

## RSsS: SemanticWeb メタデータ標準を活用した XML サマライゼーションの一試み

野村直之\*1 小林茂\*2

概要: Web サイトのサマリーを統一的な様式で記述するための標準 RSS が提案されて久しい。しかし、文書内部、文章内部に立ち入って細やかに抽出箇所を指定することを想定していないためか、コンテンツ本体と誤りなく同期させつつ手作業で作成させられる煩雑さ故か、余り普及していない。本稿では、コンテンツ主導の情報統合という今後の流れに沿って、XML 文書スタイル関係の属性をオブジェクト指向的に統一管理し、著者や編集者の意図を反映したサマリーやデータベース・レコードへの情報抽出の処理内容を SemanticWeb のメタデータ標準で統一的に記述する試みを提案する。

## RSsS: a XML Compound Document Summarization Control Framework using SemanticWeb Metadata Standards and Stylesheets

Naoyuki NOMURA\*1 Shigeru KOBAYASHI\*2

Abstract: We present a summarization control framework for XML Compound Document with XHTML and SVG. It takes advantages of RSS, RDF, and the stylesheets -- XSLT and CSS -- and lets the authors, editors and the readers arbitrarily control which parts of the original document structure to be extracted into the summary.

### 1 はじめに

伝統的なテキスト要約研究では、万人に認められるべき一定の正解の存在、もしくは優れた要約の存在を仮定していた([Sparck-Jones99])。しかし、本来、要約は、文書流通のプロセスにおいて、読者の興味や目的、編集者の価値判断、著者の意図等に基づいて制御されるべきものである([野村 2001a], [野村 99])。そのために必要な要件として、下記をあげることができる:

- (1) 複合文書のシンタックス、フォーマット自体が広く普及した標準に準拠していること、
- (2) 当該文書に対し抽出箇所を指定するアノテーションの素材(語彙とシンタックス)が広く普及した標準に準拠していること、
- (3) 当該文書へのトップダウンの要約方針を構造的に記述・反映でき、また本文中の詳細部分にもイン・ラインで容易に記述できること、
- (4) 可読性が高く、できれば人手による追加・削除・修正・流用等の編集作業も容易であること、
- (5) 抽出箇所(重要箇所)の指定方法が他の標準的な文書修飾の方法と矛盾なく共存し、できれば  
高度に整合性、一貫性を保っていること、

\*1 法政大学  
Hosei University

(6) 抽出指定自体をメタデータとして流通させたり、他の類似の構造をした複合文書の要約指定に流用可能であること、

\*2 株式会社日本ユニシス  
Nihon Unisys

(7) 上流から著者、編集者、そして最下流の読者による指定へと上書き的に優先されるなど適用優先順位に混乱のない運用の仕組みを確立

可能であること、

- (8) 既存の標準的な Web クライアント環境で要約結果を閲覧、確認し、簡単な修正が可能、
- (9) 複数の抽出結果を一定の仕組みで配信したり、スプレッドシートやデータベースに取り込めるなど、高度な再利用性の仕組みを備え、他の Web 利用システム、アプリケーション(例えば電子市場のサマリーレポート生成[野村 2002c])と容易に接続できること。

以上の特性を備えるべく、既存の Web サイト・サマリー記述言語 RSS([RSS2001])を活用し、RDF、XSLT、CSS の各標準を利用した記述方法を考案した。以下では、まず第 2 節で基本素材 RSS、RDF、XSLT、CSS について若干の解説と共に吟味を行い、第 3 節で本提案の方式 RSsS による要約指定、要約の手順を記す。第 4 節では、実際の入力文書の例をとって要約指定、要約の流れ、要約結果、スプレッドシートでの再利用性を確認する。ここでは、さらなる前処理として要約指定自体を、その XHTML+SVG 文書の構造的特徴に基づいて XSLT で付与し、人手による手間の軽減ならびにミス防止をはかっている。最後に RSsS の今後の課題とともに、SemanticWeb やドキュメント流通の今後における位置付けについて展望する。

## 2 RSS (RDF Site Summary) 1.0 について

Web サイトのサマリーを統一的な様式で記述するための標準案として、RSS が提案されている [RSS2001]。RSS は、元々、いわゆる“Push Technology”と呼ばれた、インターネット上の多数向け情報配信の仕組みに根がある。Netscape 社が“My Netscape”と呼んだポータル・サイト上に、利用者が選択する「チャンネル」を提供するための記述言語として、1999 年に公開したのが、RSS ver. 0.9 である。その後、[野村・小林 2003]に図解されたように、主に、リソース間の相対的定義を担う SemanticWeb 基盤規格 RDF を採用するか否かで考え方が分裂し、異なるバージョンが乱立しているような状況にある。本稿で提案する恣意的サマリー記述の仕組み RSsS は、SemanticWeb の技術標準を可能な限り採用し、モジュール化、部品化しやすく、再利用性の高い“Controlled Summary”記述・流通のフレームワークを提案するため、RDF 準拠の“RDF Site Summary (RSS) 1.0 [RSS2001]”に準拠する。2つのリソースを関係付ける RDF は、「原文」と「サマリー」との関係を表すのに、原理的に親和性が高い、と考えたからである。

RSS 1.0 は、記述対象のサイトを<channel>要素の中で指定し、そのタイトル、リンク情報、そしてサマリーの任意文字列をブラウザに表示させることを想定した<description>要素を含む<item>要素を兄弟要素に指定する、というシンプルな構造をもつ (図 1)。

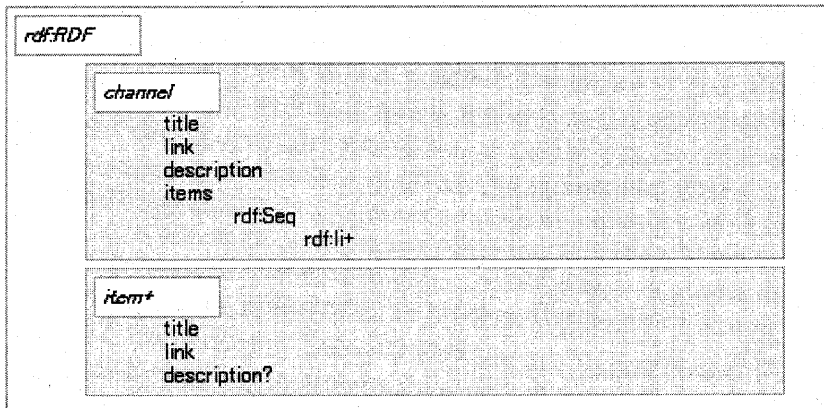


図 1 RSS 1.0 の基本構造 [神崎 2003] 神崎正英:「RSS -- サイト情報の要約と公開」より引用

RSS1.0 の特徴の 1 つはモジュール性である。図 2 に、SemanticWeb 最下層の基盤規格 RDF と、汎用メタデータ・セマンティクス分類項目の DublinCore (接頭辞 dc)、そして、部分コンテンツの更新サイクルなどを表現するシンジケート・モジュール(接頭辞 sy)とを名前空間で導入した例を示す。

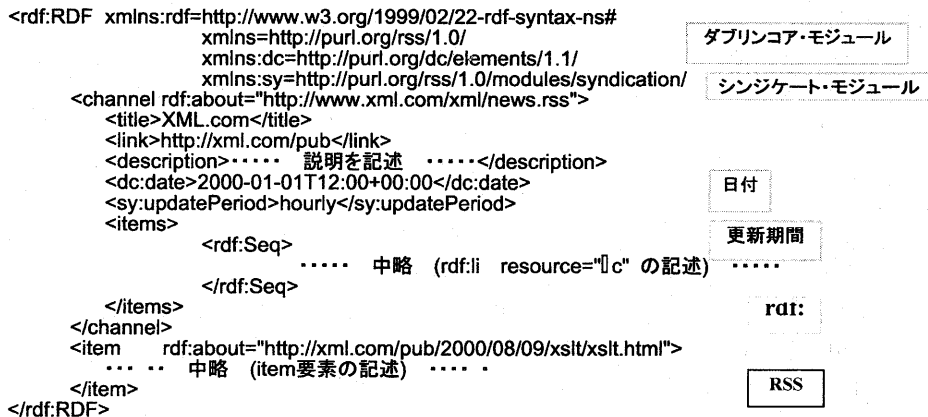


図2 他モジュールを利用した RSS1.0 の構造 [野村・小林 2003]

RSS1.0 の要素、属性、構文の定義の詳細は規格書[RSS2001]に譲るとして、我々は重要な特徴、制約として、下記 2 点に着目した：

- (1) サイトや文書に対してあくまで『外付け』の定義となっている；
- (2) <description>要素へのサマリー格納は手作業； 自動でサマリーを抽出したり、原文とサマリー間で同期管理する仕組みは内蔵していない；

つまり、RSS1.0 と図 2 のような若干の他言語モジュール活用だけでは、文書内部、文章内部に立ち入って細やかに抽出箇所を指定することは出来ない。また、コンテンツ本体と誤りなく同期させつつ手作業で作成させられる煩雑さを甘受せざるを得ない。我々は、この限界、機能の貧弱さこそが現状の RSS の普及を阻んでいる本質的原因の 1 つだと考えた。主たるコンテンツの作成、運用管理だけで手一杯のサイト管理者やライター達は、サイト更新のたびに、自サイトの“What’s New!”とは別に、慎重な手作業で作文して RSS 用の description 要素内容を埋めている時間的余裕は無いのではなかろうか。あるいはもし、サイト更新の内容に連動して description 要素内容を更新しないのだとすると、RSS には、「ここはこんなサイト!」という少量の、間接的で、新鮮さの乏しい記述しか反映されないことになる。これではサイトへの誘導効果は最小限にとどまる。よって、RSS 作成のインセンティブが不十分になる。これらの要因が複合して、Web にとって有用な、歴史ある規格でありながら普及していない現状に至っているのではなかろうか。

我々はこれらの欠点を克服するため、RSS の上位の言語や運用手法を編み出し、標準化を模索することにした。第一に、原文書中からのサマリー抽出箇所や抽出方法を、例えば単語単位、文字単位で木目細かく指定でき、制御可能とする仕組みを RSS に対する上位層として付加する。これにより、著者や編集者、再編集者、読者による恣意的な情報抽出&再構成、すなわちサマリー自動生成を実現する。第二に、逆に細かく指定することを常に強制された結果煩雑過ぎて使えなくなることはないよう、XHTML、文書の一定の標準的、規範的な構成パターンに対してサマリー抽出箇所指定を自動的に生成する仕組みを最初から提供する。原文書内部のメタデータ記述は、[川崎 2003]に準じてパターン認識する。「重要箇所」という修飾情報を原文書に付随させる仕組みであるから、構造変換のスタイルシート XSLT [XSLT]を採用するのが最適。この他、Web 上の応用目的で幅広く使える恣意的要約のための必要要件として、第 1 節の 9 条件を満たせるよう設計した。

### 3 RSsS による要約指定、要約の手順

本稿で提案する RSsS は、最終的に複数サイトからのサマリーを指定した条件(時間周期等)で自動抽出することを目指す。しかし、複数の XHTML-SVG の規範的な文書の構造特徴を抽出しながらマージするエンジンの標準については今後期待するとして、本稿の実験では、何らかの方法で既に1つにマージされた、XHTML-SVG の規範的な文書を初期入力としている。そして、名前空間で識別され、「島」形式で複合([野村 2001b])された XHTML、SVG 要素から、まずは前処理として、規範的な構造特徴を手がかりに、CLASS 属性に RDF 記述を埋め込んだ形の抽出指定を埋め込む。

一般には、常に更新される複数文書、サイトの XML 文書を入力として、

Step 1. XHTML, SVG の規範的な構成パターンを登録。

Step 2. これらのパターンを認識、識別する分類ルール等を作成、実行。

Step 3. 分類結果に基づいて、その構成パターンによる XHTML, SVG ソースからの書誌情報類を抽出、(一部は処理日時等から新たに生成)

Step 4. 抽出した書誌情報類 class 属性に RDF を記述する等で SemanticWeb 形式に変換

Step 5~. 汎用的な SemanticWeb 形式書誌情報をもつ文書から、

RSS やスプレッドシート、DB ファイル等を自動生成

上記手順の中では、複数ページをいつ、どのようにマージするかについては触れていない。マージのタイミングは大別すると、要約の前で実行するか、あるいは後で実行するか、となる。一般に、個々のページ、サイトごとに、文書構造の特徴は異なることから、それらごとに構成パターンを高精度に同定(Step2)する都合上、要約後のマージの方が有利である。さもなければ、Step 1 で用意されるべき構成パターンの数が組み合わせ爆発を引き起こし兼ねない。ただし、複数ページの内容、構造に深く依拠し、全体で一貫した要約方針を実現するには、マージ後に要約する必要がある。このため、部分ごとの要約率(抽出率)を制御する処理は、マージ後の内容全体に即して、即ち例えば類似検索技術における類似度判定([野村 2001a], [野村 99]等)を実行して、構成ページ、各部ごとの要約率を割り当てる。一方、XHTML, SVG 言語が規定する文書構造の特徴に即してそこから抽出対象を決めるメタデータを割り振る処理は、マージ前の個別文書、ページ、サイトの特徴で決めてしまう、という折衷案が有望と考えられる。

特定の複数サイト群を一定の周期で要約して報告する RSsS のアプリケーションを想定すると、マージの構成、順序が一定していて、それ専用で、即ちマージ後の全体構造に特化した構成パターンを登録(Step 1)する作業が割に合う、と考えられる。本稿では、このような例に該当する『今週の官庁』を次節で例題として採りあげる。また、XML 複合文書のマージの方式については、W3C 標準の汎用性の高い方式が現時点で確立されていないため、今後の XHTML2.0 モジュール化を中心とした標準化動向を見極めつつ将来の提案に委ねることとした。そこで、次節では、入力の時点で XML 複合文書のマージが既に完了した入力例を対象として、上記一連の Step と、その実行結果を追う。

### 4 入力例『今週の官庁』による要約指定、要約、結果の再利用の実際

更新頻度が高く、情報量が膨大で、スタイルや情報構造、情報提供の目的ポリシーが互いに全く異なり、公共性、重要性の高いサイトの典型例は中央省庁の提供する独立 Web サイト群である。図面がほとんどなく法律の条文の類が前面に出がちな財務省のサイトから、統計データ、図表が中心のサイト、最新情報の多くが議事録の書き起こしであるサイトなど実に様々である。利用者のプロフィール、利用目的によって期待される要約結果がまるで違う点においても説得力が高い。また、多段階の編集、勝手な要約を施されるべき対象として著作権問題がクリア済みのコンテン

ツが多くて、開発、試行、運用に好都合である。現状は、個別サイトの設計思想に合わせていちいち頭を切り換え、自分にとって関心の高い最新情報を人手で、目視で抽出する煩雑さが大きく、結局途中で諦める。すなわち大半の省庁サイトは滅多に訪れなくなっていて、重要な新着情報を見逃しているケースが多いのではなかろうか。このような弊害を無くす効果の立証のためにも、本例題は有効と考えられる。

今回、1つのXHTMLに6省(経済産業省、外務省、環境省、国土交通省、総務省、財務省)のトップページの内容、ならびに一部、リンクされたリソースをマージした。元のHTML文書中の要素をXHTMLに変換したものが大半である。加えて今回、サマリー生成実験のための画像ロゴは『今週の官庁』はSVGを追加して記述し、ブラウザ上で左上に表示されるようにした(図3)。

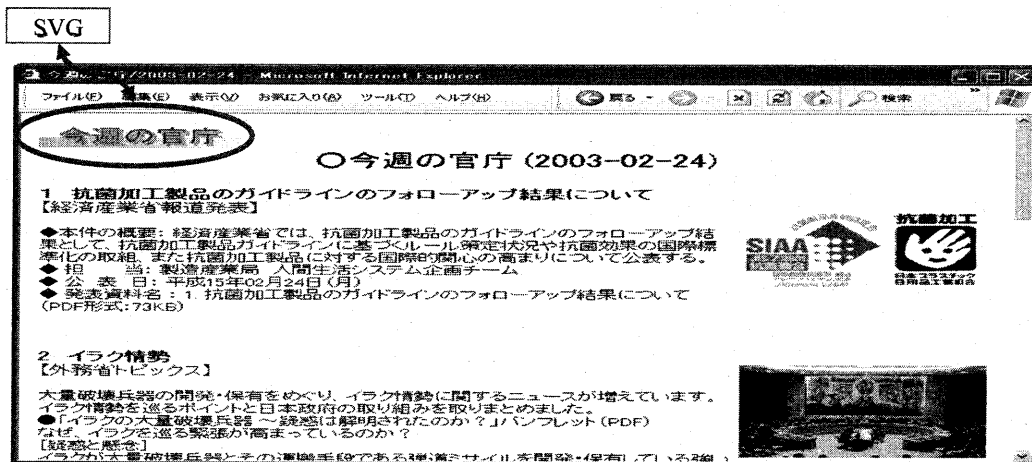


図3 入力例『今週の官庁』

図3のように文書構造がほぼ固定され、新鮮な内容が更新されていく入力XHTML文書に対して、規範的な構成パターンを登録する(Step 1)。この際、極力オーソドックスに、すなわち、広く浸透したHTMLの記述作法として、文書内のメタデータ記述と思しき構造要素を優先する。これは、意図せざる更新に遭うファイルのタイムスタンプや、HTMLエディタが自動付与するものではなく、著者が読者に伝えるべく本文内に記述されたと思しき書誌情報のことである。大半のブラウザで表示対象となる。非表示の<meta>要素は原則として対象外である<sup>1</sup>。ただしメタデータ内容が、本RSsSの枠組みが準拠するRDFで与えられていた場合(例:<rdf:Description about="http://www.w3.org/dc:title="抗菌加工製品のガイドライン">)その記述は原則信用し<sup>2</sup>、抽出対象としている。

例えば、<head>要素中の<title>要素には、多くの場合、文書主題が記述されるため、様々な規範的な構成パターンで共通に抽出対象としやすい。本稿の試作でも採用している。次に、<body>要素中のタグであるが、XHTMLのボキャブラリはDublinCoreのように意味内容として書誌情報を指定する要素を原則、欠いている。そこで、1文書に対して唯一出現し、文書レイアウトの点でも文書先頭に近いあたりでほぼ固定的に使われているヘディング・タグ<h1>、<h2>、...、<h6>の中には、見出しや作成者、責任者・権利者、その所属、公式な発行日付、などが格納されることが多い([川崎 2003])。ページ全体が幅ゼロの罫線(border="0")の表となっている場合を中心に、<table>要素

<sup>1</sup> <meta>要素の中に、DublinCoreに準拠して整然と書誌情報が記述されていて、その内容も概ね本文中の記述と矛盾なく整合しているようであれば(類似度判定が必要)、積極的に採用可能かもしれない。

<sup>2</sup> 本来は、XML Signature等で真正性を確認し、参照先も信頼がおけると確認した上で抽出対象とすべきであろう。

中の同等の要素<th>や、<caption>、<thead>、<td>の先頭、なども同様に文書書誌情報を格納する構造要素として登録対象の候補となる。文書先頭に近い術語定義の<dl>や汎用の箇条書き要素も候補である。Inline 要素の中で、意味的に書誌情報を表すとされる<address>要素が文書先頭近くで1カ所だけ用いられた際にも登録対象にすることを考慮すべきである。試作システムでは、この他に、スタイル属性による文書先頭近くのセンタリング指定 <div style="font-size:20pt; text-align:center;"> にパタンマッチさせ、最終的にメタデータ中の文書タイトルに変換させている。

精度が保証されないパターン認識にこれ以上頼れない、という状況で、RSsS の枠組みでは、XHTML、CSS の class 属性を利用する。汎用の block 要素<div>と、inline 要素<span>を新たに書き加え、RSsS のシステムが確実にメタデータとして拾うべき内容を、class 属性の属性値として記述する：

```
<div class="date">2003-03-05</div>
<div class="author">小林</div>
<div class="frequency" title="10"/>
```

複数の XHTML 文書構造テンプレートとの適合度を判定する Step 2 は、今回は実装していない。故に、その実現性、有用性は現時点では実証には至っていない。しかし、RSsS の枠組みは、一定のサイクル、流れに従って要約を流し続けるという目的実現のため、著者、編集者、読者が、ある程度 XML 複合文書の側に歩み寄り、必要ならば上記 class 属性中に確実に要約のためのメタデータを埋め込む、という思想に基づく。従って仮に、文書構造パターンを認識、識別する分類ルールを作成、分類実行に精度上の問題が生じれば、その負荷を軽減し、文書作成者、管理者、読者を RSsS 固有の class 属性の使用へと誘導する、という解決法が常に存在する。この、<div>または<span>によるメタデータ内容であれば、必ずしも文書先頭近くである必要はなく、任意の場所に埋め込むことができる。故に、今後いかに新規に奇抜な、創造的文書レイアウトを発明しても、いちいちそれを後追ひし、パターン認識するエンジン、ルール群を新規開発することなく、意図通りの抽出指定が可能となる。

次に、構成パターンによる XHTML、SVG ソースからの書誌情報類を抽出し、抽出した書誌情報類を class 属性に RDF を記述する等で可搬性の高い SemanticWeb 形式等に変換する (Step 3 & Step 4)。試作システムでは、第一に、可能な限り RDF に準拠した記述形式を採用することにした。なぜなら、抽出指定箇所に埋め込むメタデータをフルに書けば、原文書と抽出結果 (サマリー) という 2つのリソース間の関係を書き下すことになり、これはまさに RDF の機能、役割と重なるからである。こんな場合、独自の言語やボキャブラリ (タグ、要素、属性名) を作るべきではない。第二に、応用言語の層でも、類似の基本的機能を備えた言語モジュールが存在すれば、それらを可能な限り流用する、という方針により、RSS への変換を行うことにした。RSS には 3、4種のバージョンがあつて混乱しているが、第一のポイントにより RDF に準拠した RSS1.0 を採用する。これにより、RSsS 独自の RDF 埋め込み記述と整合性高く、且つ、変換コスト (計算量) が小さくなつことが期待されるからである。生成したメタデータを RSS に変換すれば、既に主要なブラウザが対応していて、表示、テストが可能となる。

数段階の変換、抽出には、スタイルシート (XSLT) を用いる。文書の部分構造にパターンマッチングを行い、原文書の特徴的構造、記述をメタデータに変換 (付加) するという静的な (宣言的な) 仕様を実装するのに最適の言語と考えられるからである。クライアント側の処理系も普及しており、追試、修正が容易という技術的な理由もある。この他に、似て非なる構造パターンの記述や変換仕様の少々の違いに合わせてスタイルシートを再利用していくことで、汎用のプログラミング言語よりも生産性が高く、利用層も広がる、と予想される。試作した RSsS では、前述の前処理よりさらに以前に、<head>要素内の meta タグに、DublinCore メタデータの記述 (例 : <meta

name="DC.Creator" content="経済産業省"/>)を埋め込んでいる。この前処理用スタイルシートが「文責情報」に相当するメタデータを保持していることになる。これにより、文書の公式メタデータがシステムに勝手に変更されぬよう、かといって本文テキストからの抽出による認識エラーを抑止することができる。同様に、各章、節ごとのタイトルや文責情報、更新情報のメタデータを事前に埋め込むこともできる。

このようにして Step3, 4 で汎用的な SemanticWeb 形式書誌情報化された抽出指定情報を、応用目的に応じて、RSS やスプレッドシートに変換する (Step 5)。図 4 はこうして生成されたメタデータ中心のサマリー情報構造である。

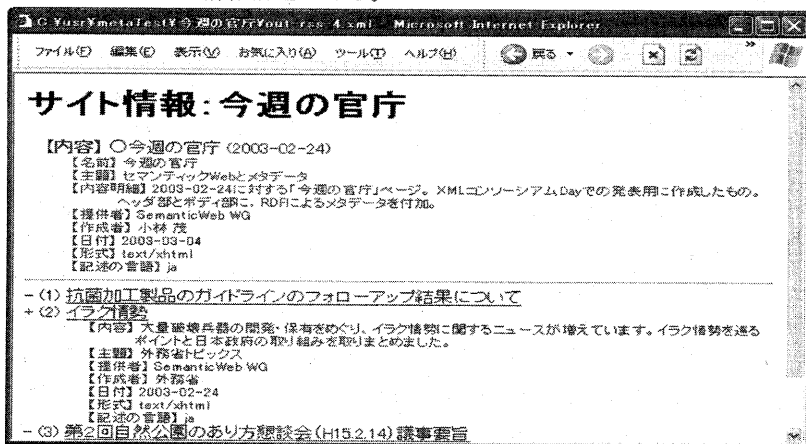


図4 メタデータ抽出結果にヘッダライン部分を開閉可能にするスタイルシートを適用

抽出された各項 (章、節) のヘッダラインサマリーは開閉可能。この意味で、複数種類の要約率を内包したサマリー構造となっている。

No.	タイトル	主眼	提供者	作成者	日付	内容
0	抗腐加工製品のガイドラインのフォローアップ結果について	環境省	SemanticWeb WG	環境省	2003/02/24	環境省では、抗腐加工製品のガイドラインのフォローアップ結果として、抗腐加工製品の取扱いに関する取組が進展していること、また抗腐加工製品の取扱いに関する取組の進捗状況についてお知らせいたします。
1	イラク情勢	外務省	SemanticWeb WG	外務省	2003/02/24	大量破壊兵器の開発・保有をめぐる、イラク情勢に関するニュースが増えています。イラク情勢を巡るニュースと日本政府の取組の概要を取りまとめました。
2	第2回自然公園のあり方懇談会(H15.2.14) 議事要旨	国土交通省	SemanticWeb WG	国土交通省	2003/02/24	平成15年2月14日、国土交通省の発注実施(スナール)による、第2回自然公園のあり方懇談会(H15.2.14)が、国土交通省の発注実施(スナール)により開催されました。
3	日本銀行 利率の現状	財務省	SemanticWeb WG	財務省	2003/02/24	日本銀行は、現在の利回り、貸付、預金、文庫などの現状についてお知らせいたします。
4	経済のあり方	財務省	SemanticWeb WG	財務省	2003/02/24	経済のあり方についてお知らせいたします。
5	経済のあり方	財務省	SemanticWeb WG	財務省	2003/02/24	経済のあり方についてお知らせいたします。
6	経済のあり方	財務省	SemanticWeb WG	財務省	2003/02/24	経済のあり方についてお知らせいたします。
7	経済のあり方	財務省	SemanticWeb WG	財務省	2003/02/24	経済のあり方についてお知らせいたします。
8	経済のあり方	財務省	SemanticWeb WG	財務省	2003/02/24	経済のあり方についてお知らせいたします。
9	経済のあり方	財務省	SemanticWeb WG	財務省	2003/02/24	経済のあり方についてお知らせいたします。
10	経済のあり方	財務省	SemanticWeb WG	財務省	2003/02/24	経済のあり方についてお知らせいたします。
11	経済のあり方	財務省	SemanticWeb WG	財務省	2003/02/24	経済のあり方についてお知らせいたします。
12	経済のあり方	財務省	SemanticWeb WG	財務省	2003/02/24	経済のあり方についてお知らせいたします。
13	経済のあり方	財務省	SemanticWeb WG	財務省	2003/02/24	経済のあり方についてお知らせいたします。
14	経済のあり方	財務省	SemanticWeb WG	財務省	2003/02/24	経済のあり方についてお知らせいたします。

```

r = startRow
c = startCol
Set root = doc.documentElement
Set itemList = doc.getElementsByTagName("rdf:Rss:item")
For i=0 To itemList.Length - 1
  Set item = itemList.item(i)
  Cells(r, c + 0).Value = i + 1
  Cells(r, c + 1).Value = item.selectSingleNode("rss:title").Text
  Cells(r, c + 2).Value = item.selectSingleNode("dc:subject").Text
  Cells(r, c + 3).Value = item.selectSingleNode("dc:publisher").Text
  Cells(r, c + 4).Value = item.selectSingleNode("dc:creator").Text
  Cells(r, c + 5).Value = item.selectSingleNode("dc:date").Text
  Cells(r, c + 6).Value = item.selectSingleNode("rss:description").Text
  r = r + 1
Next

```

図5 RSsのメタデータ抽出結果ファイルを読込んでExcel表に変換するVBAとその結果

図5は、RSsのメタデータ抽出結果ファイルを読込んでExcel表に変換するVBAとその結果である。このように、1度、原文書に即した一定のやり方で、更新のたびに使える標準メタデータを抽出できるようにしておけば、そのSemanticWeb標準に基づくDBレコード作成なども容易となる。

## 5 おわりに ～SemanticWeb, ドキュメント流通の展望とともに

以上、我々の提案する RSSs、すなわち SemanticWeb メタデータ標準を活用した XML サマライゼーション制御の一方式について概要、事例を示し、その表現能力、再利用性、機能性について議論した。また、文書本文内の書誌情報記述を抽出する工程やその下準備の一環として要約指定をとらえたことにより、文書ライフサイクル([野村 2003], [野村 2002b])の様々な局面で、様々な関わり方をする人間が各々の恣意性を要約に反映可能となった。全体として、標準への準拠、そして、似て非なる新言語は作らず、既存のものを組み合わせて新機能を実現、という設計方針を貫いた。

今後の課題の1つは、要素技術([野村 2002a])として、自然言語解析(抽出、ランキング)・生成技術を活用した文章要約エンジンを様々な局面で呼び出し、サマリーXML 文書やメタデータに反映できるように組み込み容易な部品化することである。モジュール性の良いエンジニアリングにより、プレーン・テキストから構造化文書、さらには構造化文書群を納めたデータベース、分散資源に至るまで、所望のサマライズ、メタデータ抽出を行う広範なスケーラビリティが実現可能と予想される。

指定した要素からのテキストや図表の要約率を1%~100%までスケーラブルに行うための属性と、対応エンジン(plug in)の拡張も望まれる。「オーバーライド」指定、そのポリシー指定を明示的にパラメータで与えられるようにする拡張も必要である。これにより下流絶対優先の原則以外のポリシーが導入可能となったり、相対的指定や、複数読者による多数決で一定の閾値を超えたら例外的に著者の絶対優先指定をオーバーライドする条件等が記述可能となる。汎用のアクセス制御記述言語と共存、併用させることにより、さらに機能性を高め、ドキュメント配信を高度化(例: 文書を改訂しつつ評価しながら、自然淘汰しながら配信)することも見込まれる。

## 参考文献

- [RSS2001] RSS-DEV WG: "RDF Site Summary (RSS)1.0", v.1.3.4, 2001 <http://web.resource.org/rss/1.0/spec>
- [Sparck-Jones99] Sparck-Jones, Karen: "Automatic Summarizing: Factors and Directions," in "Advances in Automatic Text Summarization", pp.6-7 'Evaluations,' MIT Press, 1999
- [川崎 2003] 川崎洋治, 野村直之, 中川尚:「文書構造情報の抽出とメタデータ化」, 情処学会研究報告 DD-38, 2003. 3
- [神崎 2003] 神崎正英:「RSS -- サイト情報の要約と公開」, 2003. 04 版, <http://www.kanzaki.com/docs/sw/rss.html>
- [野村 99] 野村直之:「ConceptBase の言語処理と新しいソリューション」, 情処学会研究報告 NL-129, 1999.
- [野村 2001a] 野村直之:「ナレッジマネジメントツールの配備、実践と次世代技術」 AI 学会誌 2001. 1 pp.33-41
- [野村 2001b] 野村直之, 川口浩司, 他 8 名:「XML 複合文書の実現方式に関する一考察 ～XHTML, SVG のポキャプタリ活用によるアプローチの有効性について」, 情処学会研究報告 DD-27, 2001. 3
- [野村 2002a] 野村直之, 中挟知延子, Key-Sun Choi :「第 1 回国際ワークショップ"NLP and XML"の概要とマルチモーダル・デジタル・ドキュメントの ISO 標準について」, 情処学会研究報告 DD-32, 2002. 3
- [野村 2002b] 野村直之:『ナレッジマネジメントと XML』, <http://xml.fujitsu.com/jp/talk/>, 2002. 7
- [野村 2002c] 野村直之, 赤井寧子:『電子注文票記述言語 eDealML ～グラフ理論応用により売買注文の 4W3H 取引制約を自動解決するための XML 言語』 ビジネスモデル学会・2002 秋季シンポジウム, 2002. 10
- [野村 2003] 野村直之:「サービス業の基幹的無形資産たるマニュアルの創造・保守・運用に関する一考察」, ビジネスモデル学会・第 2 回ナレッジマネジメント研究会, 2003. 2
- [野村・小林 2003] 野村直之, 小林茂:「セマンティック Web の動向とメタデータ」, 第 4 回 XML コンソーシアム Day 発表資料, 2003 年 3 月; <http://www.xmlconsortium.org/member/public/seminar/day04/prog1/20030304-04-2.pdf>