Semantic Web : 機械による Web コンテンツの解釈と自動処理のための知識処理アーキテクチャ

Semantic Web—Knowledge Processing Architecture for Interpretation and Automated Processing of Web Contents by Computer System

山 田 繁 夫

要 約 昨今，Web は膨大な量の文書や画像，映像などの情報で溢れている．コンピュータは，web に存在する情報を蓄積したり，表示したり，あるいは分類したりするが，それらの情報を単にデータとして扱うだけで，情報が意味するものの理解を要するような処理をすることはできない．Web 上の情報を理解し，利用するためには人間の知識を必要とする．1998 年に Tim Berners–Lee により Semantic Web の構想が示され，World Wide Web コンソーシアム（W３C）の活動の中で，機械が Web 上のリソースを解釈できるようにするための仕様となる Resource Description Framework（RDF）が標準化された．これにより，機械が我々に代わって欲しい情報を探し出したり，国際会議に出席するために必要なすべての予約をアレンジしたりするパーソナルエージェントの実現に一歩近づいた．本稿では，まず Semantic Web の背景となる現在の web 利用における問題を，情報の検索，共有，サービスの実行の観点から述べ．それに続いて，現在も仕様策定の途中段階にある Semantic Web のアーキテクチャの概要を述べる．

Abstract Now the web provides huge volume of documents, images, digital contents and etc. The computer system performs the processing including storage, categorization, transformation and representation of such digital information. However, it processes such information simply as plain data, and cannot process them in a manner that requires the understanding of the meaning of information. To understand and use the information transmitted over the Web, the human intelligence is required. The Semantic Web vision was first presented in 1998 by Tim Berners–Lee, the utilization of having our own software agent who knows everything our needs, understands our preferences and constraints, searches requested information on the web, as well as arranges and registers anything relevant to the attendance to the conferences, which was scientific futuristic story, has seemed to be realistic. This paper reviews requirements and technologies relevant to Semantic Web and presents an architectural overview of its technology.

1. は じ め に—背 景 と 目 的

World Wide Web が初めてインターネットに登場した時，インターネットの使われ方は電子メール，電子ニュースなどのコミュニケーションと，文書やプログラムを共有するためのファイル転送が主であった．Mosaic という名のオープンソース形式で配布された Web ブラウザと，ハイパーテキストと URL の強力な記述力により急速に普及し，World Wide Web は，すぐに，インターネットで最も良く利用されるアプリケーションとなった．ほぼ同時期に，インターネットが商用目的の利用に開放され，e–コマースサイトや，Yahoo，Google などのディレクトリや検索エンジンサービスも登場し，Web はますます便利になり，web の利用は利用者数，コンテンツ数共に爆発的に増加し，現在もなお増加を続けている．そして，最近，Seman-

tic Web と呼ばれる，人ではなく機械が解釈し自動処理を可能とする web 技術が登場した．機械が web 上にある様々なコンテンツを解釈し，自動的に処理することができるようになれば，ユビキタスコンピューティング環境と合わせて，web を媒体とする人のコミュニケーションに新しい形をもたらす可能性がある．

本稿は，Semantic Web にフォーカスし，それがどのようなものなのか，機械はどのように意味を解釈できるのか，どのような価値をわれわれにもたらすのか，という観点で，それを現在の web の延長線上に捉えることを試みる．

1） 情報基盤技術としての Web―成長を続ける Web

今日，Web といえば World Wide Web を指すほど，Web はわれわれの日常に欠かすことのできない存在となっている．Uniform Resource Identifier（URI）[1]は世界中に散らばる情報やサービスを一意に識別することを可能にし，URI を知っていれば誰でも目的の情報資源を利用することができる．たとえ事前に URI を知らなくとも，Yahoo のような索引サービスは図書館の目録や電話帳のようにカテゴリ別の分類を手がかりに目的の情報資源を得ることができるし，Google のような検索エージェントサービスは思いつくキーワードから目的の情報を探し出すことを可能とする．そして，Web はまた，情報の消費者のみならず，情報の提供者側に対しても情報を発する自由を与え，誰でも，何に関するものでも，情報を発し，サービスを提供することができる．現在，Google のような検索サービスから取得可能な情報資源の数は一説によれば 10 の 9 乗のオーダであり，その数は現在でも指数オーダで増加している．検索サービスに登録されていない情報資源や動的に生成されるものを含めれば，それ以上のオーダ数の情報資源が web 上に存在している．これの意味することの一つは，索引サービスや検索サービスにより到達できる情報資源のカバー率が小さくなっていき，目的とする情報を得ることが難しくなるということである．たとえば，現在，検索エンジンにキーワードを与えて検索した結果のリスト長が数十のオーダであるならば，近い将来にそれは数千のオーダになり，目的とする情報を探し当てるのに人が要する労力の許容範囲を超えてしまう．現在良く使われているページランクアルゴリズムを使った重みづけによるフィルタリング技術がこの問題に対する一つの解を与えているが，別の解として，それら膨大な数の情報資源に対し，それら各々の〈意味するもの〉を機械が処理可能な形式で現在の web に追加し，検索サービスはその付加された知識を利用して，より精密で賢い検索を実行できるようにすること，が考えられる．

われわれが web の情報空間を探索するとき，二つの種類があると考えられる．一つは，ある対象が判っていて，その対象の属性に関するもの–たとえば，ある国際会議の開催場所やそのチケットの値段など–を問い合わせる場合である．あらかじめ対象領域の概念と概念間に存在する関係をメタ知識として構造化して与えておき，その概念構造を機械が理解し，目的の情報リソースへナビゲーション（誘導）する方法が考えられる．もう一つは，逆の場合で，ある属性を持つ対象クラスを探し出す場合で，たとえば，〈ジュネーブにあり一泊 200 フラン以下のホテルはあるか？〉という問いがこれに相当する．先の概念構造に，ホテル，所在地，ジュネーブ，料金などの宿泊に関する語彙とそれらの関係が与えられていれば，機械と人は求める対象の意味を共有できるので，〈属性，値〉の組を与えて web 空間を検索（クエリ）できるようになる．

2) イントラネットにおける知識管理

企業ポータルや部門ポータルと呼ばれる，企業内にある文書や知識を体系化し，Yahoo のような索引サービスや，Google のようなクローラによる情報の自動収集とキーワード検索サービスを使って，情報共有を目指すアプローチがある．インターネットよりもスケールは小さいが，企業内のイントラネットも前述の情報量増大に伴う問題を持っている．企業の競争力が，企業内に蓄えられている知識や記憶を如何に上手に利用できるかに依ることを考えれば，多くの企業にとって，これは大きな問題である．

企業に蓄えられる知識や記憶には，印刷された文書や，PDF 形式，ワードプロセッサやプレゼンテーションアプリケーションのファイル形式など，多様な形態があり，また，蓄積される場所も，個人の PC，部門のファイルキャビネットや Web サーバなどに分散している．このような，ばらばらに存在する情報の断片がそれぞれ何であるかという概念と，それらの関連を機械が解釈可能な形式のメタ情報として記述し，大きな知識ネットワークに編み上げることができれば，機械の持つ情報処理能力を使って，社員の誰もがそれら知識を効率良く利用することができるようになる．

3) Web Service

Web が扱う情報資源は，当初はテキストとグラフィックスによる情報共有が主たる応用領域であったが，われわれの実世界の状態を変更する―副作用を伴う―サービスがこれに加わった．たとえば，飛行機のチケットを販売するサービスや，遠隔でコーヒーポットを操作するサービスなどがそれである．このような web の情報資源を Web サービスと呼んでいる．Web サービスを人ではなく機械が呼び出し，サービスの実行を自動化させるための枠組みがある．XML-RPC[2]や SOAP[3]など，この枠組みを使って Web を媒体とするアプリケーション間のコミュニケーションを利用するケースが増えてきている．これを実現可能とするプログラムインタフェースを XML Web サービスと呼ぶ．XML Web サービスの技術的な枠組みは，次の三つの要素技術の上に成り立っている．

UDDI：Universal Description, Discovery and Integration（UDDI）は，サービスを要求するクライアントが目的とするサービスを探し出す機構を提供する．

WSDL：Web Service Definition Language（WSDL）は，サービスをネットワーク上の終端，あるいはポート，の集まりとして定義するための言語仕様を定める．

SOAP：Simple Object Access Protocol（SOAP）は，XML（Extensible Markup Language）で符号化したデータを統一された方法で受け渡すためのメッセージの配置を規定する．また，HTTP[4]を通信規約とする場合の結合（バインド）を規定する．SOAP は，基本的には，web 上で実装された遠隔手続き呼び出し(RPC)を実現する技術となる．

これら三つの要素技術は，誰が（UDDI），どのようなサービスを（WSDL），提供するかを探索し，その中から最も目的に合致したサービスを選択し，利用する（SOAP），という web サービスを実現するために最小限必要なものである．

一般には，ある目的を達成するためには単一のサービス呼び出しだけでは不十分で，複数の異なるサービスを，より複雑な手続きやシナリオに組み上げることが要求される．今はこれら上位層の部分を，エンジニアが静的な構造をもつ手続きとして手作業で組み立てているが，それを実行時に，機械により即興的に実施することができれば，その時点で最

も有利なサービスを呼び出したり，実行時に新たなサービスを見つけてシナリオに組み入れ，呼び出すことが可能となる．

ソフトウェアエージェントが，web サービスを利用して，人に代わって，出張の旅程をアレンジしたり，チケットの購入を実行するような構想を考えるとき，前述の web サービスの枠組みに不足している要素がある．一つは，機械がサービスの意味を解釈するための領域知識，もう一つは，機械が達成すべき目的を理解するための領域知識である．

4） 本稿の構成

本章では，web 上の情報資源の利用について，情報の検索，企業の知識管理，サービスの実行の三つの視点から述べた．領域知識を機械が解釈可能な形で現在の web に導入し，それを機械が処理することにより，人の知識に依らずに web リソースを自動処理する新しい web のモデルが実現することを示唆している．以下本稿では，この示唆を背景として，まず，2 章で semantic web の技術的な基底となる web の技術アーキテクチャを整理し，ひきつづき 3 章にて〈意味するもの〉を機械が処理可能なように形式的に記述することに対する考察を与え，それを semantic web がどのようにアプローチしているかを述べる．また，4 章では semantic web service という名称で話題となっている要素技術について述べる．最後に，現時点において未解決の課題を述べ，結びとする．

2. Web のアーキテクチャ

本章では，Web の概念モデルを構成する三つの概念；リソース，リソースの識別子と情報空間，リソース参照とフラグメント識別子，を技術アーキテクチャの視点から整理する．後で述べるように，これらの概念は，semantic web を実現するための技術基盤となっている．

2.1 リソースと URI

Web は多様な目的に利用できる普遍的な情報空間で，空間内には Uniform Resource Identifier（URI）[1]によって一意に識別されるリソースがある．リソースは情報空間からエンティティやエンティティの集合に対する概念的な写像で，電子化されたドキュメントや画像，サービス（たとえば東京の現在の天候を報告するサービス）などが良く知られるリソースの例である．リソースは必ずしも電子的に取得できる必要は無く，人や会社，図書館にある本などもリソースとして取り扱うことができる．

汎用的な URI の構文は，〈スキーム名〉:〈スキーム固有部分〉で，スキーム名の部分とスキームに固有の部分の二つの要素で構成される．一般的には，スキーム固有の部分は三つの部分に構造化され，URI は次のような形となる：

〈scheme〉://〈authority〉〈path〉?〈query〉

〈authority〉部分は非営利法人である ICANN が管理するルートを根元にして階層的に名前の管理が委譲された DNS のドメイン名，〈path〉部分は委譲されたドメイン名の管理権限を持つ所有者が割当てる任意の文字列，〈query〉部分は問い合わせ式である．この構造化によりリソースは web 空間で一意に識別される．次に挙げるものはよく使われる URI の例である：

http://www.math.uio.no/faq/compression/part 1.html （例 1）

ftp://ftp.is.co.za/rfc/rfc 1808.txt （例 2）

telnet://melvyl.ucop.edu/ （例 3）

mailto：mduerst@ifi.unizh.ch　（例 4）

news：comp.infosystems.www.servers.unix　（例 5）

urn：uuid：58f202ac−22cf−11d1−b12d−002035b29092　（例 6）

最初の三つの例は一般的な形の URI で，それぞれ，Hypertext Transfer Protocol でアクセスする http スキーム[4]，File Transfer Protocol でアクセスする ftp スキーム，TELNET プロトコルによる対話型サービスの telnet スキームである．例 4 と 5 は，それぞれ電子メールアドレスのスキーム，USENET ニュースグループのスキームである．これら五つの URI スキームはネットワークの終端などのリソースへのアクセス手段を内包しているため，Uniform Resource Locator（URL）でもある．最後の例は，Uniform Resource Name（URN）の名称スキームで，この形式の識別子は名前のみを持ち，リソースへのアクセス手段を持たないため，リソースのコンテンツにアクセスするためには名前を解決する別の仕組みが必要となる．

2.2　URI 参照とフラグメント識別子

URI により識別されるリソースを参照することを URI 参照と呼ぶ．URI 参照は絶対的か，相対的かのいづれかであり，どちらの場合にもフラグメント識別子と呼ばれる情報を付加できる．URI 参照の完全な形は次のようになる：

〈スキーム名〉：〈スキーム固有部分〉#〈フラグメント識別子〉

http：//foo/bar#frag　（例 7）

/bar#frag　（例 8）

#frag　（例 9）

上記三つの例はすべて同一の URI 参照である．例 7 は絶対的な URI 参照形式，例 8 は〈scheme〉と〈authority〉部分を省略した相対形式，例 9 は URI 部分を省略した相対形式である．

フラグメント識別子（上記例の frag）は，リソースをアクセスする側で解釈される情報で，サーバへはフラグメント識別子は渡されず，リソースをアクセスする側にコンテンツ全体が渡された後に，フラグメント識別子を使って参照の対象となるオブジェクトが生成される．

3.　Semantic Web のアーキテクチャ

Semantic Web は，対象領域の概念と，それら概念間の 2 項関連の集合から構成される知識空間である．2 項関係をラベルつき有向グラフ（DLG：directed labeled graph）の弧で記述すれば，単純な意味ネットワークとなる．このような知識空間は対象領域ごとに多数遍在するが，それらの空間は，概念の関連付けにより結合され，一つの普遍的な知識空間を形成する．これにより，「誰もがあらゆることについて何でも発信できる（allowing anyone to make statements about any resource[5]）知識空間」という Semantic Web の構想が実現する．このような普遍的な知識空間は，新しい知識をどんどん組み込むことができる分散した巨大な知識ベースと見ることができる．一方，結合された知識ベースは全体としての整合性が保たれることを保証できず，無秩序な部分的な不完全な断片知識の集まりとなる．このような知識空間は，かつて古典的な人工知能の研究領域が扱った一貫して整合性の取れた知識ベース系とは異なる性質をもち，知識空間に記述された知識概念を利用して有意な結果を得るための理論的な枠組みが必要となる．

3.1　分散された知識を扱うための記述言語の階層構造

さまざまな対象世界を記述した知識ベースがバラバラに作られ，web 空間に分散して存在している．Semantic web 構想を実現するために，それらローカルに記述された知識の相互運用性を確保するための理論的工学的な枠組みが必要となる．そのようなアーキテクチャを示すものの一つに，Tim Berners–Lee が提唱した“layer cake”として知られる階層構造（図 1）がある．開いた系において相互運用性を確保するために階層構造をもつアーキテクチャは，他にもインターネットプロトコルや CORBA などがあり成功しているが，提唱されるアーキテクチャでは，それを対象世界を記述するための言語に適用している．図において，各層は対象世界を記述するための言語と機能を示しており，“より表現力の高い言語が相互運用性の層を一段ずつ上がっていく”（文献[6]より引用）．上位層の信頼層（Trust）と証明層（Proof）は，ばらばらで無秩序な知識断片の集まりから推論規則を使って導出された結論に対する妥当性を判断するための品質保証のような機能を提供する．デジタル署名技術と組合わせれば，誰によって書かれた知識を使って，どのような推論経路で結論を得たかを説明することができるようになる．このような機能は，自律的なソフトウェアエージェントのような開かれた系を扱うアプリケーションが実用レベルとなるために必須のものとなる．

階層構造を持つアーキテクチャの設計と構築は，現在のところ下位層から順次ボトムアップで進められており，5 番目のオントロジ層までの仕様が固まっている．以下，本章では，semantic web の基本的な構成要素となる RDF 層を中心に各層について順次述べる．

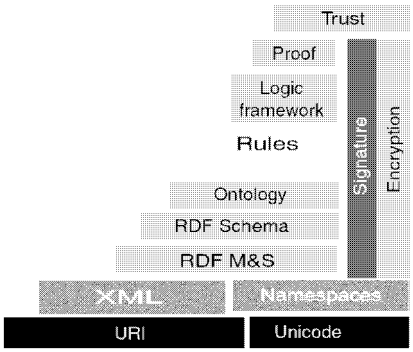

図 1　Semantic web 階層アーキテクチャ

（http://www.w3.org/2002/Talks/09–lcs–sweb–tbl/Overview.html より引用）

3.2　知識を宣言的に記述する RDF 言語

Resource Description Framework（RDF）[5]は Semantic web において知識を記述する基本的な言語である．RDF の抽象的な構文は RDF グラフと呼ばれる RDF 3 項組（RDF triple）の集合で，個々の 3 項組は，主語（subject），述語（predicate），目的語（object）の三つの部分から構成される．この 3 項組は図 2 で示すように，節–弧–節，という有向グラフ図（DLG）で表現することもできる．RDF グラフの節は主語と目的語，弧は述語となる．グラフ図の中の弧の向きは重要で，常に主語から目的語の方向へ向いている．RDF 3 項組は主語と目的語で示す二つの概念の間に述語で示される関係が成立することを表明している．3 項組の集合である RDF グラフは，グラフに含まれる 3 項組の示す表明がすべて成立する（論理積）ことを

示している．この RDF グラフを自然言語風に記述すれば，「http://www.xyzbook.com は商品 http://www.xyzbook.com/sku/00498 を販売する．商品 http://www.xyzbook.com/sku/00498 の名称は XML 4 You で，その売上数は 12417 冊である」となる．

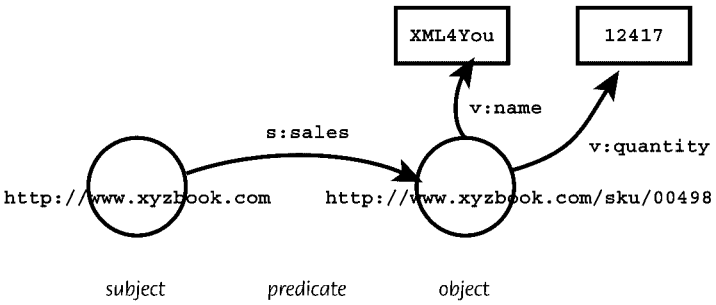

図 2　RDF グラフ DLG 図

RDF グラフを形式的に記述する方法は他にもいくつかある．Notation 3 (N 3) 記法[7]は RDF グラフをテキスト形式で記述する記法で，3 項組の主語，述語，目的語をこの順に並べピリオドで文を終了する．図 2 の RDF グラフは N 3 記法を使って次のように記述できる．

〈http://www.xyzbook.com〉〈s : sales〉〈http://www.xyzbook.com/sku/00498〉.
〈http://www.xyzbook.com/sku/00498〉
　〈v:name〉“XML4You”,
　〈v:quantity〉“12417”.

さらに，図 2 は次のように XML 文書として記述することもできる．

```
〈rdf : RDF〉
  〈rdf : Description about=“http://www.xyzbook.com”〉
    〈s : sales
      rdf : resource=“http://www.xyzbook.com/sku/00498”/〉
  〈/rdf : Description〉
  〈rdf : Description about=“http://www.xyzbook.com/sku/00498”〉
    〈v : name〉XML 4 You〈/v : name〉
    〈v : quantity〉12417〈/v : quantity〉
  〈/rdf : Description〉
〈/rdf : RDF〉
```

この XML 文書形式による記述が，RDF グラフの公式の交換構文形式となる．

3.3　URI 参照形式にもとづく概念の識別

本稿前半に述べたように，Web のアーキテクチャでは，URI は情報空間の中でリソースを一意に識別するための識別子であった．Semantic Web の知識空間では，この URI の解釈を拡大し，web 空間上のあらゆる概念（RDF グラフの節と弧）を URI 参照形式により一意に指し示す．概念を URI 参照形式を使って記述することにより，バラバラに作成された RDF グラフを結合（関係データベースにおける join 操作）することができるようになる．

RDF グラフをテキスト形式で記述する際には，XML 名称空間の機構 （XML name–space）を使い，先の例で示したように〈s : sales〉や〈v : quantity〉のように URI を簡略化して記述したり，名前の衝突を回避することができる．

3.4 RDF の形式的な解釈＝セマンティックス

RDF 形式で与えられた記述が〈意味するもの＝記述の解釈〉は色々なファクタに依存する；ラベル名前が示す社会通念，自然言語の形式で付与された注釈，URI 参照リンク先の文書に書かれた記述，……しかしこのような形で与えられる意味は機械が処理することができない．RDF グラフを機械が解釈するための形式的仕様[8]は，モデル理論を基礎におく．モデル理論は形式的な意味を取り扱う理論で，形式的に記述された式を解釈に関連づける．

モデル理論によるセマンティックス

RDF 言語の文である RDF 3 項組は，対象世界に対する何らかの事柄を宣言しており，それは対象世界が満たすべき要件を述べている．換言すれば，これら宣言文が対象世界が取り得る可能な状態を制約している．宣言文の数が多くなるほど，対象世界が取り得る可能な状態数は少なくなる．

解釈は名前の集合である語彙に対して与えられる．語彙は，RDF グラフにある URI 参照とリテラルの集合である．前例の図 2 の場合には，URI 参照〈http://www.xyzbook.com〉や〈s : sales〉などと，リテラル“XML 4 You”，“12417”が語彙の要素となる．語彙の要素間には，〈s : sales〉や〈v : quantity〉などの 2 項関係の抽象的な構造がある．ある文の語彙に対する解釈とは，その文に対して真理値を与えることであり，文の三つの要素が語彙の要素に対応し，かつ，2 項関係の抽象的な構造に一致するとき，かつその場合に限り，その文は真である（正しい）とする．

以上，RDF 層について簡単に整理すると，RDF グラフは，対象領域の概念とそれら概念間の関係のデータモデルであり，このデータモデルに対する単純なセマンティックスを提供する．このデータモデルは XML 構文を使って RDF 文書として記述することができる．

3.5 RDF Schema 層のセマンティックス

階層構造を持つ言語のセマンティックスの拡張とは，新たな構文要素の追加と，それらの解釈に対する強い制約の導入である．構文要素としての語彙は増加し，解釈可能な世界は狭まる．通常，セマンティックスの拡張では，語彙の中の特別な名前に対して，特別な意味を与える．そのような解釈を他と区別するために，セマンティックスの名前とハイフンを付けて，たとえば，RDF–解釈，RDFS–解釈，などと呼ぶ．

RDF Schema は RDF のセマンティックスを拡張するもので，RDF 言語を使っていくつかの語彙を追加し，それら新たに導入した語彙に対する解釈を与え，RDF–解釈に強い制約を加える．追加される主な構文要素には，総称的な概念を表すリソース（rdfs : Resource），クラス（rdfs : Class）とサブクラス（rdfs : subClassOf），プロパティ（rdf : Property）とサブプロパティ（rdfs : subPropertyOf），プロパティの定義域（rdfs : domain）と値域（rdfs : range）など解釈の制約を強めるもの，集合や列を扱うためのコンテナ（rdfs : Container），人が理解することを助ける文書化のための語彙群（これらの形式的な意味は定義しない）rdfs : comment, rdfs : seeAlso, rdfs : isDefinedBy, rdfs : label，などがある．サブクラス，サブプロパティに対

する公理を導入し，解釈の制約を強め，サブクラスやサブプロパティが推移的（transitive）かつ反射的（reflexive）であると解釈する．サブクラス関係が推移的であるとは，あるクラス A のサブクラス A 1，A 1 のサブクラス A 2 があるとき，クラス A 2 はクラス A のサブクラスであると解釈する．また関係が反射的であるとは，あるクラス A があるとき，クラス A はそれ自身（クラス A）のサブクラスであると解釈する．また，rdf：type の解釈を強め，あるクラスのインスタンス（個体）は，その親クラスのインスタンスでもあるよう解釈する．

図 2 で示した例を RDF　Schema 層で，次のように書店と書籍のクラス階層を導入して拡張することができる（N 3 記法）：

〈ex：Shop〉 rdfs：subClassOf rdfs：Resource.
〈ex：Bookstore〉 rdfs：subClassOf〈ex：Shop〉.
〈http://www.xyzbook.com〉 rdf：type〈ex：Bookstore〉.
〈ex：Product〉 rdfs：subClassOf rdfs：Resource.
〈ex：Book〉 rdfs：subClassOf〈ex：Product〉.
〈http://www.xyzbook.com/sku/00498〉 rdf：type〈ex：Book〉.

サブクラス rdfs：subClassOf に関する公理により，RDFS-解釈では，次の二つの文はいづれも真となる．

〈http://www.xyzbook.com〉 rdf：type〈ex：Shop〉.
〈http://www.xyzbook.com/sku/00498〉 rdf：type〈ex：Product〉.

以上，RDF スキーマ層について簡単に整理すると，RDF スキーマは，RDF リソースのクラスとプロパティ，それらの汎化階層構造を記述し，解釈を与えるための語彙とセマンティックスである．

3.6　オントロジ層のセマンティックス

オントロジという言葉の指すものは，存在論を指す哲学用語や，その解釈を拡張して人工知能の領域で扱われてきたものなど，多々あるが，ここで指すオントロジとは，後者のものを指し，対象世界に存在するものごとと，それらの間にある関係を記述したものである．

さて，本章の冒頭にて，いろいろな所で，いろいろな人が，いろいろな対象世界について記述した，ばらばらな知識空間を関連付けて，一つの知識ベースとして取り扱うという構想を述べた．オントロジ層は，このようなばらばらな知識空間の相互運用性を実現するための，語彙とセマンティックスを定義する．このようなオントロジを記述するための言語が，W 3 C の Web オントロジグループ（WebOnt）により作成された OWL（Web　Ontology　Language）[9] である．OWL は RDF スキーマのクラスとプロパティに対して，より高度な記述を与えるための語彙とセマンティックスを与える．たとえば，クラスが互いに排他的な関係であることや，プロパティのカージナリティなどを記述できる．

OWL には，3 種類のサブ言語があり，実装上の制約や利用目的に応じて使い分けることができるようになっている．OWL-Lite はクラス階層と単純な制約を与える構文要素を持ち，主にシソーラスを定義したり，他の分類階層を取り込む目的に設計されている．OWL-DL は OWL のすべての構文要素を含み，かつ計算可能性を持つ言語である．DL とは記述論理[10]（description　logics）のことを指し，人工知能の研究成果である知識表現のための理論で OWL 言語の形式的枠組みの基礎となっている．記述論理では，クラス，サブクラス，個体（インスタ

ンス），クラス間の制約関係，個体間の関係，という観点から対象世界を形式的に取り扱い，人も自然な形で取り扱うことができる．OWL-FULL は RDF の持つすべての構文と表現力を必要とする分野のための言語であるが，決定可能性は保証していない．

OWL の抽象構文[9]は，RDF 3 項組よりも大きな塊を単位とするフレーム風の記述スタイルを持っている．前例は，OWL 抽象構文を使って次のように記述できる．

```
Ontology(ex:ontology
  DatatypeProperty(v:name)
  ObjectProperty(ex:sales)
  Class(ex:Book partial ex:Product)
  Class(ex:Bookstore partial ex:Shop)
  Individual(〈http://www.xyzbook.com〉 type(ex:Bookstore)
    value(s:sales
      Individual(type(ex:Book)
        value(v:name“XML 4 You” ^^ xsd:string)))))
```

OWL 抽象構文から RDF へのマッピングが定義されており，OWL 言語で記述されたオントロジは XML 形式の RDF 文書として交換することができる．

3.7 オントロジ層より上位の層

提唱される階層アーキテクチャでは，オントロジ層の上位層として，ルール/論理フレームワークの層，さらに上位の層として，証明や信頼の層が位置づけられている．W３C では現在，ルール/論理フレームワーク層の言語開発が視野に入っている．ルール層では，問い合わせやフィルタリングのための汎用的なルール言語が規定されることになっている．関係データベースに対する SQL 問い合わせに相当する操作が OWL オントロジ知識ベースに対して可能となる．論理フレームワーク層には，単調な推論機構に対する論理記述を可能とする汎用的な言語が期待されている．これにより，様々な推論系に対して推論規則が出力できるようになる．このとき，推論系から返される結論に対する検証手段を提供できなくてはならない．

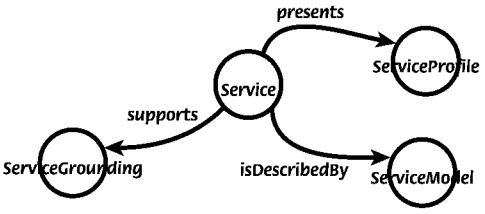

図３　OWL-S オントロジの中核概念

4. Semantic Web と Web サービス

本章では，semantic web の一つの応用領域として，形式的に記述された web サービスのオントロジを利用して，機械が自動的に web サービスを呼び出すような枠組みについて述べる．

Web サービスを特徴づけるものとして，異種混成（ヘテロ）の，疎に結合された，分散した系であり，サービスあるいは機能を提供するコンポーネントの即興的な変更に適応できるこ

と，が挙げられる．ここで，〈疎に結合された〉とは，系を構成するコンポーネント間の相互作用が設計時に厳密に与えられるのではなく，必要とする機能を提供するコンポーネントを実行時に見つけ出し呼び出すことを指す．

機械が，web 上のサービスを見出し，選択し，実行し，さらに，複数のサービスを連結したり，サービスの実行を監視し制御することを可能とするためには，それらサービスが何者であるのか，何ができるのか，どのように動作するのか，というサービスに関する概念を機械自身が理解できなければならない．Semantic　Web は Web サービスに対して，これら知識を付加することができる．

4.1　Web サービスのオントロジ

Web サービスに対するオントロジの記述のために，OWL オントロジ言語の上位オントロジとして OWL–S[11]が DAML から提唱されている．OWL–S は，Web サービスを，サービスが何をするのか（サービスプロファイル），サービスにより何が起こるのか（サービスモデル），サービスをどのように呼び出すのか（サービスグランディング），という三つの視点で意味づける．以下，OWL–S オントロジの中核となる四つの概念と三つのプロパティを説明する（図 3）．

総称的な概念クラス Service は web 上のサービスに対する参照点となり，Service のインスタンスが web 上の一つのサービスに対応し，特に XML Web サービスの場合にはポートと対応する．概念クラス ServiceProfile は〈サービスが何をするのか〉を示すもので，サービスを探しているソフトウェアエージェントなどが要件の充足性を調べるときに利用する知識となる．presents というプロパティがサービスとサービスプロファイルとを関連づける．概念クラス ServiceModel は〈サービスがどのように動作するか〉を示す．エージェントがサービスを連携させたり，実行を監視したりするときに利用する知識となる．サービスとサービスモデルとは，isDescribedBy というプロパティで関連づけられる．概念クラス Service–Grounding は，〈サービスをどのように呼び出すか〉を示す．たとえば，エージェントがサービスを呼び出すためのプロトコル，メッセージ形式，ポート番号やネットワークアドレスなどの情報が関連付けられる．サービスグランディングは実装依存であり，たとえば，XML Web サービスのための WSDL グランディングや，CORBA のための IDL グランディング，HTTP フォームによるサービスのための HTTP–form グランディングなどが考えられる，サービスとサービスグランディングは supports プロパティで関連付けられる．

4.2　サービスの発見のためのオントロジ

概念クラス Profile は，ServiceProfile のサブクラスで，web サービスが何であるのかを，サービスの名称やカテゴリ，コンタクトアドレスなどの非機能的な属性と，入出力などの機能的な属性の，二つの視点から記述する．前者は UDDI などのレジストリに登録し，サービス発見の目的に利用することができる．後者は，サービスを，入力–処理–出力というデータフローと，事前条件–行動–効果というプロセスの観点から記述する．これらは次節で述べるサービスモデルにおける最上位レベルのプロセスを要約したもので，サービス発見の過程において，サービス利用の要件が合致するかを知ることを可能にする．

4.3　サービスの実行，連結と連携のためのオントロジ

プロセスモデル（ProcessModel）は，サービスモデル（ServiceModel）のサブクラスで，web サービスの動作をプロセスの概念を使って捉える．図 4 にプロセスオントロジの中核となる概念構造を示す．プロセスは，入力されたものを加工し出力するという関数やデータフローのセマンティックスと，呼び出し前の世界の状態と完了時の状態の遷移で捉える状態遷移のセマンティックスの二つの見方で記述する．関数やデータフローのセマンティックスは，関数に対するパラメータである二つのプロパティ，入力項目（hasInput）と出力項目（hasOutput）で記述できる．状態遷移のセマンティックスは，サービス呼び出しの事前条件（hasPrecondition）とサービスの及ぼす効果（hasEffect）のプロパティを使って記述できる．

プロセスは，原子的であるか（AtomicProcess），単純であるか（SimpleProcess），複合的であるか（CompositeProcess），の三つの概念サブクラスに分割される．これにより，web サービスを連結し連携させるために必要な，プロセスの入れ子構造を持つように構造化することが可能となる．原子的なプロセスは，直接呼出し可能なプロセスで内部構造を持たず，呼び出し側から見れば，実行は単一のステップで完結する．原子的なプロセスはサービス呼び出しのためのグランディング概念と関連付けられる．単純なプロセスは，プロセスを抽象化したもので，呼び出し側から見れば，内部構造を持たない単一のステップから成るプロセスである．単純なプロセスは直接呼び出されるものではなく，原子的なプロセスにより具現化され（realizedBy），あるいは，合成的なプロセスへ展開（expandsTo）される．複合的なプロセスは，入れ子構造を持つプロセスで，連結や分岐などの制御構文要素（ControlConstruct）を使ってプロセスを構造化する．このようなプロセスの内部構造を機械が理解することにより，ばらばらな web サービスを入出力の連鎖の糸で紡いで連携させ，一連の実行を制御できるようになる．たとえば，飛行機とレンタカーを使った旅程のアレンジをするサービスを考えるとき，データフローのセマンティックスを使えば，航空会社の提供する web サービスによりフライトの予約が確定すれば，到着する空港と時刻が定まり，それがレンタカー会社の提供するレンタ

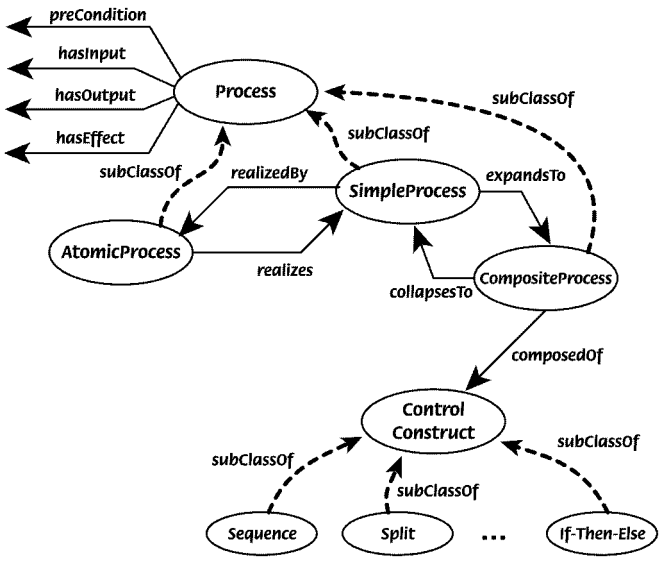

図 4　OWL–S プロセスオントロジの中核概念，http://www.daml.org/services/より引用

カーの予約のためのサービス呼び出しへの制約条件となる．

5. お わ り に

今から 15 年以上前の 1988 年，Apple 社が，21 世紀までに開発することを目標として提示した Knowledge Navigator と命名されたコンセプトがある．ノートサイズのハードウェアは，タッチ・パネルと音声により操作する知的なインタフェースを備える．後にコンセプトを可視化するために公開されたプロモーションビデオの中に，ソフトウェアエージェント（電子秘書）が所有者である教授のスケジュールをアレンジしたり，ネットワーク上にあるデータベースから研究論文を探し出したりするシーンがあった．そして現在，ハードウェア技術の向上により機器は小型化され処理性能と機能が向上した．日常生活で使われるさまざまなハードウェアには膨大な計算能力と通信能力が備わり，そのような計算能力は，われわれが意識することなく，生活の中に浸透してきている．携帯パーソナルメディア装置，車載コンピューティングシステム，コンピュータを搭載した冷蔵庫などの家電製品，通信機能を持つカメラや温度センサ，GPS や IC タグなどの位置情報を追跡するデバイスなど，多様なシステムがネットワークに接続されるようになり，また，音声や人の顔を認識するソフトウェアのようなソフトウェアコンポーネント技術も進化し，当時のコンセプトを実現するための基本的な要素技術の大半は手に入る技術となった．しかし，15 年前に鮮やかに描かれた知的なパーソナルエージェント構想は，今でも現実味の薄いサイエンスフィクションのように見える．欠けているものは，コンポーネントを柔らかく合成することを可能とするアーキテクチャとシステムレベルの工学手法にある．

高性能でスマートなデバイス達が OWL-S のようなオントロジ言語を使って web サービスに対するセマンティックマークアップを解釈して，自立的にサービス連携させたり，また，オントロジによりマークアップされた web を膨大な知識を蓄えた知識ベースとして自在に活用して必要な情報をピンポイントに探し出したりすることを可能とする semantic web 技術が，コンポーネントを柔らかく合成するためのアーキテクチャと工学手法の重要な要素技術の一つであると筆者は考えている．

参考文献
［1］ T. Berners-Lee et al, Uniform Resource Identifiers（URI）: Generic Syntax, RFC 2396
［2］ Dave Winer, XML-RPC Specification, http://www.xmlrpc.com/spec
［3］ XML Protocol Working Group, SOAP Version 1.2, http://www.w3.org/2000/xp/Group/
［4］ R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, T. Berners-Lee, “Hypertext Transfer Protocol—HTTP/1.1”, RFC 2616, June 1999
［5］ Resource Description Framework（RDF）: Concepts and Abstract Syntax, W 3 C Pro-posed Recommendation 15 December 2003
［6］ James Hendler, Tim Berners-Lee and Eric Miller, Semantic Web におけるアプリケーション統合，pp. 676-680，電気学会誌　2002 年　Vol. 122, OCTOBER 2002 Volume 122 Number 10
［7］ Tim Berners-Lee, Notation 3, http://www.w3.org/DesignIssues/Notation3
［8］ RDF Semantics, W 3 C Proposed Recommendation 15 December 2003
［9］ OWL Web Ontology Language Semantics and Abstract Syntax, W 3 C Recommendation 10 February 2004

［10］ F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, and P. F. Patel-Schneider, editors. The Description Logic Handbook : Theory, implementation, and applications. Cambridge University Press, to appear.
［11］ The OWL Services Coalition, OWL-S 1.0, http://www.daml.org/services/owl-s/1.0/

執筆者紹介 山 田 繁 夫（Shigeo Yamada）
1983年日本ユニシス(株)入社．人工知能/オブジェクト指向モデリング/インターネット技術の研究，アプリケーション開発に従事．現在，日本ユニシス・ソフトウェア(株)サービスビジネス統括部データサイエンス技術部に所属．