

< 研究ノート >

Semantic WebとRDF

山島一浩

Semantic Web and RDF

YAMASHIMA Kazuhiro

キーワード：Semantic Web RDF XML 情報交換

1. はじめに

Web上に氾濫しているHTMLで記述されたデータは、タイトルや段落など文書構造の見せ方を重視したタグで表現されたドキュメントである。このためHTMLタグを用いてドキュメント全体のデータ構造を特定することは難しい。また、データ抽出処理でタグから意味的な解釈とその処理を行うには、情報が不十分である。

情報源に格納されている情報のデータ形式は、多様であり、文書以外にも動画や音楽等のデータや複合的に構成されるマルチメディア形式のものも含まれるため、検索には、様々なデータ形式を識別して的確な情報を得るための手段を講じなくてはならない。

Webの普及以降の情報検索は、人間による多くの作業時間と多大なコストが生じており、さらには、その結果として得られた情報が実は利用者の求めるものと一致しない等のリスクも存在するのである。

本稿は、Web上のデータ処理を機械で実現

するためのプロジェクトであるSemantic Webについて、XML(eXtensible Markup Language)規格間に存在する課題、メタデータ記述と、RDF(Resource Description Framework)[RDF]に着目して概観する。そして、その実現に向けて克服しなければならない課題について検討する。

2. 機械のためのWebシステム

ある目的で記述されたデータを機械的に処理できるようにW3C(World Wide Web Consortium)で勧告したXMLを用いたデータモデル構造の定義とそれに従ったデータ記述が行われるようになった。XMLでは、データモデルをDTD(Document Type Definition)やXMLschemaを用いて、自由に定義することができる。その定義された枠組みの中でデータモデル構造を特定し、機械的な処理を可能にする。そして多くの分野でこのXMLを用いた標準化作業が行われている。XMLが標準化の手法として認知されている表れであるとい

える。

このXMLで記述されたデータモデルが普及し流通することを想定すると、その次のステップにあるのは、データモデル構造に従って記述されたデータが機械的に的確に処理され、その結果として確実な検索が行える機能をどのように整備するかである。

定義されたデータモデルがどのような構造であるのかは、DTDやXMLスキーマの参照で判断することができる。しかし、それ以外の、使おうとするデータについての利用環境など、二次的な情報は、読み取ることはできない。DTDやXMLスキーマは、データモデルの構造を定義する記述であるからである。

メタデータは、或るデータモデル構造をしたデータが蓄積された情報源に的確に迎れるようにする仕組みを担ったデータのためのデータである。データについての要約を記述し、それを用いて、情報源への確に辿り着けるようにしようというものである。

ここで述べるメタな情報とは、例えばデータを利用する際に、リンク先のURIやどのブラウザに対応しているか、どのモニタなら十分に表示できるのか等である。このような内容が記述された情報がなければ、データを機械的にすぐに利用できない。

現在、メタデータ記述の検討は、多くの分野で行われている。

RDF (Resource Description Framework) は、メタデータを組み込んで、Web上で情報交換するためのアプリケーション間の相互運用性を提供する目的で、資源の記述の枠組みを与えるXML仕様の一つである。

そしてSemantic Webは、このRDFを組み込み、様々なデータモデルが蓄積された情報源を機械的に相互運用できるようにした、次世代のWebシステム・モデルである。

Semantic Webが実現されると、人間と機械、又は機械と機械とのコミュニケーションが可能になることが期待されている。例えばWeb

システム間をエージェントプログラムがRDFの記述で枠組みされたメタデータ間を辿りながら該当する全ての情報源を検索し、それらからもっとも適した情報を利用者に提供できるようにする。

次にXML検索の課題について、具体的な説明を少し加えておく。

3 . XML規格間のコミュニケーション課題

1998年にW3Cで勧告されたXMLは、意味構造を定義することができる汎用的なマークアップ言語である。XMLではタグを自由に定義できる。つまり、データ構造の意味的な部分をXMLで自己記述できるのでプログラム処理を容易にする。

W3Cでは、このXMLをベースに既に多くの仕様が勧告されている。例えば、数式の記述に用いるMathML[MathML]、ベクターグラフィックスを記述するためのSVG[SVG]、同期マルチメディア統合言語のSMIL[SMIL]等がある。

ここではSVGとSMILについて述べる。

図1は、SVGドキュメントのリストである。SVGは、SVGタグをルートタグとして、グラフィックス表示用タグ内に記述された座標タグにより構成されるグラフィックス表示のためのドキュメントである。

そして、図2が、その出力された画面である。

このグラフィックス画面は、拡大や縮小などに対応し、ある範囲に色を塗りつぶすこともできる。この場合、色を塗るということは、PNG画像やJpeg画像のように、ある範囲のピクセル情報に対して色情報を付加するのではなく、ポリゴン図形内に対して塗りつぶす点情報をもとに色情報を付加する。このためレイヤー処理も容易である。

図3は、SMILドキュメントのリストであ

```

<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 20000303 Stylable/EN"
"http://www.w3.org/TR/2000/03/WD-SVG-20000303/DTD/svg-20000303-stylable.dtd">
[<ENTITY ln1 "fill:none;stroke:#000000;stroke-width:300">]
<ENTITY ln2 "fill:none;stroke:#000000;stroke-width:250">]
<ENTITY ln3 "fill:none;stroke:#000000;stroke-width:200">]
<ENTITY ln4 "fill:none;stroke:#000000;stroke-width:150">]
<ENTITY ln5 "fill:none;stroke:#000000;stroke-width:500"> ]];
<svg xmlns:svg="preserve" width="300" height="240" viewBox="000 000 80000 80000 "
preserveAspectRatio="none">
<polyline style="fill:#000;stroke:#000;stroke-width:300" points="
7296 80000 7217 79983 7178 79761 7225 797381
7200 79654 7324 79561 7325 79572 7409 80000 7437 79989 7514 79979 7609 799741
7620 79966 7637 79935 7676 79845 7655 79799 7684 79733 7705 79670 7736 796081
7706 79573 7823 79525 7578 79486 7575 79463 7508 79361 7565 78992 7631 794731
途中省略]
50333 4148 50292 4135 50243 4105 50207 4075 50161 4026 50143 3999 50134 39721
50121 3888 50063 3804 50016 3722 50000 3703"/></svg>]

```

図1.SVGドキュメント

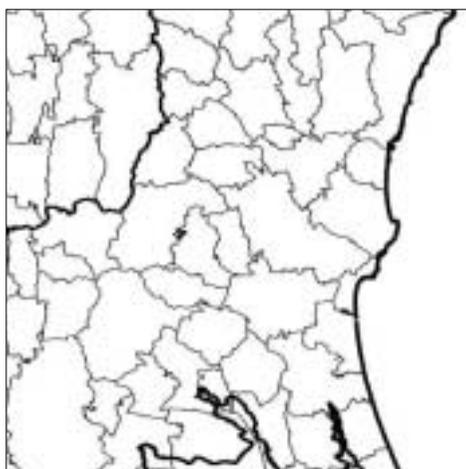


図2 . SVGドキュメントのブラウザ表示

```

<smil>]
<head>]
  <layout>]
    <root-layout height="240" width="320"/>]
    <region id="img_res" left="0" top="0" height="240" width="320" />]
    <region id="text_res" left="0" top="350" height="20" width="350" />]
  </layout>]
</head>]
<body>]
  <par>]
    <audio src="03.m"/>]
    ]
  </par>]
</body>]
</smil>]

```

図3 . SMILドキュメント

る。SMILは、テキスト、音声、静止画、動画をシンクロさせる為のタグを用いて、時間の同期をとりながら表示を制御するマークアップ言語である。

図4は、SMILドキュメントをブラウザ表示したものである。



図4 . SMILドキュメントのブラウザ表示

ここでこの例を用いたのは、SVGが幾何情報で構成されるデータモデルであり、SMILは制御系を記述したデータモデルだからである。

この場合の仕様では、構成するデータモデル構造についてアプリケーションプログラム側で処理できる書式であり、その目的においては完結された情報である。しかし、その中の映像部分を二次的に利用しようとする場合、対象とするタグに記述されている内容を何を表現しているのかを機械的に特定する情報は無い。

例えば図3のSMILで記述されている動画部分を利用する場合、対象は動画ファイル名であるが、機械的に判断するには、その動画ファイルが何を表現しているのかを記述した情報が必要である。しかし、それを記述する箇所は無い。

これが、規格間での機械によるコミュニケーションを図ろうとする場合の困難さを表した顕著な課題といえる。このためデータの再

利用を考慮した情報を得るために、データモデルに対しての意味的な記述を行うメタデータが必要になるのである。

4 . メタデータ

メタデータは、データについて記述された構造化されたデータである。メタデータの成果として図書目録を扱ったMARC STANDARDSが挙げられる[MARC]。MARC STANDARDSは、高度な知識と経験を積んだ専門家によるカタログ化が行われたことで、的確な分類がなされた完成度の高いメタデータである。そして、その利用目的も限定的である故に、成功しているメタデータといえる。

しかし、Web上には、多様なデータモデルが存在し、それに沿って記述されたドキュメントについて記述するメタデータには、より汎用的な記述が求められる。一方、そのメタデータを誰もが正確に記述することは可能なのだろうか。

仮に将来機械的に作成することが可能になるとしても、それには、少なくともその記述の枠組みが必要である。また、或る定義されたデータモデルに沿って記述した内容についてのメタデータは、その構造に応じて作成した全ての人に最低限の内容で記述してもらえばよいという意見もある。

これらのことが考慮され、先行したメタデータ記述として挙げられるのがDublin Core Metadata Element Set[DC]である。Dublin Core Metadata Element Setは、通称Dublin Coreと呼ばれ、1995年3月に開かれたOCLC (Online Computer Library Center) /NCSA (National Center for Super computing Applications) が主催したワークショップの開催地オハイオ州ダブリンにちなんで付けられた名前である。

Dublin Coreの現在の版は、1.1である。Dublin Coreは、以下に示す15のコアとなるエ

レメントを用いてドキュメントのメタ情報を表すことを定義している[杉本]。

- (1) タイトル：Title
- (2) 著者あるいは作者：Creator
- (3) 主題およびキーワード：Subject
- (4) 内容記述：Description
- (5) 公開者（出版者）：Publisher
- (6) 寄与者（他の関与者）：Contributor
- (7) 日付：Date
- (8) 資源タイプ：Type
- (9) 形式（フォーマット）：Format
- (10) 資源識別子：Identifier
- (11) 情報源（出处）：Source
- (12) 言語：Language
- (13) 関係：Relation
- (14) 対象範囲（空間的・時間的）：Coverage
- (15) 権利管理：Rights

それぞれのエレメントは、対象とするドキュメントの「内容」、「知的所有権」、「インスタンス生成」についてのメタ情報を表している。

「内容」を表すエレメントは、Title、Subject、Description、Type、Source、Relation、Coverageである。

「知的所有権」を記述するエレメントは、Creator、Publisher、Contributor、Rightsである。「インスタンス生成」のエレメントは、Date、Format、Identifier、Languageである。

Dublin Coreの15エレメントは、ネットワークで提供されるリソース記述のスキーマを完全に表したのではない。何故ならば、この15エレメントは、専門家以外にも記述できるようにした簡略化された最小限のフィールドだからである。

故に、これだけでは情報を特定するには不十分であり意味を十分に表現できていないという要求もでてきた。そして現在、15エレメントに対して、詳細なqualifier[DCqualifiers]を付加するようになっている。

この処置は、例えばSubjectについて、どの統制語彙を用いたものであるのかをLCSH、MeSH、DDC、LCC、UDCをqualifierとして記述してやることで、より限定された意味を表現することができるようにするものである。現在も進められているDublin Coreの活動は、独自のメタデータを設計する際に、参考となる活動といえる。

メタデータの役割は、データモデルを適切に表現し、それを利用目的と合致してメタデータ検索に繋げるものである。このため用途によりメタデータの利用目的も異なる。つまりメタデータ記述が全てDublin Core的記述に集約されてしまう訳ではない。例えば、前項で指摘した課題（ブラウザの選択や画面サイズを規定するメタデータ）の記述については、CC/PPという複合能力と指向性プロファイルとしてW3Cで検討されているメタデータがある。またマルチメディア検索のための記述規格として検討されているものに、MPEG-7がある。

このようにWeb上のメタデータは、データモデルにアクセスするための手がかりとなるデータで構成される。そしてその利用目的により必要に応じた内容の記述がとられ、結果的にその記述は、複数のポキャブラリとして存在することになる。

5 . RDFについて

これまで述べてきたメタデータとデータモデルとを結びつける動きをするのがRDFである。

RDFは、メタデータを使った高機能な検索機能やドキュメント間の関係記述をするカタログ作成、情報共有及び情報交換を促進する知的ソフトウェアエージェント機能、内容の格付け、知的財産権の記述、ユーザ・プライバシーに関する個人の趣向表現、ウェブサイトのプライバシー保護方策表現などを実現す

る手段として期待されている仕様である。

XMLベースでのメタデータ記述のためのモデルと構文を記述した仕様として1999年に勧告されたのがRDF Model and Syntaxである。これによるとRDFは、資源 (Resources)、特性 (Properties)、文 (Statements) の三つのオブジェクト型で構成されたラベル付き有向グラフで表現される。例えば、RDFを用いると、次のような言明ができる。

「山島は、<http://www.kasei.ac.jp/cs/mt-is/>の作者である。」

この表現についてのグラフは、[図5](#)のように表され、RDFでは、[図6](#)のように記述される。

RDFでは、複数のメタデータを扱うために名前空間でメタデータを識別する。例えば、

Dublin Core Metadata Element Setの場合には、名前空間にDCを用いることが奨励されている。

RDFのグラフ構造により、メタデータからそのデータへたどり着くための手段が提供される。

6 . Semantic Webの構造

W3CにおけるSemantic Webの活動は、Webの開発者Tim Berners-Leeが1998年に示したRoadmap[Berners-Lee1998]から、2001年2月に正規の活動として認められた経緯を持つ。

現在示されているSemantic Webの構造は、[図7](#)で示すような階層構造である。

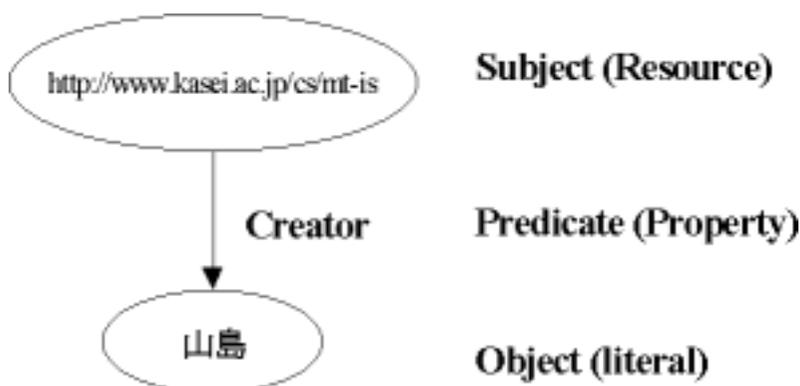


図5 . グラフ表現

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:s="http://description.org/schema">
  <rdf:Description about="http://www.kasei.ac.jp/cs/mt-is">
    <s:Creator>山島</s:Creator>
  </rdf:Description>
</rdf:RDF>
```

図6 . RDFの記述

最下層のUnicodeとURIは、内容の記述のための文字コードとWeb上でアドレスを記述するためのデータの内容記述を行うための基盤となる仕様である。

Unicodeは、多言語を扱うための統一的な文字コードを規定する文字コードである。Unicodeは、Unicodeコンソーシアムで検討されている。

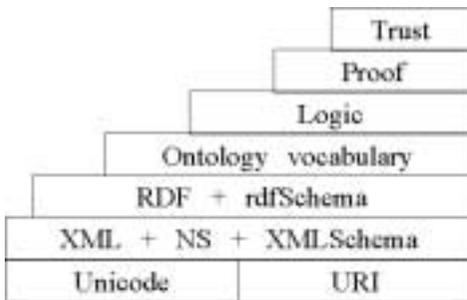


図7 . Semantic Webの構造図[Berners-Lee2000]

URIは、コンテンツをWeb上でアドレスできるようにするためのものであり、URLやURNを含めたアドレスの概念である。

この基盤仕様の上に、XMLschemaで定義されるデータモデルとその構造に従って記述される「XML+NS+xmlschema」層がある。この層では、XMLの記述ルールに従ったデータモデル構造をXMLschemaやDTDで定義し、その構造に従ってデータは、格納される。ここで、XML規則によれば、各データモデルは一つのファイルに混在して複数のデータモデルを記述できる。このため、構造の異なる仕様の中で、同じ名前のタグ名の使用も考えられるので、それを区別するためにNS (Name Space) と呼ばれる名前空間を使用する。

このデータに対するメタデータとの関係を記述するのが、その上の「RDF+rdfschema」層である。RDFの役割については、前述した通りである。

ここまでの層、すなわちXMLやRDFにより、知識化した情報が表現されるが、その情報を一つの語彙で表すことには無理がある。情報をどの観点で見るかにより、規定された概念に基づく語彙が複数存在するからである。その概念間の定義や階層関係を記述するのが「Ontology Vocabulary」層である。この層では、概念間の関係を記述するオントロジー言語が利用される。現在、オントロジー言語として注目されているのは、DAML+OILである。DAML+OILは、アメリカで開発されてきたDAMLと欧州で開発されてきたOILが、2001年3月に一つに統一されたものである。DAML+OILは、RDFschemaをベースにしたオントロジー記述を行う仕様である。

「Logic」層では、RDFでは表現できない論理表現を記述できるようにし、知識から新しい知識を推論するための仕組みを提供する。このための汎用的な推論プログラムを実装できるような環境の構築を目指している。

Semantic Webを実現するためのXML関連の仕様の勧告状況は、RDF階層までが、ほぼ完了しており、現在は、オントロジー階層以上に、関心が集まっている。

7 . Semantic Webの実現について

Semantic Webの実現に期待を寄せる反面、実現は懐疑的であるという意見もある。ここでは、Semantics Webの実現に向けて考慮すべき要因について、次の三点を指摘する。

7.1 基盤仕様の完成度

如何に完成度の高い標準仕様が策定されても、それに従った利用の枠組みが完成しなければ、淘汰されてしまうのが今日の状況である。さらに仕様の版更新が継続的に行われれば、普及を遅らせる原因にもなる。

W3Cで勧告されたドキュメントは、現段階においても、名前空間や、言語混在問題等

仕様の不具合が指摘され、継続的な版更新作業が行われている。

7.2 企業間競争の思惑

また、W3Cの活動は、Web上でデータモデルを相互利用するための共通の仕様を策定するものであり、基本的には、その普及活動を行っている団体ではない。W3Cで勧告されたドキュメントの実用的な普及を支援する実用化を推進するのは、OASYS[OASYS]である。OASYSは、非営利の、国際的なコンソーシアムである。参画する団体は、97のSponsors、91のContributors、13のLiaisonからなる(2001年10月時点)。このため、市場を優位にすべくキラアプリケーションを誕生させる為の企業間競争の思惑が少なからず働くことになろう。

このためSemantic Webについても、その構成の中で用いられる処理言語系の組み合わせについての汎用性を認める設計となりうることから、幾つかのSemantic Webと称するWebシステムが誕生しかねない。そこで懸念されるのは、Semantic Web間でのプログラム間のやり取りで双方がコミュニケーションが取れないことである。人間間で起きているこのような現象が機械間コミュニケーションでも同様に起こることが危惧される。

7.3 データ量の肥大化

標準化の目的は、技術の共通化を図ることでモノや情報の大量生産を可能にし、生産コストを下げ、流通コストの低下をもたらせることにある。

標準化されたネットワークは、相互接続の容易性、利用・運用方法に関する共通の知識の共有により、様々な利便性と効用性を実現する手段である。

これまででは、データは知的な情報としての構造化がなされておらず、垂れ流し状態であった。そして知的な情報を発掘するのは、人

間の作業に委ねられていた。故に機械処理の可能性を探る要求は必然性を帯びている。

機械処理を行うため、知識を表現するデータモデル構造定義を行う。それに従いデータを記述する。そして、メタデータを記述する。これまでの仕組みの記述は、XMLschema、XMLとRDFで行う。

そして、その実現は、メタデータを含めた総データ量が、肥大化することを意味する。このため、より一層高速なネットワークの普及と高速化の技術の確立も不可欠である。

8 . まとめ

Semantic WebとRDFについて、メタデータを軸として概観し、幾つかの要因を取り上げて考察を行った。

Semantic Webは、データとそのデータのためのメタデータ記述とを機械的に処理するための次世代Webシステムモデルである。

Semantic Webは、単独のサイト構築のためのシステムモデルとして捉えるのではなく、Web上で分散して存在するシステムが意味処理を行う上で協調して機能するWeb全体を見通したシステムモデルと捉えるべきである。それ故に、その実現には、Webが爆発的に普及した時と同様の起爆剤が必要となる。

【参考文献】

- [RDF]<http://www.w3.org/TR/1999/REC-rdf-syntax-19990222/>
- [MathML]<http://www.w3.org/Math/>
- [SVG]<http://www.w3.org/TR/SVG/>
- [SML]<http://www.w3.org/TR/smil20/>
- [Berners-Lee1998]<http://www.w3.org/DesignIssues/Semantic.html>
- [DC]<http://dublincore.org/documents/dces/>
- [杉本]<http://www.dl.ulis.ac.jp/DC/RefDsc/DCS-J2.html>

[DCqualifiers]<http://dublincore.org/documents/dcmes-qualifiers/>

[1206-xml2k-tbl/slide10-0.html](http://www.w3.org/2006/xml2k-tbl/slide10-0.html)

[OASYS] <http://www.oasis-open.org/>

[Berners-Lee2000]<http://www.w3.org/2000/Talks/>