セマンティックWEBの現状と課題

慶應義塾大学 環境情報学部
World Wide Web Consortium
萩野 達也

セマンティックWebは次世代のWebと期待されている技術の一つである．セマンティックWebは，これまでのHTML文書によるWebではなく，機械処理可能なメタデータの作るWebである．人が与えた問題を解決するためにエージェントはこのセマンティックWeb上で推論を行い結論を導き出す．セマンティックWebの概要について説明し，その現状および課題について述べる．

キーワード: セマンティックWeb，メタデータ，RDF，オントロジー

Semantic Web – its current status and problems –

Tatsuya Hagino
Faculty of Environmental Information, Keio University
World Wide Web Consortium

The semantic web is regarded as the next generation web. It is not like the current web which consisits of HTML documents, but it is a space consisting of metadata which machines can understand and process. In order to help people to solve problems, agent programs will do inference on the semantic web to deduce conclusion. In this paper, overview of the semantic web is presented and its current state and some problems are discussed.

keywords: Semantic Web, metadata, RDF，ontology

1 はじめに

セマンティック Web は次世代の Web と期待されている技術の一つである．提唱したのは Web の創始者でもある Tim Berners-Lee である．彼は 1990 年ごろに現在の Web を提唱し，1998 年ごろにセマンティック Web を提唱し始めた．セマンティック Web は従来の Web を置き換えるのではなく，従来の Web への追加の機能である．名前の通り,「セマンティック」すなわち「意味」を取り扱う Web 空間である．ここでは，このセマンティック Web に関して現状と問題点について取り上げる．

2 Web の発展

Web は現在第 3 の段階に入ろうとしていると言えるかも知れない．第 1 の段階の主役が **HTML** (HyperText Markup Language) であり，第 2 の段階の主役が **XML** (Extensible Markup Language)，そして第 3 の段階の主役がセマンティック Web である．

第 1 段階: HTML
↓
第 2 段階: XML
↓
第 3 段階: セマンティック Web

HTML と XML が文書のフォーマットであるのに対して，セマンティック Web だけが違った名称で違和感があるが，背景にある哲学をメインに出すための名称であるからである．実際にはデータフォーマットである **RDF** (Resource Description Framework) が中心的な役割を果たすと考えられる．

2.1 第 1 段階: HTML

HTML は Web ページの記述言語として考え出されたものである．**SGML** (Standard General Markup Language) のシンタックスを利用し，ハイパーテキストを記述することができる．HTML がインターネットのキラーアプリケーションの一つとなり，爆発的に普及をした．一時はインターネットのトラフィックはすべて HTML の転送の HTTP (HyperText Transfer Protocol) に埋め尽くされて，利用不可能になるのではないかと騒がれたが，HTTP 1.1 による TCP の利用効率の向上とプロキシーに対する細かな制御などの技術的な改良と，バックボーン回線の大容量化が進んだために，インターネットの危機とはならずにすんでいる．

HTML の普及にともない，商用のブラウザが開発され，それにともないいろいろな機能の追加が行われた．一時は，Web ページ毎に，これはあるブラウザで見ることができ，別のブラウザでは見ることができないなど，かなりの混乱が発生した．機能拡張も HTML の元々の設計思想にしたがったものであれば問題もなかったが，見た目だけを重視し設計思想を無視した拡張もかなりあった．HTML の設計思想のもっとも重要なものの一つが,

内容とプレゼンテーションの分離

である．HTML はあくまでも内容を記述したものであり，それがどのように見えるべきものであるかを記述するものではない．HTML の定義[1] を調べても，文書をどのように表示すべきかの定義は見当たらない．どのように表示すべきかは別のスタイルシート文書によって定義する．典型的なスタイルシートが **CSS** (Cascading Style Sheet) である．`CENTER`, `BLINK`, `FONT` などのタグは表示に関するタグであり，本来は使用してはいけない．逆に `EM`, `STRONG`, `DFN`, `CODE` などにより意味をマークアップすることができる．

内容とプレゼンテーションを分離することによって以下のような利点が生まれる．

- スタイルシートを変更することによって同じ文書を別に対して別の見せ方が可能になる．
- 同じサイトにある文書の同じスタイルシートを使うことによって統一的なデザインをすることができる．
- 表示するためのスタイルシートではなく，音声用スタイルシートを用いると，同じ文書を読み上げることも可能になり，アクセッシビリティの高いページを作成することができる．
- スタイルシートをユーザが定義し直すことによって，見る人にあったページにすることができる．アクセッシビリティを向上することができる．
- 一つの HTML により，PC だけでなく，携帯電話やテレビなどに表示することができるページを作成できる．PC 用，携帯用などと別

[1] `http://www.w3.org/TR/html401/`

けて作った場合には，内容に変更があった場合に，一つのHTMLを変更することによってすべてのページを変更することが可能になる．

このような考え方のもとに最終的に作られたのがHTML4.01であり，HTMLとしては最終的なものである．それ以後のバージョンは次に説明するXML化されたHTMLとなる．HTML4.01は，内容とプレゼンテーションを正しく分離したstrictと呼ばれる文法だけでなく，従来までのものをある程度許したtransitionalと呼ばれる文法，フレームを使うframesetと呼ばれる文法が定義されている．また，定義書の中には使うことを推奨しないdeprecatedと書かれた部分が多数ある．

2.2 第2段階: XML

XML[2]は，ビジネス等における文書形式として期待されて1998年に登場した．XMLはHTMLとはレベルの異なる言語である．XMLは文書型を定義することによって新しい文書形式を定義する言語である．XMLではDTD (Document Type Definition)により，文書を構成する要素としてどのようなものがあり，どのようなタグで識別され，その要素の中にどのような子要素を含むことができるのか，また，要素の属性としてどのようなものを書くことができるのかを定義する．DTDにより文書の集合を定義することができる．このようにDTDにより定義された文書のことをXMLアプリケーションと呼ぶことがある．XMLはインターネット時代におけるSGMLの改良版であるが，以下のような特徴がある．

- 大文字と小文字を区別する．
- タグの省略を禁止することによって処理を簡単にする．
- 空要素に関してはタグを工夫することにより区別する．
- DTDに従わなくとも整形式(well-formed)であればXML文書として取り扱う．

HTMLでは使うことのできるタグはHTMLの定義書に書かれていて，独自のタグを追加したりすることはできないが，XMLでは文書型を定義することによって自由にタグを使うことができる．より意味のあるタグを定義することによって文書の内容を明確にすることができる．しかし，XMLの登場によりHTMLが不要になったわけではない．HTMLはWebページを記述する言語として使い続けられる．XMLでは自由にタグを定義できるかも知れないが，自由度が高すぎるためにXMLの書き方のノウハウの蓄積は難しい．HTMLは決まったタグであり，すでに多数のHTMLのページが作られており，ノウハウも蓄積されている．その意味でHTMLは今後もWebページを記述する言語として存在し続ける．

HTMLはSGMLに基づいた文書であるが，XML化された`XHTML1.0`[3]が定義されている．今後インターネットで利用する文書はXMLに基づいたものに統一される．XMLに統一することにより，XMLの名前空間の機能を使うことによって，異なる文書形式により定義されたXML文書を混在させることが可能になる．すでにWebで用いるために，数式のための**MathML**[4] (Mathematical Markup Language), グラフィックスのための**SVG**[5] (Scalable Vector Graphics), マルチメディアの同期を記述する**SMIL**[6] (Synchronized Multimedia Integration Language)などがある．

HTMLで成功した概念もXMLに採り入れられている．ハイパーリンクをXMLの一般文書で取り扱うための**XLink**[7] (XML Linking Language), スタイルシートを取り扱う**XSL-T**[8] (XSL Transformations)と**XSL**[9] (XML Stylesheet Language)などである．さらにXMLによりすべてのデータの記述を目指して，DTDの代わりとなる**XML Schema**[10]，プロトコル記述のための**XML Protocol**，サービス記述のための**WSDL** (Web Services Description Language)などが考案されている．

2.3 第3段階: セマンティックWeb

HTMLは基本的には人が読むための文書である．そのため機械的に処理することが難しかったりする．検索エンジンなどはページの中身を解析しキーワードを取り出し，それによって検索を可能にしている

[2] `http://www.w3.org/TR/REC-xml`

[3] `http://www.w3.org/TR/xhtml1/`
[4] `http://www.w3.org/TR/MathML2`
[5] `http://www.w3.org/TR/SVG/`
[6] `http://www.w3.org/TR/smil20/`
[7] `http://www.w3.org/TR/xlink/`
[8] `http://www.w3.org/TR/xslt`
[9] `http://www.w3.org/TR/xsl/`
[10] `http://www.w3.org/xmlshema-0/`

が，文章の意味を機械が理解している訳ではないので，正確な検索であるとは言えない．たくさん出てきた検索結果の中から，人が一つづつチェックして最終的に必要な文書を探し出さなくてはいけない．

検索などのように，問題解決のために Web を用いることも多くなっている．旅行を計画する場合には，旅行会社に電話をせずに，Web 上で航空券を手配したりホテルを予約したりすることが可能である．病院の診察などの予約を受け付けてくれるところもある．Web を使った問題解決はますます重要となってきている．しかし，現在の Web では，完全な問題解決を自動的に行うことは難しく，複数の答えの中から人が苦労して正しい答えを探さなくてはならなかったり，複数のサイトを連携したような利用は手作業で行わなくてはならなかったりする．

たとえば,「札幌にある旅行代理店」を検索使用とした場合,「札幌」と「旅行代理店」の 2 つのキーワードを入力して検索する．検索結果の中には札幌にある旅行代理店もあるが，それ以外にも，札幌へのツアーを企画している旅行代理店も見つかる．あるいは，札幌に住んでいる人が書いた旅行代理店に関するページや，もしかすると札幌という名前の旅行代理店のページも見つかったりする．キーワードだけでは，札幌が北海道にある「地名」であり,「その場所にある」旅行代理店を探したいという検索している人の意図は伝わらない．

さらに,「今日営業している旅行代理店」を探したいとしたとき，検索エンジンは役にたたない.「今日」の意味も分からないし，それぞれの旅行代理店の「営業日」に関する情報も検索エンジンは持っていない．人がそれぞれの旅行代理店のページを見て判断するしかない．

また，もし検索エンジンをうまく使って札幌にある旅行代理店のリストを出すことができたとしても，どの旅行代理店が信頼できるかの情報は何も与えてくれない．個人の適当な判断によって選ぶしかない．互いの信頼性が確保できない状態では，インターネットにおける商取引は広がっていかない．

この問題を解決する一つの方法がセマンティック Web であり，セマンティック Web では機械的に処理することのできるデータ空間を別に用意する．HTML で書かれた文書を処理したのでは，自然言語であるため曖昧さがあり正確な処理はできない．その代わりに，HTML の文書が何を意味するのかを記述した**メタデータ**を処理する．セマンティック Web はメタデータによって構成されたデータ空間である．

たとえば，旅行代理店の Web ページのメタデータとして,「職種」として「旅行代理店」,「住所」として「北海道札幌市……」,「営業日」として「月〜金」などを与えておけば,「今日営業している札幌の旅行代理店」のページであるのかどうかは，人がページの中身を読むことなく，機械的に判断することが可能になる．

セマンティック Web 上のプログラムは，人の代わりになっていろいろな処理を自動的に行うため，**エージェント**と呼ぶことが多い．エージェントはメタデータを処理することのよりページの内容を理解することができ，人が発した複雑な質問に的確な答えを導き出すことができる．メタデータに関する処理は単なるパターンマッチだけではないため**推論**と呼ぶ．たとえば，旅行代理店によっては,「営業日」ではなく「休業日」をメタデータとして与えるかもしれない．正しく処理を行うためには,「1 週間が日〜土の 7 つからなっている」などの規則を使わなくてはならない．これまでの検索エンジンにおける and と or だけでなく，3 段論法や否定，述語論理で使っているような量化記号 (quantifier) もある程度取り扱える必要がある．

論理などを扱うため，セマンティック Web は人工知能技術の一つであると見なされることもあるが，人工知能関連の問題は難しく未解決の部分も多いため，Web に取り込むことにより Web 自身が使用不可能なものになると恐れている研究者もいる．しかし，人工知能の技術をそのまま使うのではなく，Web の特性や哲学に従ったものに変更して利用する．たとえば，人工知能では問題を完全に解決しようとするかもしれないが，Web では完全ではなくベストをつくせば良い．たとえば,「すべての旅行代理店をあげよ」という問題は解決しなくて良い．見つけることのできた適当な数の旅行代理店をリストすれば良い．量化記号の「すべて」という意味を厳密にとらえると解くことができなくなる．述語論理のもう一つの量化記号の「存在する」の方が Web では取り扱いやすい．

セマンティック Web では推論を行うため，その推論がどのように行われたかを表した「証明」や，推論が正しかったかどうかをもう一度「検証」したりすることが可能になる．エージェントは複雑な処理の後に結果を出すのであるから，それがどのようにして出されたのか確認することも必要になる．あるいは別のエージェントの出した結論が本当に正しいのか追試を行うかも知れない．

また，現在の検索エンジンではページのリンクが

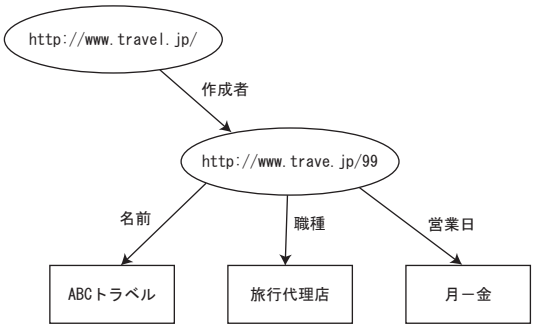

図 1

図 1: RDF によるメタデータ

結果として返されるだけであるが，セマンティック Web ではメタデータとして結果が返されるため，それを利用して別の検索や操作が可能になる．たとえば,「札幌から東京までの航空券を安く購入したい」場合，検索した旅行代理店のリストを調べ，その旅行代理店が札幌から東京までの航空券を販売しているか，また販売している場合いくらなのかを調べ，それらの中で一番安いものを見つけて実際に購入することになる．現在の Web でも航空券の購入が可能であるが，HTML のフォームと CGI や JSP など組み合わせて実現されている．HTML のフォームの場合には，それぞれの項目に何を入れれば良いのかは人が読んではじめて分かるようになっている．また，フォームの結果も HTML で返されてくるため，購入できたかどうかは人が読んではじめて分かる．セマンティック Web ではメタデータの形でサーバとやり取りするため，コンピュータが自動的に質問を発したり，結果を判断することができる．セマンティック Web では，札幌から東京への出張のための処理を，航空券の購入からホテルの予約，スケジュール調整まですべてを自動的に行うことができる．UNIX におけるパイプのように複数の処理を組み合わせることが可能になる．

3　メタデータ

Web のメタデータの起源は **PICS**[11] (Platform for Internet Content Selection) である．PICS では Web ページをある基準にしたがってレーティングし，その値によってページへのアクセスを規制するものである．PICS のレーティングが Web ページに対するメタデータである．PICS の場合には値として数字しか許していないが，もっと一般化し，あらゆる値をメタデータとして表現できるように考え出されたのが RDF (Resource Description Framework) である．

RDF ではそれぞれのメタデータを 3 つ組として表す．

(資源　属性　値)

「資源」は **URI** (Uniform Resource Identifier) で指し示すことができるものであり，Web 上のあらゆるものである.「属性」は属性を表す名前であり,「値」は文字列であるか別の RDF のメタデータである．値として複雑なものを表すためには，値を URI として表し，その URI に対していろいろな値を関係付けることによって表現する．たとえば，旅行会社の Web ページに対して次のように与えることができる．

```
("http://www.travel.jp/"
 作成者
 "http://www.travel.jp/99")
("http://www.travel.jp/99"
 名前
 "ABC トラベル")
("http://www.travel.jp/99"
 職種
 "旅行代理店")
("http://www.travel.jp/99"
 営業日
 "月-金")
```

このように RDF は関係を表したものであり，図のように有効グラフとして表すと分かりやすい．

RDF は 3 つ組であり，そのモデル自身は表現形式に依存しないが，データとしてやり取りを行うためのシンタックスが XML として定義されている．上記の例を XML シンタックスで表すと次のようになる．(正確には XML の名前空間の指定が必要であるが，ここでは簡単のために省略した．)

```
<RDF>
 <Description
  about="http://www.travel.jp/">
  <作成者 resource=
    "http://www.travel.jp/99">
 </Description>
 <Description
```

[11] http://www.w3.org/TR/REC-PICS-labels

```
    about="http://www.travel.jp/99">
    <名前>ABC トラベル</名前>
    <職種>旅行代理店</職種>
    <営業日>月-金</営業日>
  </Description>
</RDF>
```

RDF のメタデータは HTML のページの中に埋め込んでもかまわないが，別に分離しておいておくこともできる．PICS の場合にも，自分自身でレーティングを行うセルフレーティングと，レーティングビューローとよばれる第 3 者がレーティングを行うものがある．セマンティック Web においても，メタデータの付与は，ページの作成者が直接与えても良いし，別のところが与えてもかまわない．

どうしてメタデータとして XML を直接使わないのか疑問かもしれない．メタデータを XML で表すことは可能である．しかし，XML で表してしまうと，タグとして使う名前を固定することになったり，メタデータの書く順番を固定してしまうことになる．新しいメタデータの属性を追加したいと思うと，XML では DTD の定義から変更し，それを処理するプログラムを書き換えなくてはならない．また，複数のメタデータを集めて一つにすることも RDF では単に並べれば済むが．XML では DTD の制約を気にしなくてはいけない．

さらに，セマンティック Web では，全く別の形で定義されたメタデータの融合を可能にしたい．たとえば，作成者の名前を日本では「<名前>」をタグとして使ったが，欧米では「<name>」を使うかもしれない.「名前」と「name」が同じものを表すということを教えてやれば，これらを同等に扱ってほしい．また，日本では，名前は「姓」と「名」の 2 つからなるが，欧米では「first」,「middle」,「sir」の 3 つからなっているかもしれない．このような場合でも，欧米からの商取引情報として送られてくる名前のデータを自動的に日本の名前のデータに変換してほしい．

4 オントロジー

このようなデータ間の関係を表すのが**オントロジー** (ontology) である．哲学の分野ではオントロジーは存在論を意味し，存在することの意味を考察する学問であるが，人工知能や Web では用語間の関係の定義を意味している．機械にとって RDF の属性や XML のタグとして使っている「名前」や

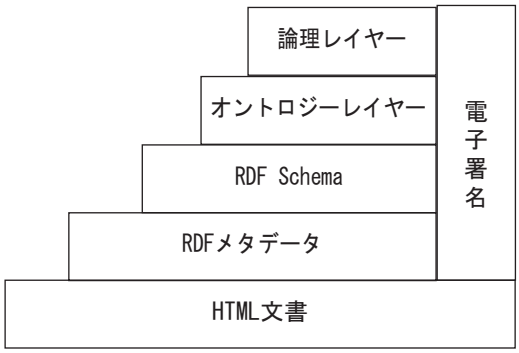

図２

図 2: セマンティック Web の階層構造

「name」は単なる文字列である．文字列だけであれば，それが一致するかどうかしか処理できないが,「名前」が「姓」と「名」からなっていることや,「name」に等しいこと,「nickname」は名前の特殊なものであるなどが定義されていれば，いろいろな処理が可能である．単に定義されている関係をたどるだけかもしれないが，それが推論である．

基本的なオントロジーは **RDF Schema**[12] としてすでに定義されている．RDF Schema では，RDF の 3 つ組の値の型を定義するものであるが，オブジェクト指向で用いいられるクラス階層を使って型を定義することができるようになっている．たとえば,「父親」は「両親」の一種であると定義することができる．これによって「父親」として与えられたデータを「両親」のデータだとみなして処理することが可能になる．

DAML+OIL[13] などで現在研究されている．また，World Wide Web Consortium においても Ontology Working Group が発足したところである．

5 セマンティック Web の今後の課題

セマンティック Web を構成する部品は図 2 として与え得られることが多い．人が読み書きを行う HTML による文書があり，その上に RDF を用いたメタデータのレイヤーがある．メタデータの定義としては，基本部分は RDF Schema が取り扱い，よ

[12] http://www.w3.org/TR/rdf-schema/
[13] http://www.daml.org/2001/03/daml+oil-index.html

り複雑な部分に関してはオントロジーレイヤーで取り扱う．その上には，メタデータを使った質問や推論を行うための論理レイヤーが置かれる．現在，RDF Schema までのレイヤーはほぼ完成した状態であるが，それから上の部分に関してはこれからである．また，メタデータを支えるためにもデータの信頼性などが問題になり，そのためには電子署名に関する研究も重要となる．これらのセマンティック Web の全体を実現することにより，Web における活動の信頼が生まれ，商取引も活発に行われる次世代の Web が実現される．

セマンティック Web の今後の課題としては，以下のようなことが考えられる．

- **オントロジーなどの仕様の確定**
  仕様が決まらないことには，いろいろなアプリケーションの構築が難しいため，中心となる仕様を早急に決める必要がある．特にオントロジーおよびその上での推論あるいは問い合わせに関する仕様を決める必要がある．

- **メタデータの付与**
  多くの Web ページにメタデータが付与されなくては，セマンティック Web は利用できない．自動的に付与することも考えられるが，単なる自動では現在の検索エンジンが行っていることとあまり変わらず，検索の精度を良くすることにはならない．Web ページを作るときに付与するのが一番である．ページの作者はどのような内容を書いているのは分かっているはずであるから，それをメタデータの形で与えてくれれば良い．また，多くのページがデータベースなどから機械的に生成されていることも多いが，その場合には，機械的なメカニズムに HTML だけでなくメタデータも同時に生成できるように変更すれば良い．機械的処理はデータベースの意味を解釈して HTML を生成しているはずであるから，メタデータを生成するものそれほど難しいことではないと思われる．

- **ツールの開発**
  Web ページを作成するときにメタデータを付与するのを助けるため，オーサリングツールではメタデータの入力を促す機能の追加が望まれる．Web サイトを構築する場合には，取り扱うデータに関するオントロジーの設計からはじめる必要があり，すでに存在するオントロジーのブラウズやその一部の利用，変更や追加ができ，それを利用した各種のデータを取り扱い Web サイトを構築するためのサーバソフトウェアが望まれる．

- **エージェント・スクリプト**
  クライアント側ではエージェントがユーザの要求を，セマンティック Web のメタデータを利用して推論することによって解決する．エージェントにどのようなことをさせるのかスクリプトのようなものを与えるかも知れない．あるいは推論のための論理式を与えるのかも知れない．どちらにしてもエージェントにユーザの意図を間違いなくしかも簡単に伝える機能が必要である．

- **アプリケーション分野の開拓**
  セマンティック Web は分散アプリケーションの新しいプラットフォームである．これまでの Web だけでなく，家電ネットワークや電子マーケットプレース，電子政府などあらゆる分野で応用可能である．

6 おわりに

ここでは，セマンティック Web について，その概要について説明した．セマンティック Web はいまだ普及していないが，もしそのメリットが明確になれば急速に普及すると思われる．TCP/IP がインターネットの基盤となるプロトコルであるように，セマンティック Web は拡張性や将来性を考慮した分散アプリケーションの基盤となるプロトコルになっていくであろう．

参考文献

1. 「自分で推論する未来型ウェブ」T. Berners-Lee, J. Hendler, O. Lassila. 日経サイエンス 2001 年 8 月号 pp. 54-65.
2. Cooking up the Semantic Web，IEEE Intelligent Systems, 2001 年 3/4 月号.
3. World Wide Web Consortium セマンティック Web アクティビティ: `http://www.w3.org/2001/sw/`