国際経営・文化研究　Vol.12 No.1　November 2007

（論　文）

Semantic Webにおける Resource Description Frameworkの役割

大　武　信　之

キーワード

Resource Description Framework　Semantic Web　HTML　XML　W3C

1．はじめに

近年は、様々な調べ事をする際に、辞書で調べるとか、図書館で調査するより、とりあえずインターネットで調べることが多くなっている。これはインターネットが使える環境においては、専門分野や世代間の差などの違いは無く、世界的に共通な行為となっている。インターネットで調べる、あるいはホームページで調べるという行為は、どちらも何らかの検索エンジン（検索ソフト）を用いて、全世界に張り巡らされたWWW（World Wide Web）上に存在するHTML（Hyper Text Markup Language）[1] ファイルに書かれている内容を調べることを指す。従って調べられる範囲は、検索エンジンが抽出した内容の一部であり、異なる検索エンジンを組み合わせて調べても、調査可能な範囲は検索エンジンの結果による。それも検索者のキーワードの設定により得られた結果であるため、キーワードの選び方や、検索方法も問題となる。検索者は、キーワード漏れは無いか、キーワードが適切か、検索方法の組み合わせは良いのかという不安を持ちつつも、得られる検索結果が膨大であるため、ある程度の求め解が満たされるため、辞書や図書館等で調べる以前にWeb検索を利用しているのが現状である。より良い検索の手段はないか、これだけの量の情報を知識として蓄えられないか、またそして、それらを自動化できれば、使い勝手の良い環境が得られるはずである。

Web上に存在するHTMLは、人間が利用するために作られたもので、コンピュータが理解可能な形式ではない。Webで何らかの操作を自動化することは難しく、Webが含んでいる情報量の多さのために、人間が管理することも困難になってきた。そこで提案された解決策が、Webに含まれるデータを記述するために**メタデータ**を使うことである。メタデータは、**データについてのデータ**のことで、例えばライブラリのカタログは、出版物について記述しているのでメタデータにあたる。取り分けこれから述べる本論での扱いは「Web資源（リソース）を記述しているデータ」のことである。データとメタデータの間の区別は、厳格なものではない。それは特定のソフトウェア（アプリケーション）によって初めて生み出される区別で、同一の資源に対して、２つの解釈がなされることも起こる。

おおたけ　のぶゆき：国立大学法人・筑波技術大学　教授（理博）

Resource Description Framework（RDF）[2] はメタデータを処理するために、World Wide Web Consortium（W3C：注１）[3] により定義されたデジュール・スタンダード（注２）で、Web上でコンピュータが理解可能な情報を交換するアプリケーションの間に互換性を提供するものである。W3Cでは、RDFがWeb資源（リソース）の処理の自動化を容易に実現できることが強調されている。RDFはいろいろな応用領域で使うことができ、W3Cには次のような例が挙げられている。

1．より良いサーチ・エンジン能力を提供するリソースリカバリー
2．特定のWebサイトやWebページ、ディジタル図書館においてアクセス可能なコンテンツとコンテンツ関係の記述のためのカタログ作り
3．知識の共有と交換を容易にするための知的なソフトウェアのエージェント
4．コンテンツレイティング
5．論理的に単一なドキュメントを表すひとまとまりのページ記述
6．Webページの知的財産権記述
7．Webサイトのプライバシーポリシーとユーザのプライバシープリファレンス表現

上記はRDFの一例で、ディジタル署名を持ったRDFは、今後より利用の機会が増すと予想される電子商取引や、他のアプリケーションのための信頼できるWebの構築をなす鍵となる。

2．Resource Description Framework

まずRDFメタデータを表すためのモデルと、RDFメタデータのコーディングのための文法を示す。RDFの記述方法を示すため、XML（Extensible Markup Language）[4] が使われるが、ここでは理解を容易にするためEBNF（Extended Bucks Normal Formula）を用いる。なお斜体で書かれた部分（例：*rdf*）はより正確なBNF（Bucks Normal Formula）記法より、可変的なネーム空間接頭辞を表すために使用している。

```
[1] RDF             ::= ['<rdf: RDF>'] obj* ['</rdf: RDF>']
[2] obj             ::= description | container
[3] description     ::= '<rdf: Description' idAboutAttr? bagIdAttr? propAttr* '/>'
                      | '<rdf: Description' idAboutAttr? bagIdAttr? propAttr* '>'
                            propertyElt* '</rdf: Description>'
                      | typedNode
[4] container       ::= sequence | bag | alternative
[5] idAboutAttr     ::= idAttr | aboutAttr | aboutEachAttr
[6] idAttr          ::= ' ID="' IDsymbol '"'
[7] aboutAttr       ::= ' about="' URI-reference '"'
[8] aboutEachAttr   ::= ' aboutEach="' URI-reference '"' | ' aboutEachPrefix="' string '"'
[9] bagIdAttr       ::= ' bagID="' IDsymbol '"'
```

2

```
[10] propAttr          ::= typeAttr
                           | propName '="' string '"'
[11] typeAttr          ::= ' type="' URI-reference '"'
[12] propertyElt       ::= '<' propName idAttr? '>' value '</' propName '>'
                           | '<' propName idAttr? parseLiteral '>' literal '</' propName '>'
                           | '<' propName idAttr? parseResource '>'
                             propertyElt* '</' propName '>'
                           | '<' propName idRefAttr? bagIdAttr? propAttr* '/>'
[13] typedNode         ::= '<' typeName idAboutAttr? bagIdAttr? propAttr* '/>'
                           | '<' typeName idAboutAttr? bagIdAttr? propAttr* '>'
                             propertyElt* '</' typeName '>'
[14] propName          ::= Qname
[15] typeName          ::= Qname
[16] idRefAttr         ::= idAttr | resourceAttr
[17] value             ::= obj | string
[18] resourceAttr      ::= ' resource="' URI-reference '"'
[19] Qname             ::= [ NSprefix ':' ] name
[20] URI-reference     ::= string, interpreted per [URI]
[21] IDsymbol          ::= (any legal XML name symbol)
[22] name              ::= (any legal XML name symbol)
[23] NSprefix          ::= (any legal XML namespace prefix)
[24] string            ::= (any XML text, with "<", ">", and "&" escaped)
[25] sequence          ::= '<rdf: Seq' idAttr? '>' member* '</rdf: Seq>'
                           | '<rdf: Seq' idAttr? memberAttr* '/>'
[26] bag               ::= '<rdf: Bag' idAttr? '>' member* '</rdf: Bag>'
                           | '<rdf: Bag' idAttr? memberAttr* '/>'
[27] alternative       ::= '<rdf: Alt' idAttr? '>' member+ '</rdf: Alt>'
                           | '<rdf: Alt' idAttr? memberAttr? '/>'
[28] member            ::= referencedItem | inlineItem
[29] referencedItem    ::= '<rdf: li' resourceAttr '/>'
[30] inlineItem        ::= '<rdf: li' '>' value </rdf: li>'
                           | '<rdf: li' parseLiteral '>' literal </rdf: li>'
                           | '<rdf: li' parseResource '>' propertyElt* </rdf: li>'
[31] memberAttr        ::= ' rdf: _n="' string '"' （ここで n は整数）
[32] parseLiteral      ::= ' parseType="Literal"'
[33] parseResource     ::= ' parseType="Resource"'
[34] literal           ::= (any well-formed XML)
```

3

RDFの目標の１つが、標準化された互換性のある方法で、XMLに基づいたデータのために意味（セマンティク）を指定できるようにすることである。RDFとXMLは補完的で、RDFはメタデータのモデルで、リファレンスにより移植とファイル保存を必要とする国際化や文

字セット等のコーディング問題について述べているだけである。RDFの目標は、特定のアプリケーション・ソフトウェアに偏ること無く、資源（リソース）を記述するための機能を定義することである。RDFの機能定義はドメインに対して中立的で、機能はどのドメインに対してもの情報記述に適したものでなければならない。

上記RDFは、オントロジーを構築する道具である。哲学で用いられるオントロジー（Ontology）は存在論であるが、これが派生し人工知能分野をはじめとするコンピュータの世界では、概念化の明示的な仕様と言われる。Webをはじめとした文書検索において、従来の方法では単語単位での一致か、よくても類義語を含む文書を検索するのが限度であった。ここにオントロジーの概念を導入して、それぞれの文書の内容を説明する意味情報（メタデータ）を各文書に付加し、メタデータを記述する用語を定義する構造を構築する。この構造がオントロジーであり、オントロジーを導入することにより、検索対象となる文書が単なる単語の集まりとしてではなく、文書全体で大きな意味を持ったデータとして扱われ、各文書について統一的な付加情報をもたせることができる。このように、メタデータとオントロジーの技術を用い、文書の意味に即した処理をコンピュータが行うことが出来るWebをSemantic Web[5]と呼び、次世代の検索技術が実現されることなどで期待されている。Webにおけるオントロジーでは、本当に必要な情報を的確に検索することが可能となる。インターネット上に存在するオントロジーを用いて出版を行うためのマークアップ言語OWL（Web Ontology Language）が存在するが、OWLはRDFの語彙拡張であるため、ここでは取り上げない。なお、Web Ontology Languageの本来の訳語はOWLではなくWOLではないかと言われるが、これはクマのプーさんに登場するOwl（フクロウ）が自分の名前をWOLと書くことにちなんで名づけられたと言われる。

では実際どのような記述になるか、以下W3Cに示されたMathMLを含む例を示す。MathML[6]は、XMLにおける数式記述を定義したマークアップ言語で、HTMLやXMLに埋め込まれて記述される。

```
<rdf: Description
        xmlns: rdf="http: //www.w3.org/1999/02/22-rdf-syntax-ns#"
        xmlns: dc="http: //purl.org/metadata/dublin_core#"
        xmlns="http: //www.w3.org/TR/REC-mathml"
        rdf: about="http: //mycorp.com/papers/NobelPaper1">

        <dc: Title rdf: parseType="Literal">
        Ramifications of
            <apply>
                <power/>
                <apply>
                    <plus/>
                    <ci>a</ci>
                    <ci>b</ci>
                </apply>
                <cn>2</cn>
```

4

```
        </apply>
    to World Peace
    </dc: Title>
    <dc: Creator>David Hume</dc: Creator>
</rdf: Description>
```

上記サンプルだけでは、何が意味記述であるか理解しがたい。次節において、意味のある記述および意味の無い記述について具体例を用いて示し、Semantic Webに求められているものを明示する。

3．Semantic Web

これまでHTMLはページ記述の自由度が高い反面、利用されるタグに統一性が無いという問題があった。第1世代のHTMLに対し、第2世代のページ記述言語であるXMLは、構文（シンタックス）レベルで統一を目指し、W3Cを中心としてXMLをベースとした様々な標準化規格が提案され、アプリケーション分野を中心として広く利用されるようになってきた。Semantic Webはさらに意味（セマンティックス）レベルでの統一を目指そうとする試みである。意味レベルでの統一が実現できれば、Web情報の機械的処理がより一層容易になり、Web情報の検索や統合が格段に進歩することが期待される。Semantic Webに関する標準化活動は現在の米国や欧州において積極的に進められ、日本でも大学と企業の連携が進んでいる。Semantic Webで重要視されるオントロジーをはじめとするWeb意味論の基礎、社会システム応用のための方法論、さらにはインターネット上での新しい社会基盤作り等についての研究・応用が進む。本節においては、Web上の意味について考察する。

3.1　意味記述

数式を記述する手段の1つに、米国数学学会が登録商標を持つLaTeX[7]がある。日本における理工系学会の多くは、投稿論文をLaTeXで書くことを推奨するほどスタンダードな数式記述の道具となっている。LaTeXでは意味記述を行うことは出来ないが、前述のXMLやMathMLより、記述が簡単であるため、まずLaTeXを用いて意味記述が何であるかを明らかにする。

LaTeXで分数y/xと、微分dy/dxを記述する際、いずれもLaTeXにおける分数コマンドを用いて表現する。記述方法が何であれ、表示された数式を読む者にとって、何ら支障はない。微分記号が分数コマンドで記述されていても、読者は記述方法を知る必要はない。以下が、LaTeXコマンドで記述された分数と微分記号である。

分数		微分	
\frac{y}{x}	$\frac{y}{x}$	\frac{dy}{dx}	$\frac{dy}{dx}$

微分を学習していない者、例えば中学生がdy/dxを見た場合、分数として理解するであろう。また微分・積分を理解している者は、dy/dxを微分として理解するが、これがyd/xdであれば分数として解釈する。これは無意識に、読み手が表示された記号dがdifferential（微

分）であることを理解しているため、意味を持たせて読むことができ、表示内容を微分と解釈しているからである。しかしながら、上記LaTeXの例では、分数であれ微分であれ、表示される文字が正しければ問題は無いため、LaTeXにおける分数コマンド命令を用いて記述している。従って、機械（コンピュータ）がLaTeXコマンドを解釈して、自動的に区別して判断することは出来ない。この区別を行うには、分数用の記述と微分用の記述方法を、分けて記述するしかない。上記数式記述では意味記述ができないLaTeXを用いたが、次節では意味記述ができるMathMLを例に取り、意味記述について述べる。MathMLでは、LaTeXとは異なり意味を記述するコマンド（タグ）が定義されている。

3.2 XML用記述言語

XMLに数式を埋め込むには、数式記述用言語MathMLを用いる。他に、ベンゼン環に代表される化学記号を書くにはChemical MLがあり、楽譜を書くためにはMusic MLが用意されている。前節ではLaTeXを用いて分数と微分を書く際に、意味の区別が出来ないため、分数と微分ともに分数コマンドで記述する例を示した。MathMLでは、数式記述定義は2種類あり、表示形式と意味形式がある。また数式の記述方法は3種類あり、表示形式を用いて書くもの、意味形式を用いて書くもの、及び表示形式と意味形式を混在させて書くことが認められている。前節のLaTeXによる記述は、MathML定義における表示形式による記述にあたる。従ってMathMLにおいて、表示形式を用いて書くもの、及び表示形式と意味形式を混在させて書く記述は、数式の意味記述が出来ないことになる。

前節の分数と微分の違いは、読み手によって理解されるもので、書かれている数式の前後関係で解釈されるものである。前後関係が示されていない場合、読み手にとり理解できないものがある。

| X |	| Y |	| Z |
絶対値	行列式	濃度

上記の数式は、絶対値、行列式、濃度であるが、いずれも縦棒2本の間にアルファベット大文字1字が書かれたもので、その意味を区別することは出来ない。人間がXMLで表示されたものを解解する場合、記述されているコマンドを読むのではなく、Webページに表示されたものを理釈する。これに対して、コンピュータが解釈する場合、XMLコマンドそのものである。これは、人間がホームページを読む場合、Web表示画面を眺めるが、コンピュータはHTMLタグ（XMLコマンド）を解釈する場合も同じである。絶対値、行列式、濃度の区別を、上記の表示だけでは区別できないが、マークアップ言語で区別され記述されているのであれば、意味解釈が可能となる。

次の例は、高校数学1年で学ぶ2次方程式の解の公式を、MathMLで記述した例である。
6 前述の通りMathMLの定義には表示形式と意味形式の2種類があり、その記述方法は、表示形式、意味形式、および表示形式と意味形式の混在の3種類がある。以下では、その区別を明確にするため、表示形式のみによる記述と、意味形式のみによる記述を併記する。

$$ax^2 + bx + c = 0 \qquad x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

意味形式と表示形式の違いは、MathMLの詳細について定義を知らないと理解しがたいが、その定義を知らない者でも、上記右側の公式で使用されている演算記号（±，－，√）や、アルファベット記号（a, b, c, x）を元にひも解くと、記述パターンが理解できる。意味形式は、ポーランド記法を基にした記述であるのに対して、表示形式は解の公式における記号が出てくる順番に、その位置関係を記述していることが分かる。

意味形式

```
<apply>
  <eq/>
  <ci>x</ci>
  <apply>
    <divide/>
    <apply>
      <mo>&PlusMinus;</mo>
      <apply>
        <minus/>
        <ci>b</ci>
      </apply>
      <apply>
        <root/>
        <apply>
          <minus/>
          <apply>
            <power/>
            <ci>b</ci>
            <cn>2</cn>
          </apply>
          <apply>
            <times/>
            <cn>4</cn>
            <ci>a</ci>
            <ci>c</ci>
          </apply>
        </apply>
          <cn>2</cn>
      </apply>
    </apply>
    <apply>
      <times/>
      <cn>2</cn>
      <ci>a</ci>
    </apply>
  </apply>
</apply>
```

Content MathML

表示形式

```
<mrow>
  <mi>x</mi>
  <mo>=</mo>
  <mfrac>
    <mrow>
      <mrow>
        <mo>-</mo>
        <mi>b</mi>
      </mrow>
      <mo>&PlusMinus;</mo>
      <msqrt>
        <mrow>
          <msup>
            <mi>b</mi>
            <mn>2</mn>
          </msup>
          <mo>-</mo>
          <mrow>
            <mn>4</mn>
            <mo>&InvisibleTimes;</mo>
            <mi>a</mi>
            <mo>&InvisibleTimes;</mo>
            <mi>c</mi>
          </mrow>
        </mrow>
      </msqrt>
    </mrow>
    <mrow>
      <mn>2</mn>
      <mo>&InvisibleTimes;</mo>
      <mi>a</mi>
    </mrow>
  </mfrac>
</mrow>
```

Presentation MathML

7

わずか1行の数学の公式を、MathMLを用いて記述すると、意味形式および表示形式のいずれの場合でも、上記のように相当な行数になる。これらを人手により記述する場合、出来ないことは無いが、タイプミスや定義にそぐわない記述がなされて効率的ではない。上記例の記述は、何らかのアプリケーション・ソフトウェアにより、自動的に生成されることが望まれる。ただし、表示形式で記述を行う自動化はできても、意味形式による記述を完全に自動化することは不可能であるため、人手による校正のプロセスを設ける必要がある。

2節で提示したRDFには、MathMLの意味形式で記述された数式が示されている。W3Cにより公表されているRDF仕様書のコンテキストでは、Web資源（リソース）を記述するデータをどのように作成するか、何を用いて作成するか等は記載されていない。従って、2節で示したEBNFで定義されたRDFに準拠して記述しても、意味記述が出来る訳ではない。つまりRDFで書かれたWebページが、直ちにSemantic Webに成り得るものではない。

4．開発プログラム

意味が何であるかを、全てのWeb上のページについて定義することは困難である。ここで取り上げた数式は、意味のあるものと無いものの差異が、比較的理解しやすい記述になっているためMathMLを用いて前節で説明した。W3Cのホームページにも、RDFに埋め込まれたMathMLのサンプルが掲載されているが、やはり意味が何であるかを理解するには難しい例である。その違いを明らかにするため、行数の少ないMathMLによる意味のある記述と、意味の無い記述を対比させると次のようになる。

$$\int_a^b f(x)\mathrm{d}x$$

意味形式記述－1

```
<apply>
  <int/> <bvar><ci>x</ci></bvar>
  <interval><cn>a</cn><cn>b</cn></interval>
  <apply>
      <fn><ci>f</ci></fn><ci>x</ci>
  </apply>
</apply>
```

意味形式記述－2

```
<apply>
  <int/> <bvar><ci>x</ci></bvar>
  <lowlimit><ci>a</ci></lowlimit>
  <uplimit><ci>b</ci></uplimit>
  <apply>
      <fn><ci>f</ci></fn><ci>x</ci>
  </apply>
</apply>
```

表示形式記述

```
<mrow>
  <msubsup>
    <mo>&int;</mo> <mn>a</mn> <mn>b</mn>
  </msubsup>
  <mrow>
    <mi>f</mi> <mo>&ApplyFunction;</mo>
    <mrow>
      <mo>(</mo> <mi>x</mi> <mo>)</mo>
    </mrow>
    <mo>&InvisibleTimes;</mo>
    <mrow>
      <mo>&DifferentialD;</mo> <mi>x</mi>
    </mrow>
  </mrow>
</mrow>
```

上記例は、関数$f(x)$に対して区間［a, b］における積分を求めるもので、意味記述として２つの書き方を示してあるが、いずれもMathMLの意味記述定義に従って書かれたもので、区間［a, b］の関数$f(x)$の積分であることが理解できる。この２つの意味記述に対して、上記右側の表示形式による記述は、表示される記号が現れる順に列記したものである。意味記述がなされている場合、その内容をコンピュータが理解し、解釈された内容を別の形で扱うことが出来る。例えば、上記の積分の記述であれば、これをサードベンダーが提供する数式処理ソフトMathmatica等と連携し処理を行うことが可能である。あるいは統計処理ソフトと組み合せることにより、グラフなどの自動作成も可能となるであろう。また記述された内容の一部を変更することで、教材として練習問題を作成することも出来る。意味が記述されていることで、記述内容から派生したリソース（資源）を生成するような２次利用も可能となる。

Semantic Webを実現するには、容易なページ作成ツールが必要である。前述の表示形式でページ記述されたものもRDFになり得るが、意味形式で記述しないことにはSemantic Webにはならない。この問題を解消するために、Semantic Webを構築するための補助ツールを開発した。入力は理工系文書を対象とし、印刷された文書の認識はInfty[8]により自動的に行い、Inftyの出力はHTML、MathML、LaTex等が可能である。Inftyは手書き数式の認識も可能であるが、今回は印刷された文書あるいはPDF化された文書を入力とし、認識結果をLaTeXに変換し、さらにLaTeXを変換対象とし意味形式による出力機能を持たせた。動作環境はWindows XP（Vistaは未対応）、記述はC/C++で行った。本プログラムは、既に提案済の変換技術[9]を具現化したものである。変換アルゴリズムは、各数式命令に対する意味記述を１対１に対応させ、再帰的に解析を行う。印刷文書をスキャナで光学的に読み込むため、全ての入力文字から意味を抽出することは不可能である。例えば以下の例における縦棒の意味を、印刷文字からだけでは理解できない。

$\|X\|$	$\|Y\|$	$\|Z\|$
絶対値	行列式	濃度

上記例では、記号を機械的に取り込むだけでは、縦棒の意味が何であるかは理解不能であり、これらは人間により記述されている文書の意味を前後関係から理解するしかない。したがって、本プログラムは完全に自動化されたものではなく、半自動化システムである。上記例では、行列式の縦棒が絶対値の縦棒に解釈された場合は、自動変換後に修正が必要である。つまり縦棒は、絶対値・行列式・濃度、さらには集合で使われる要素の区切り記号$\{x|x\in R\}$であるが、上記の例では判別ができない。本プログラムで完全に自動化されない部分は、記号のみでは意味が導き出せない点で、自動生成されるXML（MathML）タグに、生成後に修正を必要とする。これはワープロ使用時に、同音異議語の選択が完全に自動化できないことと同じである。

Web上のホームページに表示される情報で、文字は何らかのコード体系によるキャラクタであり、画像はイメージであることが察せられるが、数式や化学式（ベンゼン環や構造式）は、XMLファイルに記述されたタグを解釈して表示している内容か、他のアプリケーションソフトで作成した画像を貼り付けたものかは、表示画面からでは判断できない。ファイル内部の記述が、意味を持ったタグで記述されていれば、Web上の表示のみならず、他のアプ

リケーション・ソフトウェアでも利用可能である。本プログラムは、意味記述されたタグを生成するため、生成コードの２次利用が可能である。例えば、数式ソフトMathematica用のコードに変換でき、２次利用できる。

XML用タグ・コードに変換するソフトウェアとして、LaTeX2HTML、TtM、MathType、TtH、MathML、Renderer等[6]が提案されているが、いずれも表示形式記述のコード生成しか行われていない。特にsoft4science社のMathML Rendererは、表示形式のみを対象にしていることが明記されている。本プログラムが自動生成する意味形式のコード生成機能を持った応用プログラムは、まだ提案されていないのが現状である。意味が記述されていることで、記述内容を自動的に判断し、他のソフトウェアに応用することや、記述内容から派生したリソース（資源）を生成する２次利用を可能とすることで利用範囲が広がり、本開発システムが半自動化システムであっても、RDFの一助となる意味は大きい。

5．まとめ

Resource Description Framework（RDF）とSemantic Webの関係は、冒頭でも述べたとおり、RDFはWeb資源の処理の自動化を容易に実現する道具であり、RDFを用いて記述することがSemantic Webを構築することを指している訳ではない。第１世代ページ記述言語HTMLは自由度が高い反面、利用されるタグに統一性が無いという問題から、第２世代ページ記述言語XMLでは構文（シンタックス）レベルで統一がなされ、より効率の良い自動化を図るためにRDFが提案された。この流れの中でSemantic Webを構築すべくオントロジーの概念を導入して、内容を説明する意味情報（メタデータ）を各文書に付加し、メタデータを記述する用語を定義する構造を構築していくものである。オントロジーを導入することで、検索対象となる文書が単なる単語の集まりとしてではなく、文書全体で大きな意味を持ったデータとして扱われ、各文書について統一的な付加情報をもたせることができる。意味を持った記述は、人工知能の構築や検索システムの基礎となる。

コンピュータが誕生し、これまで行われてきたデータの蓄積はData Base（DB）に代表される統合的なデータの管理であった。1990年代より爆発的に発達したインターネット環境は、コンピュータの専門家のみならず一般家庭までWeb検索を可能にし、Web上には膨大な知識の蓄積がなされている。Web上の情報を検索するには、統合的に管理されたDBの検索方法とは異なり、WWWにおけるページ間のリンク情報を使ったランキング計算を行いGoogle的検索がなされる。Google的検索が行われるようになり、ロングテール（注３）にも検索が可能となり、ロングテールに標準を合わせた需要[10]も掘り起こされた。さらに今後は、Web記述においてRDFを用いたSemantic Webの構築が進めば、オントロジーに基づく知識表現が構築されていく。今後は、強力で安全なネットワーク技術と膨大な並列に存在するコンピュータを基に、知識ベースが蓄積され、エキスパート的問題解決が可能となる。将来のSemantic Web構築のための基礎的技術として、RDFは重要な道具である。

参考文献

[1] HTML, http://www.w3.org/MarkUp/
[2] RDF, http://www.w3.org/RDF/
[3] World Wide Web Consortium(W3C), http://www.w3.org/

[4] XML, http://www.w3.org/XML/
[5] Semantic Web, http://www.w3.org/2001/sw/
[6] MathML, http://www.w3.org/Math/
[7] Leslie Lamport: "LaTeX, A Document Preparation System", Addison-Wesley, 1986.
[8] M.Suzuki, T.Kanahori, N.Ohtake and K.Yamaguchi: "An Integrated OCR Software for Mathematical Documents and its Output with Accessibility", Lecture Notes in Computer Science, No.3118, pp.648-655, Springer-Verlag, 2004.
[9] N.Ohtake and T.Kanahori: "A Conversion Tool for Mathematical Expressions in Web XML Files", Journal of Visual Impairment and Blindness, pp.713-719, November, 2007.
[10] Chris Anderson著、篠森ゆりこ訳、ロングテール、早川書房、2006.9.

注1．W3Cは、Web技術の標準化と推進を目的とした会員制の国際的な産学官協同コンソーシアムである。参加会員の合意に基づいた技術仕様やガイドラインの勧告としての策定を通じ、ベンダ中立でオープンな仕様に基づくマークアップ言語や通信プロトコルの開発と推進により、Webの相互運用性の向上とユニバーサルアクセスの実現に努めている。

注2．デファクト・スタンダード（de facto standard）が、競争の結果、市場で認知された**事実上の標準**に対し、デジュール・スタンダード（de jure standard）は、公的に組織された標準化機関により認証された標準を指す。ここでは、公的機関W3Cが公表されているドキュメントが、**認証された標準**にあたる。

注3．一般商店では、商品の20％が稼ぎ頭であり、残り80％は売れない商品として存在している。この売れない残りの商品80％部分がロングテールにあたるが、これまで余り売れなかった商品が、Web店舗での欠かせない収益源になった。このWeb上のネット売買の現象を、米Wired誌編集長Chris Andersonが指摘（2006.7）し、現在ではロングテール効果、ロングテール現象、ロングテール論などと呼ばれている。

（受理　平成19年９月４日）