそれ以上のオントロジに関連付けるための表現形式
- 3) 2つ以上のオントロジが同じ用語を含む場合に、どの



オントロジを利用するかの曖昧性を解消するための表現形式

なお、DAML+OIL では、オントロジ全体の特性を記述する `daml:Ontology` エレメントの中で、`daml:imports` によって、用語が関連付けられているオントロジを URI で指定する。

##### (2) オントロジの進化 (Ontology evolution)

オントロジが変化する可能性のあるタスクに対し、オントロジは改版ができなければならない。オントロジが変化するのは、以前の版にエラーがあったり、ドメインをモデル化する新しい手法が導入されたり、あるいは、現実が変化してしまう場合である。オントロジに改版で定義を変更した場合には、同じ記述であっても、別の意味付けが与えられる (図-5 参照)。

改版における重要な問題は、オントロジの1つの版に關係付けられたドキュメントが、別の版に關係付けられたドキュメントと互換かどうかである。互換あるいは非互換のいずれの改版も許されるべきであるが、この2つは区別されなければならない。

RDF Schema では、スキーマの版をスキーマ自身の URI とは独立したリソースとして取り扱うことを推奨している。すなわち、クラスとプロパティの階層関係を記述するための `rdfs:subClassOf` と `rdfs:subPropertyOf` を、クラスとプロパティの新しい版を以前の版に關係付けるのに使用することができる。しかしながら、この方法は、正しくない定義を取り消すことができないという欠点を持っている。たとえば、スキーマ v1 の中に “v1:Dolphin が `rdfs:subClassOf` v1:Fish である” という記述が存在し、この誤りに気付いて改版したスキーマ v2 では、“v2:Dolphin が `rdfs:subClassOf` v2:Mammal である” という記述を書いた場合に、v2:Dolphin を `rdfs:subClassOf` v1:Dolphin と關係付けると、v2:Dolphin が `rdfs:subClassOf` v1:Fish と見なされてしまう。

DAML+OIL では、`daml:Ontology` エレメントの中で、



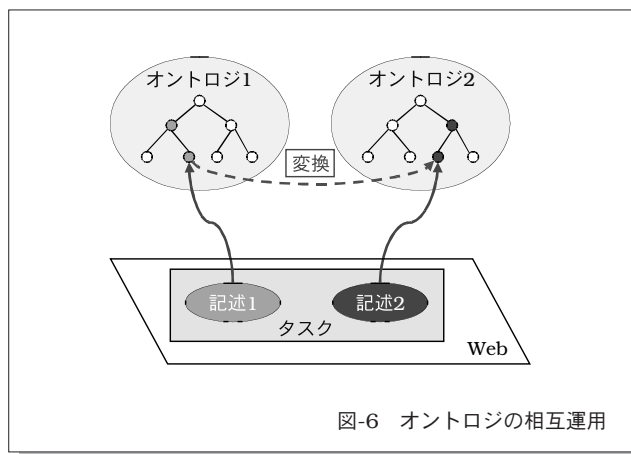


図-6 オントロジの相互運用

daml:versionInfoによって、版の情報を記述することができる。しかしながら、版を表現する文字列の形式が定められていないため、オントロジの版を機械的に処理するソフトウェアにとっては、ほとんど役に立たない。

### (3) オントロジの相互運用性 (Ontology interoperability)

複数の情報提供者が、異なる語彙セットに基づいて記述されたデータを統合しなければならないタスクに対し、オントロジ記述言語は、それぞれの記述の関係を理解するための相互運用性の観点から、オントロジ間でのデータ変換を可能にしなければならない(図-6参照)。要求事項(1)「オントロジの共有化」でも、ある程度の互換性を満足してはいるが、データモデル自体が異なると、同じオントロジを共有化できないケースも多いため、相互運用性は不可欠な要求事項である。

以下は、オントロジ間の変換のためのプリミティブで、オントロジ記述言語がサポートすべき項目である。

- 1) 階層関係 (subclass/superclass relations)
- 2) 逆向き関係 (inverse relationships)
- 3) 同等性 (クラス、プロパティ、個体について)
- 4) 論理的な関係子 (implication, conjunction, disjunction)
- 5) 算術関数
- 6) 集合体
- 7) 文字列処理
- 8) 手続き付加 (任意の複雑なマッピングを定義するための実行コード)

DAML+OILの場合には、クラスとプロパティの階層関係を表現するために `rdfs:subClassOf` と `rdfs:subPropertyOf` を、クラス、プロパティ、個体の同等性を表現するために `daml:equivalentTo`、および、`daml:sameClassesAs`、`daml:samePropertyAs`、`daml:sameIndividualAs` 等のプロパティを、逆向き関係を定義するために `daml:inverseOf` プロパティを持っている。しかしながら、

DAML+OILは、含意関係、算術関数、集合体、文字列処理、手続き付加等の表現能力を持っていない。

### (4) 矛盾の検出 (Inconsistency detection)

データが分散し、全体の管理者が存在しないために、データ間の矛盾をもたらすような場合や、オントロジの拡張によって、一貫性のない記述をもたらすような場合に対して、オントロジ記述言語は、自動的に矛盾の検出ができなければならない。

実現のアプローチとして、オントロジ記述言語は、まず、首尾一貫しない状況が表現できなければならない。これは、否定演算子、集合の非重複性(disjointness)、集合の要素数の制約などによって実現できる。第2に、推論機構が矛盾を検出できなければならない。第3に、可能であれば、不整合が生み出された理由とともに、矛盾を報告するメカニズムが用意されなければならない。

DAML+OILの場合には、共通要素を持たないクラス (`daml:disjointWith`)、要素数の制約 (`daml:cardinality`、`daml:minCardinality`、`daml:maxCardinality`)、補完関係 (`daml:complementOf`) を表現できるため、集合と要素の間の矛盾を検出することができる。

### (5) 表現能力とスケーラビリティとのバランス (Balance of expressivity and scalability)

大規模なデータセットとオントロジを使用する用途に対し、オントロジ記述言語は、多様な知識を表現する一方、効率的な推論手段を提供しなければならない。これら2つの要求は通常は相反するため、オントロジ記述言語は、表現能力の豊かさとスケーラビリティのバランスを保たなくてはならない。

### (6) 使いやすさ (Ease of use)

利用者がセマンティック Web ドキュメントをマークアップしたり、問い合わせたりする用途に対し、オントロジ記述言語は、高度な学習なしで利用できる使いやすさを提供する必要がある。

DAML+OILの場合、オブジェクト指向言語と同等のクラスやプロパティの概念を採用し、構造的な知識の記述を可能にしているが、シンタックスは決して記述しやすいものではない。

### (7) XMLシンタックス (XML syntax)

標準的な形式で書かれたデータをオントロジで取り扱う用途に対し、オントロジ記述言語は、XMLとの連続性を確保しなければならない。XMLはすでに広く受け入れられ、XMLの処理ツールが多数開発されているため、オントロジ記述言語がXMLシンタックスを採用すれば、これらのツールを再利用できる。

なお、RDFはXMLに準拠しているが、メタデータの

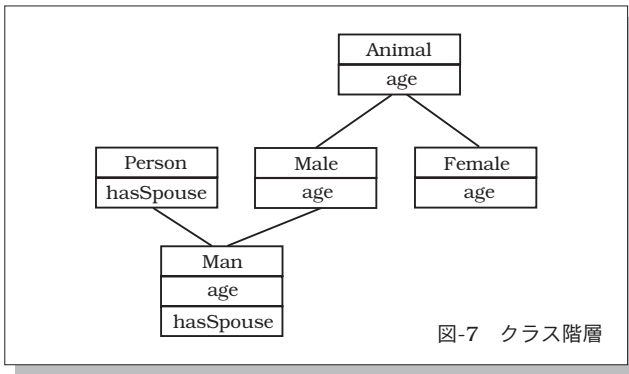


図-7 クラス階層

記述にRDFを使用するのは、あくまでもW3Cの推奨であり、オントロジ記述言語がRDFを基盤とするかどうかについては、まだ意見の一致をみていない。

### (8) 国際化 (Internationalization)

複数の国や文化で利用される用途に対し、オントロジ記述言語は多言語対応のオントロジを提供しなければならない。

RDF仕様では、xml:langプロパティによって言語を指定することはできるが、多言語間に関係付けなどのデータモデルについては何も規定していないため、オントロジ記述言語での対応が必要である。

## ■ オントロジ記述言語OWL

2002年3月に、Patel-SchneiderらによってOWLの最初の提案<sup>5)</sup>がWOWGに提出された。これはOWLの「知識ベース言語」部分の抽象構文に関する提案であり、DAML+OILに類似するfull partに加えて、利用者にとっての使いやすさを意図したlight partが提案されている。その一方、XMLに準拠した具象構文や他のオントロジの参照の問題は扱われていない。本章では、この提案をもとに、現在検討が進められているOWL知識ベース言語を紹介する。なお、OWLは標準化途上にあるため、以下の内容は現時点でのスナップショットである。

### 記述例

OWL記述の基本単位は、クラスとプロパティである。知識を表現する上で、これらがどのように使われるかを、簡単な例をもとに示す。知識の例として、文献5)のほかにいくつかの文献で共通に使用されている、次のものを考える。

- 動物 (Animal) は年齢 (age) を持つ。
- 動物は、オス (Male) とメス (Female) に分かれる。
- 人間 (Person) には配偶者 (hasSpouse) を持つものもある。
- 男性 (Man) は、人間かつオスである。

これらをOWL light partを用いて表した例を以下に示す。また、この記述例におけるクラス間の階層構造を、図-7に示す。

PrimitiveClass (Animal, slot (age, range = xsd:integer, required, singlevalued))

Animalがクラスであり、プロパティとしてageを持つことを表す。ここで、ageの値は必須 (required) かつ1つのみ (singlevalued) で、値域は整数である。

PrimitiveClass (Male, supers (Animal))

Maleはクラスであり、上位クラスはAnimalであることを表す。ただし、Animalを上位クラスに持つクラスがすべてMaleであることは意味しない。

PrimitiveClass (Female, supers (Animal))

Maleと同様にしてクラスFemaleを定義する。

Disjoint (Male, Female)

この文は、クラスMaleとFemaleが互いに素であることを表す。

PrimitiveClass (Person, slot (hasSpouse, range = Person, optional, singlevalued))

Personがクラスであり、プロパティとしてhasSpouseを持つことを表す。ここで、hasSpouseの値はオプションで、もしある場合には1つのみ (singlevalued)、値域はPersonである。

DefinedClass (Man, supers (Person, Male))

ManがクラスでありPersonとMaleを上位クラスとして持つこと、さらに、PersonとMaleを上位クラスとして持つクラスはManであることを表す。

上記のクラス定義を利用すると、13歳の男性であるIchiroは、クラスManのインスタンスとして次のように記述することができる。

Individual (Ichiro, Man, (age, xsd:integer, 13))

### 文法の概要

OWL知識ベース言語の構文は、人工知能分野で使われるフレームの概念を基盤としたlight partと、推論機構による矛盾検出も視野に入れたfull partの2つの部分から構成され、言語機能上はfull partがlight partを包含する。Light partとfull partのいずれも、構文の中核はクラス定義とプロパティ定義である。

#### (1) クラス定義

クラス定義には、クラスの要素が満たすべき性質によってクラスを規定するDefinedClassやPrimitiveClassと、要素の列挙によってクラスを規定するEnumerationClassがある。DefinedClassとPrimitiveClassは、要素が満たすべき性質が前者ではクラスの規定のための必要十分条件となるのに対し、後者では必要条件となる点で異

なる。クラスの要素が満たすべき性質は、上位クラス、プロパティ定義、descriptionと呼ばれる種々のクラス記述などによって表す。

## (2) プロパティ定義

プロパティは、クラスおよびデータ型の要素の間の2項関係である。プロパティ定義では、プロパティに定義域と値域を与えることができるほか、対応付ける要素の数を指定することができる。ここで、定義域はクラス、値域はクラスあるいはデータ型である。プロパティはクラスの中でスロットとして定義することもでき、その場合の定義域はそのクラスになる。

Light partは、full partに比べると、クラス定義、プロパティ定義のいずれにおいても、使用できる記述形式を制限する代わりに、フレームに近い簡単な記述ができるようになっているが、OWLの中でlight partとfull partをどのように位置付けるかについては、現在も議論が続けられている。

## OWL知識ベース言語の特性

OWL知識ベース言語は、オントロジ記述言語に求められる特性のいくつかを提供する。その主なものを以下に挙げる。

### • 階層構造の表現

クラスやプロパティは階層化することができる。

例：PrimitiveClass (Man, supers (Person, Male))

SubPropertyOf (hasMother, hasParent)

### • 同等性の表現

クラスやプロパティの同等性を定義できる。

例：SameClass (HumanBeing, Person)

SamePropertyAs (shoesize, bootsize)

### • 非重複性の表現

クラスが互いに素であることを表現できる。

例：Disjoint (Male, Female)

### • クラスに対する制約の表現

プロパティ値を制約したクラスを規定することができる。次の例では、ageの値を19より大きい整数に制約して、クラスAdultを定義する。

例：DefinedClass (Adult, supers (Person),  
slot (age, range = foo:over19, required))

### • 集合演算を使った表現

クラス間の集合演算を使って、クラスを規定することができる。次の例では、ManとWomanの合併集合としてPersonを定義する。

例：DefinedClass (Person,  
unionOf (Man, Woman))

次の例は、背の高いものを表すクラスTallThingと

Manの共通部分として、TallManを定義する。

例：DefinedClass (TallMan,  
intersectionOf (TallThing, Man))

このようなクラスの規定は、プロパティ定義においても利用することができる。次の例は、プロパティhasAdultDaughterの値域をクラスAdultとFemaleの共通部分によって表す。

例：Range (hasAdultDaughter,  
intersectionOf (Adult, Female))

### • 記述の分散

クラス定義は、複数の文に分散することができる。たとえば、クラスPersonがすでに定義されている場合に、次の文はPersonにプロパティshoesizeを追加する。

例：PrimitiveClass (Person,  
slot (shoesize, range = xsd:decimal))

## ■ 今後の課題

WOWGは1年間程度の活動期間を想定してスタートしており、2002年秋までにはOWL 1.0の言語仕様をまとめる計画である。ただ、オントロジ記述言語の仕様が確定したとしても、オントロジにとって本当に困難な課題は、その言語仕様で、現実の運用に耐え得るオントロジ知識を記述・収集できるかという点であることを理解いただきたい。オントロジ知識をゼロから人手で記述するのは、時間、労力、コストのいずれの観点から考えてもナンセンスであり、オントロジ知識の学習・自動獲得が重要な技術課題となる。すでにいくつかのグループでこうした研究が始められており<sup>6)</sup>、ドメインごとのドキュメントからのオントロジ知識の抽出、オントロジ変換規則の獲得、知識のメンテナンス、推論機構などの研究の発展が期待されている。

### 参考文献

- 1) Fensel, D. et al.: OIL: An Ontology Infrastructure for the Semantic Web, IEEE Intelligent Systems, MARCH/APRIL 2001, pp.38-45 (2001).
- 2) W3C Web-Ontology Working Group:  
http://www.w3.org/2001/sw/WebOnt/
- 3) Heflin, J., Volz, R. and Dale, J. (Eds.): Requirements for a Web Ontology Language, W3C Working Draft (Mar. 2002).  
http://www.w3.org/TR/2002/WD-webont-req-20020307
- 4) Harmelen, van F., Patel-Schneider, P. F. and Horrocks, I.: Reference description of the DAML+OIL Ontology Markup Language (Mar. 2001).  
http://www.daml.org/2001/03/reference
- 5) Patel-Schneider, P. F., Harrocks, I. and Harmelen, van F.: Proposed OWL Knowledge Base Language (2002).  
http://www.cs.vu.nl/~frankh/spool/OWL-first-proposal/
- 6) Maedche, A. and Staab, S.: Ontology Learning for the Semantic Web, IEEE Intelligent Systems, MARCH/APRIL 2001, pp.72-79 (2001).

(平成14年5月21日受付)