

2003年度第2回XMLコンソーシアムWeek

セマンティックWebとメタデータ

XHTML, SVGにRDFを埋め込んで
Webサマリー生成コントロール
(RSsS)

2003-05-27

XMLコンソーシアム

XMLテクノロジー部会 セマンティックWeb WG

日本ユニシス株式会社 小林 茂

主要内容

- **メタデータ**
 - 記述言語 RDF
 - ダブリンコア
- **RSS (RDF サイトサマリ)**
 - RSS 1.0 の紹介
- **メタデータの指定方法**
 - XHTML, SVG, NewsML
- **RSSの自動生成・表示**
 - XHTML, SVG, NewsML からRSS生成
 - RSSの表示
- **RSsSについて**
 - RSsSの表示, Webサービスに利用, PDF化

メタデータとは

- コンテンツに対するデータ
 - **HTML**では, titleタグ, metaタグ等で付加する
 - Webページに対するメタデータを与える
 - **NewsML**は, メタデータが豊富。Id, ……等
- それぞれの言語によって異なるタグが使われている
- **RDF**はメタデータを付加するための統一的な手段を提供する。W3Cで規定

メタデータをどこに指定するか

- **HTML, NewsML**では, そのドキュメントの中で指定される
- **RSS(RDF サイトサマリ)**は, ドキュメントの外から与える
- XML文書の中では, 応用の言語体系の中にあっても良いし, 名前空間を用いて他で定義するモジュールを利用することもできる
- **RDFをHTML(XHTML)**でも指定可能である

メタデータの項目

- **ダブリンコア**

- 15種類のメタデータとその意味を定義する
- 主に図書に関するメタ情報を起源としている
 - Subject, ……等
- この用語を利用すると, タグ名でその意味が明確になる
- 通常, dc:Subject と記述する

- 不足する項目は応用領域で定義する。なるべくdcと同じ意味のものを新たに定義しないこと

セマンティックWebでの メタデータ的位置付け

- セマンティックWebの基盤技術
 - **メタデータ**
 - **オントロジー**
 - **エージェント**
- Web上のコンテンツそのものだけでなく、メタデータに重点を置くことがセマンティックWebの特徴である
- そもそも、現在のWebページでは、メタデータが少なすぎる
 - 付けにくい
 - 現状では付けても利用価値が低い
 - 面倒である
 - 付け方が難しい

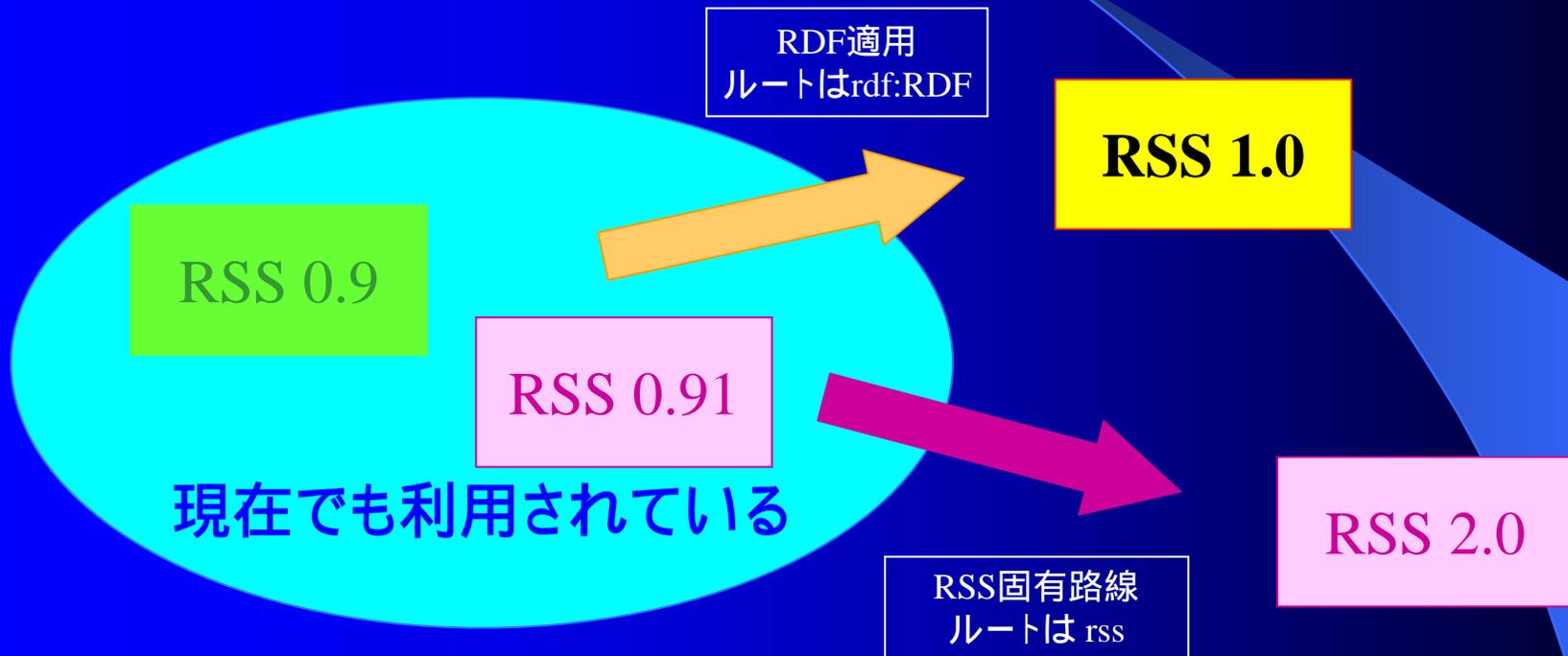
セマンティックWeb化への第一歩

- HTMLからXML化(XHTML化)する。
- XHTMLの中に, title , meta , を設定する
- idの活用,classの活用
 - class属性の利用:スタイル,メタデータ,データの振り分けに便利。表示不要の場合には, CSS(display:none)にて指定する
- XSLTを用意し,RSS化を図る
- RSSをサイトで公開する

Webページにメタデータを付ける

- Webページにメタデータを付ける方法の1つに**RSS**(**RDF** サイトサマリ, リッチサイトサマリ)がある
- **RSS**を利用して, サイトのインデックスを提示できる

RSS進化の流れ



RDF Site Summary(RSS)1.0

- RDF 1.0 とは,
 - RDF Site Summary (RSS) is a lightweight multipurpose extensible metadata description and syndication format.
 - RSS is an XML application, conforms to the W3C's RDF Specification and is extensible via XML-namespace and/or RDF based modularization.
- 作者
 - RSS-DEV Working Group
- 仕様書
 - <http://web.resource.org/rss/1.0/spec>

RSS1.0の構造

```
<rdf:RDF xmlns:rdf=http://www.w3.org/1999/02/22-rdf-syntax-ns#
  xmlns=http://purl.org/rss/1.0/ >
  <channel rdf:about="http://www.xml.com/xml/news.rss">
    <title>XML.com</title>
    <link>http://xml.com/pub</link>
    <description>..... 説明を記述 .....</description>
    <items>
      <rdf:Seq>
        <rdf:li resource="http://xml.com/pub/2000/08/09/xslt/xslt.html" />
        ..... 中略 (rdf:li resource="..." が続く) .....
      </rdf:Seq>
    </items>
  </channel>
  <item rdf:about="http://xml.com/pub/2000/08/09/xslt/xslt.html">
    <title>Processing Inclusions with XSLT</title>
    <link>http://xml.com/pub/2000/08/09/xslt/xslt.html</link>
    <description>..... 説明を記述 .....</description>
  </item>
  ... 中略 (item要素が続く。互換性のために15個以内が望ましい) ...
</rdf:RDF>
```

rdf:

RSS

他モジュールを利用したRSS1.0の構造

```
<rdf:RDF xmlns:rdf=http://www.w3.org/1999/02/22-rdf-syntax-ns#
  xmlns=http://purl.org/rss/1.0/
  xmlns:dc=http://purl.org/dc/elements/1.1/
  xmlns:sy=http://purl.org/rss/1.0/modules/syndication/ >
  <channel rdf:about="http://www.xml.com/xml/news.rss">
    <title>XML.com</title>
    <link>http://xml.com/pub</link>
    <description>..... 説明を記述 .....</description>
    <dc:date>2000-01-01T12:00+00:00</dc:date>
    <sy:updatePeriod>hourly</sy:updatePeriod>
    <items>
      <rdf:Seq>
        ..... 中略 (rdf:li resource="..." の記述) .....
      </rdf:Seq>
    </items>
  </channel>
  <item rdf:about="http://xml.com/pub/2000/08/09/xslt/xslt.html">
    ..... 中略 (item要素の記述) .....
  </item>
</rdf:RDF>
```

ダブリンコア・モジュール

シンジケート・モジュール

日付

更新期間

rdf:

RSS

メタデータの作成

- XHTMLの中からRSSを自動生成する
- アイデア
 - XHTMLのメタデータを表すタグを利用する
 - `<title>タイトル</title>`
 - `<meta name="date" content="2003-03-05"/>`
 - 特定のタグをメタデータ記述として利用
 - tableタグの利用(caption, thead, th/td)
 - `<address>inlineタグが書ける</address>`
 - XHTMLのclass属性を利用する
 - `<div class="date">2003-03-05</div>`
 - `<div class="author">小林</div>`
 - `<div class="frequency" title="10"/>`

メタ情報を埋め込む場所

- ヘッダの中に埋め込む
 - ページ全体としてのメタ情報を指定
 - 検索ロボットが探しやすい(metaタグ)
- ボディの中に埋め込む
 - 必要な場所にメタデータを埋め込める
 - ページを分割した単位でメタ情報を指定可能

サンプルによるデモ

- 6省のページを集めた「今週の官庁」
- Webページを集めて切り貼りしたもの

SVG



- XHTML, イメージ, SVG
- 1つのXHTMLに6個の話題(6省分)

今週の官庁
経済産業省
外務省
環境省
国土交通省
総務省
財務省

XHTMLでの記述内容

```
<html>
  <head>
    <title>今週の官庁/2003-02-24</title>
    <meta http-equiv="Content-Type" content="text/html; charset=Shift_JIS" />
  </head>
  <body background-color="yellow">
    <embed src="images/thisWeek-title.svg" width="450" height="35"/>
    <div style="font-size:20pt; text-align:center;">
      今週の官庁 (2003-02-24)
    </div>
```

SVG

ヘッダ部で meta タグを利用

```
<html>
```

```
<head>
```

```
<title>今週の官庁/2003-02-24</title>
```

```
<meta http-equiv="Content-Type" content="text/html; charset=Shift_JIS" />
```

```
<meta name="DC.Title" content="今週の官庁"/>
```

```
<meta name="DC.Subject" content="セマンティックWebとメタデータ"/>
```

```
<meta name="DC.Description" content="2003-02-24に対する「今週の官庁」ページ。XMLコンソーシアムDa
```

```
<meta name="DC.Publisher" content="SemanticWeb WG"/>
```

```
<meta name="DC.Creator" content="小林 茂"/>
```

```
<meta name="DC.Date" content="2003-03-04"/>
```

```
<meta name="DC.Format" content="text/xhtml"/>
```

```
<meta name="DC.Language" content="ja"/>
```

ボディ部分にもメタデータを

- XHTMLの文法(スキーマ)の中では class タグを利用する方法を採用
- 簡便な規則を設定

meta__item属性
のあるブロック
内でメタデータ
を指定

```
<div class="meta__item">  
  <a name="1" />   
  <span class="meta__item__xxxx">  
    </span>  
  <div class="meta__item__yyyyy">  
    </div>  
</div>
```

具体的なコード

```
<div class="meta__item">  
  <a name="1" />  
  <font size="4"><b>1 <span class="meta__item__Title">抗菌加工製品の・・・  
  </span></b></font><br />  
    
  <span class="meta__item__Creator">経済産業省</span>  
  <font size="4">[<span class="meta__item__Subject">経済産業省報道発表  
  </span>]</font><br /><br />  
  本件の概要: <span class="meta__item__Description">経済産業省では、・・・  
  .....  
</div>
```

RDFによるメタデータ (XHTML)

<rdf:RDF

```
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:dc="http://purl.org/dc/elements/1.1/" >
```

<rdf:Description

```
about="http://www.xmlconsortium.org/xml-day-2003-03-04#1"
dc:title="抗菌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"
dc:subject="経済産業省報道発表"
dc:description="経済産業省では、抗菌加工製品のガイドラインのフォロー..."
dc:publisher="SemanticWeb WG"
dc:creator="経済産業省"
dc:date="2003-02-24"
dc:format="text/xhtml"
dc:language="ja" >
```

</rdf:Description>

</rdf:RDF>

• ダブリンコアで意味するメタデータ

• 要素内容は無いので表示には無関係。
• head, body どちらにも入れられる。外部でもよい

メタデータからRSSを生成する

- スタイルシート(XSLT)で試行する
- 4種類の「今週の官庁」に対し検討
- オリジナルページ

```
<title>今週の官庁/2003-02-24</title>
```

```
<div style="font-size:20pt; text-align:center;"> 今週の官庁 (2003-02-24)</div>
```

- ヘッダ部のmetaタグからの抽出

```
<title>今週の官庁/2003-02-24</title>
```

```
<meta name="DC.Creator" content="経済産業省"/>
```

- ボディ部のclass属性からの抽出

```
<div class="meta__item">
```

```
<a name="6" />
```

```
<font size="4"><b>6 <span class="meta__item__Title">[図説] 財政の現状と……
```

- RDF記述からの抽出

```
<rdf:Description about="http://www.xmlconsortium.org/xml-day-2003-03-04#1"
```

```
dc:title="抗菌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"
```

metaタグからRSSを生成

```
<xsl:for-each select="head/meta">
  <xsl:for-each select="@name='dc:*' ">
    . . . . .
    <xsl:when test="name()='DC.Creator' ">
      <dc:creator><xsl:value-of select="@content"/></dc:creator>
    </xsl:when>
    . . . . .
  </xsl:for-each>
</xsl:for-each>
```

```
<dc:creator>経済産業省</dc:creator>
```

class属性(meta__item)から

```
<xsl:for-each select="//*[@class='meta__item']">
  <xsl:choose>
    <xsl:when test="a[@name]">
      <rdf:li rdf:resource="{ $default-uri }#{a/@name}"/>
    </xsl:when>
  </xsl:choose>
</xsl:for-each>
```

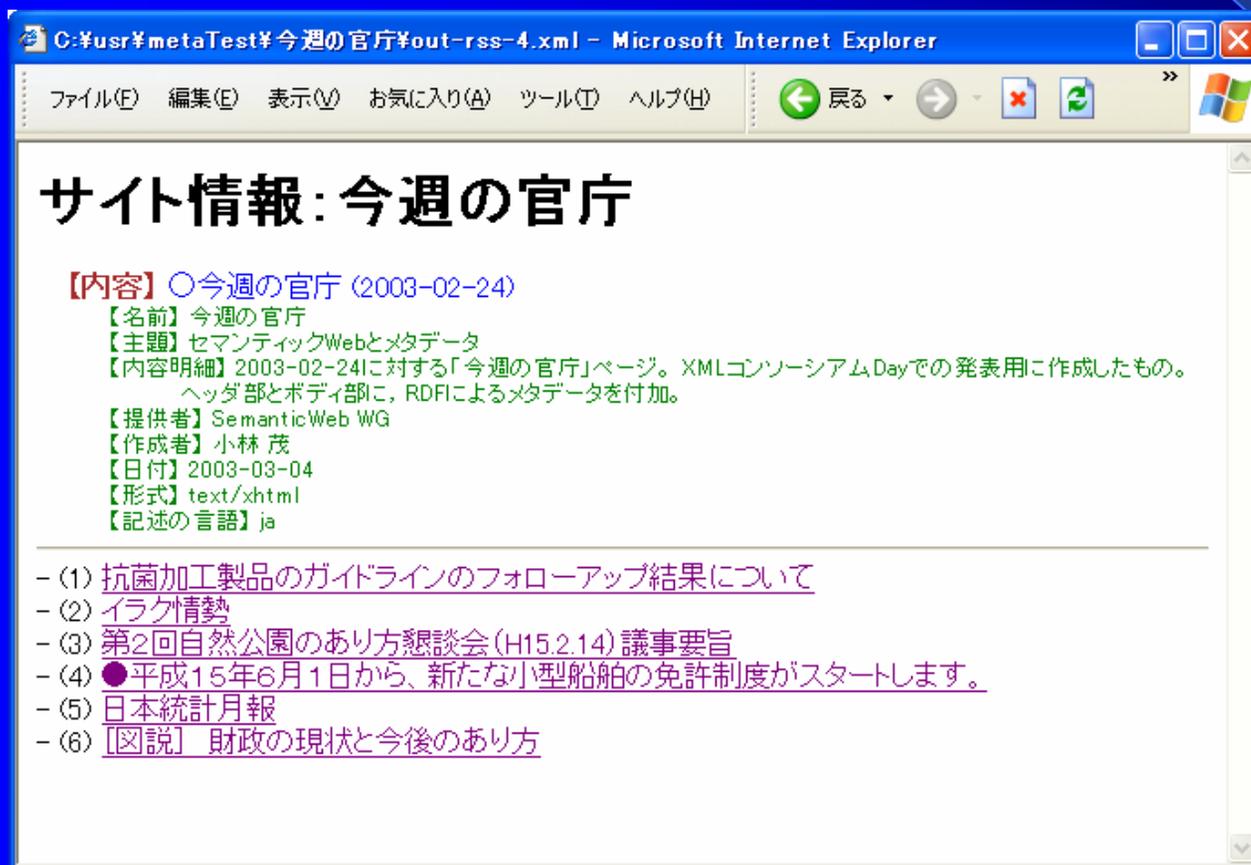
```
<rss:items>
  <rdf:Seq>
    .....
    <rdf:li rdf:resource="今週の官庁-2.xml.htm#1" />
    .....
  </rdf:Seq>
</rss:items>
```

RDFデータからRSSを生成

```
<xsl:for-each select="rdf:RDF">
  <xsl:for-each select="rdf:RDF/rdf:Description/@dc:*">
    . . . . .
    <xsl:when test="local-name()='title' ">
      <dc:title><xsl:value-of select="."/></dc:title>
    </xsl:when>
    . . . . .
  </xsl:for-each>
</xsl:for-each>
```

```
<dc:title> [図説] 財政の現状と . . . . . </dc:title>
```

RSSの表示



スタイルシート
「rss-simple-list.xsl」

RSSの表示(2)

C:\usr\metaTest\今週の官庁\out-rss-4.xml - Microsoft Internet Explorer

ファイル(F) 編集(E) 表示(V) お気に入り(A) ツール(T) ヘルプ(H)

戻る

サイト情報: 今週の官庁

【内容】○今週の官庁 (2003-02-24)

【名前】今週の官庁
【主題】セマンティックWebとメタデータ
【内容明細】2003-02-24に対する「今週の官庁」ページ。XMLコンソーシアムDayでの発表用に作成したものの、ヘッダ部とボディ部に、RDFIによるメタデータを付加。
【提供者】SemanticWeb WG
【作成者】小林 茂
【日付】2003-03-04
【形式】text/xhtml
【記述の言語】ja

- (1) 抗菌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

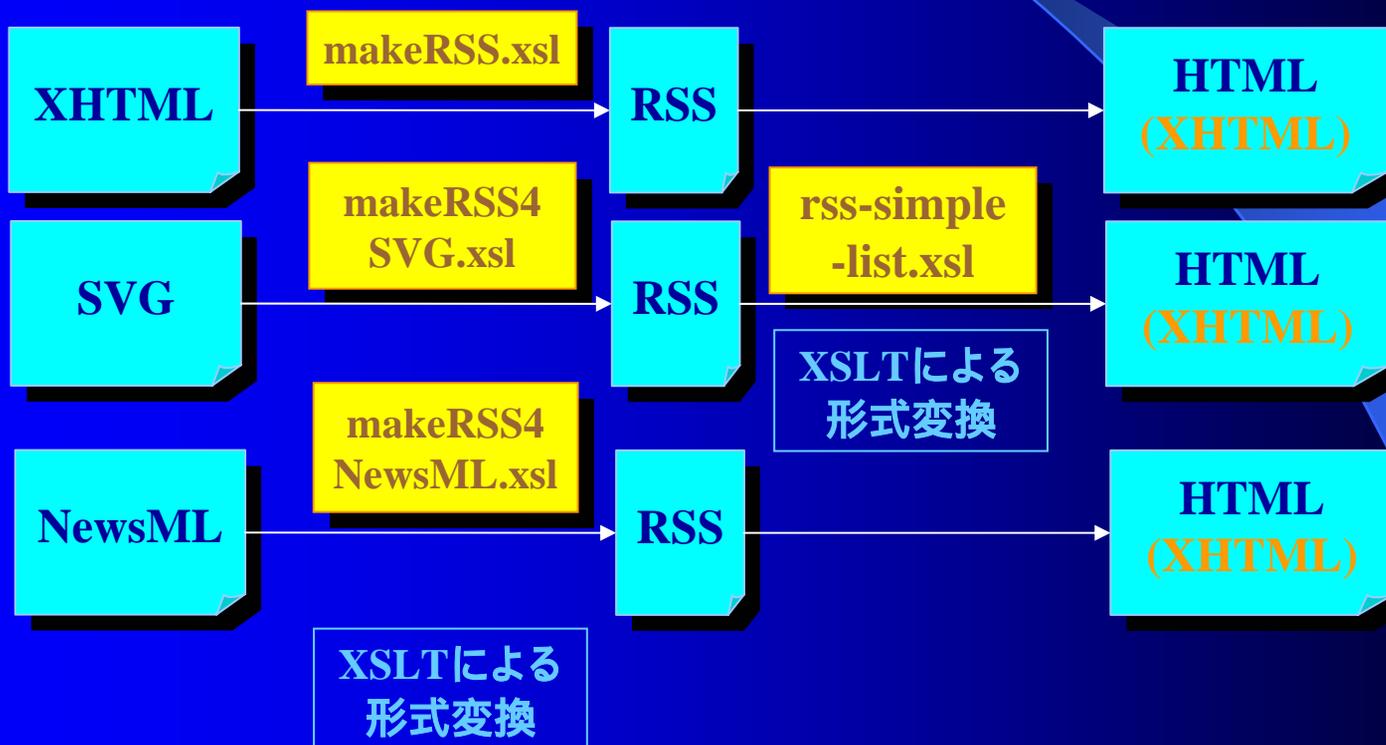
+ (2) イラク情勢

【内容】大量破壊兵器の開発・保有をめぐり、イラク情勢に関するニュースが増えています。イラク情勢を巡るポイントと日本政府の取り組みを取りまとめました。
【主題】外務省トピックス
【提供者】SemanticWeb WG
【作成者】外務省
【日付】2003-02-24
【形式】text/xhtml
【記述の言語】ja

- (3) 第2回自然公園のあり方懇談会(H15.2.14)議事要旨

2番目の項目を展開したところ

RSS生成と表示



他のXML言語でのメタデータ

- 固有のタグを規定している
 - NewsML
 - SVG
- 固有には保持しない。RDFを組み込む
 - SVG

SVG言語でのメタデータの指定

- title , desc , metadataタグが規定されている

```
<title>今週の官庁のタイトルロゴ</title>
<desc xmlns:sw="http://www.xmlconsortium.org/xml-tech/semanticWeb-Wg">
  <sw:meta name = "DC.Title"    content = "今週の官庁のタイトルロゴ"/>
  .....
</desc>
<metadata>
  <rdf:RDF .....>
    <rdf:Description
      about="http://www.xmlconsortium.org/xml-day-2003-03-04"
      dc:title="今週の官庁のタイトルロゴ"
      .....>
    </rdf:Description>
  </rdf:RDF>
</metadata>
```

SVGデータからRSSを生成する

```
<xsl:template match="/">
  <rdf:RDF>
    <rss:channel>
      <rss:title><xsl:value-of select="svg:svg/svg:title"/></rss:title>
      <rss:link><xsl:value-of select="$default-uri"/></rss:link>
      <rss:description>
        <xsl:choose>
          <xsl:when test="svg:svg/svg:desc/sw:meta/@name='DC.Description'">
            <xsl:value-of select="svg:svg/svg:desc/sw:meta/@content"/>
          </xsl:when>
        </xsl:choose>
      </rss:description>
      <rss:items>
        <xsl:apply-templates select="//svg:desc/sw:*"/>
      </rss:items>
    </rss:channel>
    <xsl:apply-templates select="//svg:desc[sw:*]" mode="item"/>
  </rdf:RDF>
</xsl:template>
```

<rdf:li rdf:resource .../>を生成

<rss:item 生成

NewsML

- **メタデータの塊り**
- **NewsML固有の情報が豊富**
- **利用するにはNewsML向けのツールが必要**
- **RSSでメタ情報記述するためのXSLTサンプル**

NewsMLでの例

一部のみ

```
<NewsML> .....
<NewsItem>
  <NewsComponent xml:lang="ja" Duid="NC0001">
    <NewsLines>
      <HeadLine>(SMIL 2.0) 仕様</HeadLine>
      <DateLine>20010328T000000+0900</DateLine>
    </NewsLines>
    <AdministrativeMetadata>
      <Creator>
        <Comment>http://www.w3c.org</Comment>
        <Party FormalName="W3C" />
      </Creator>
    </AdministrativeMetadata>
    <Metadata>
      <MetadataType FormalName="newsdoc-metadata" />
      <Property FormalName="DocumentDate"
Value="20010301T000000+0900" />
    </Metadata>
    .....
  </NewsComponent>
</NewsItem>
</NewsML>
```

NewsMLデータからRSSを生成する

```
<xsl:template match="/">
  <rdf:RDF>
    <rss:channel>
      <rss:title><xsl:value-of select="//NewsItem[1]//HeadLine[1]"/></rss:title>
      <rss:link><xsl:value-of select="$default-uri"/></rss:link>
      <rss:description>
        <xsl:value-of select="//NewsItem[1]//HeadLine[1]"/>
      </rss:description>
      <rss:items>
        <rdf:Seq>
          <xsl:apply-templates select="//NewsItem" mode="head"/>
        </rdf:Seq>
      </rss:items>
    </rss:channel>
    <xsl:apply-templates select="//NewsItem" mode="body"/>
  </rdf:RDF>
</xsl:template>
```

<rss:item を生成

複数サイトのサマリ RSsS

- 複数サイト(Sites)のサマリ言語RSsS(RDF Sites Summary)を考える
 - RSsSによって、異なる言語を持つドキュメントのメタデータを統一的に扱うことが可能である
 - XHTMLからRSSを自動生成する
 - SVGからRSSを生成する
 - NewsMLからRSSを生成する
 - RSsSによる応用の例
 - データベースへメタデータを格納する例、EXCEL表に設定
 - メタ情報をWebサービスで提供し、EXCELから利用する例
 - RSsSのドキュメント化の例、XSL-FOを編集しPDF作成
- これまでを準備段階として、セマンティックWebへの第一歩が始まる

RSsSの実体

```
<!DOCTYPE rsss:RSsS [  
<!ENTITY file1 SYSTEM "out-rss-4.xml">  
<!ENTITY file2 SYSTEM "out-rss-svg.xml">  
<!ENTITY file3 SYSTEM "out-rss-newsml.xml">  
>
```

```
<rsss:RSsS
```

```
  xmlns:rsss="http://www.xmlconsortium.org/xml-tech/semanticWeb-Wg"
```

```
>
```

```
  &file1;
```

```
  &file2;
```

```
  &file3;
```

```
</rsss:RSsS>
```

現在は、
RSSを単純に
実体参照している

RSsSに対するスタイルシート

rss-simple-list.xml
を利用する

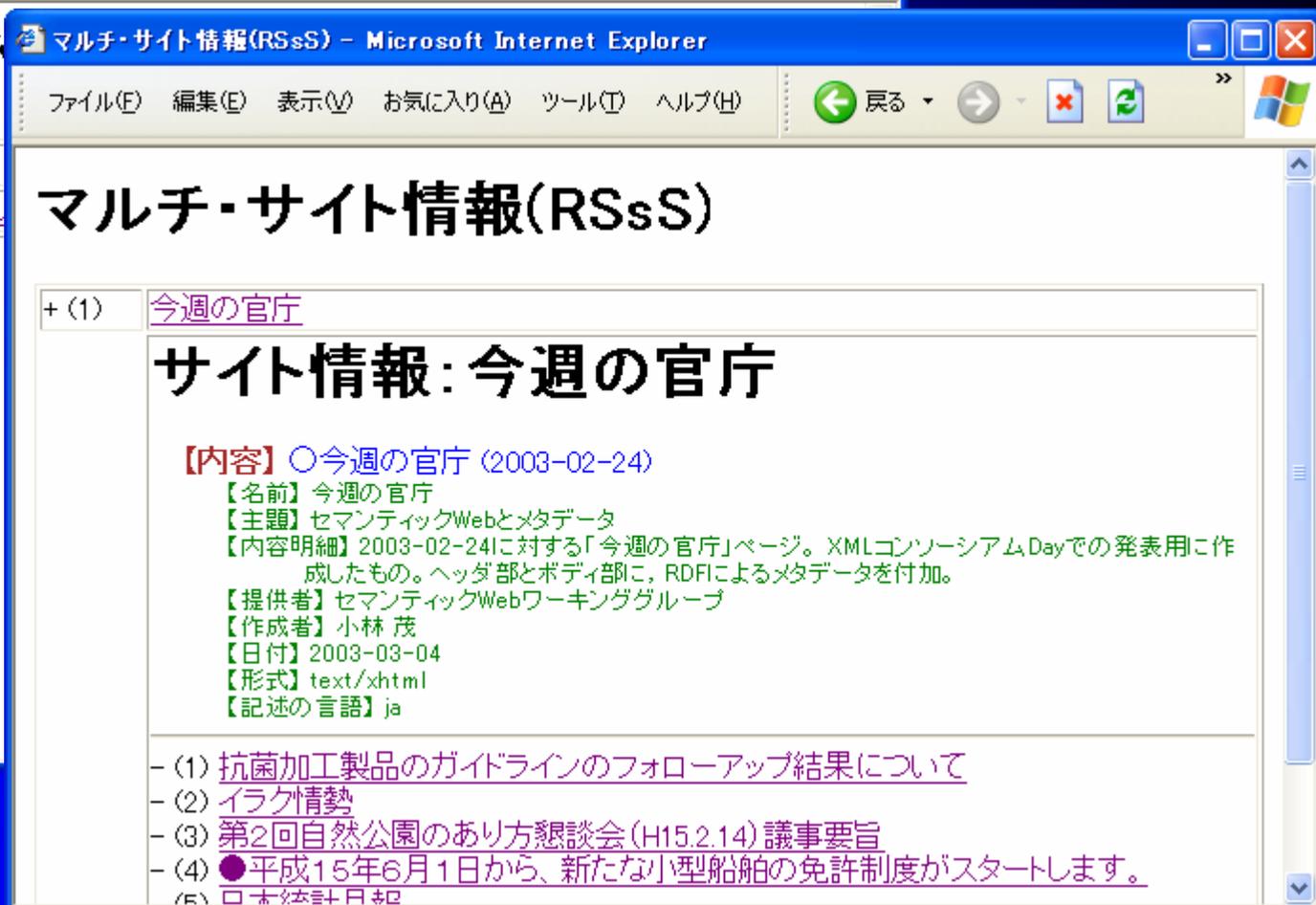
```
<xsl:import href="rss-simple-list.xml"/>

<xsl:template match="/">
  <html>
    .....
  <body>
    <h1>マルチ・サイト情報(RSsS)</h1>
    <xsl:apply-templates select="rss:RSsS"/>
  </body>
</html>
</xsl:template>
```

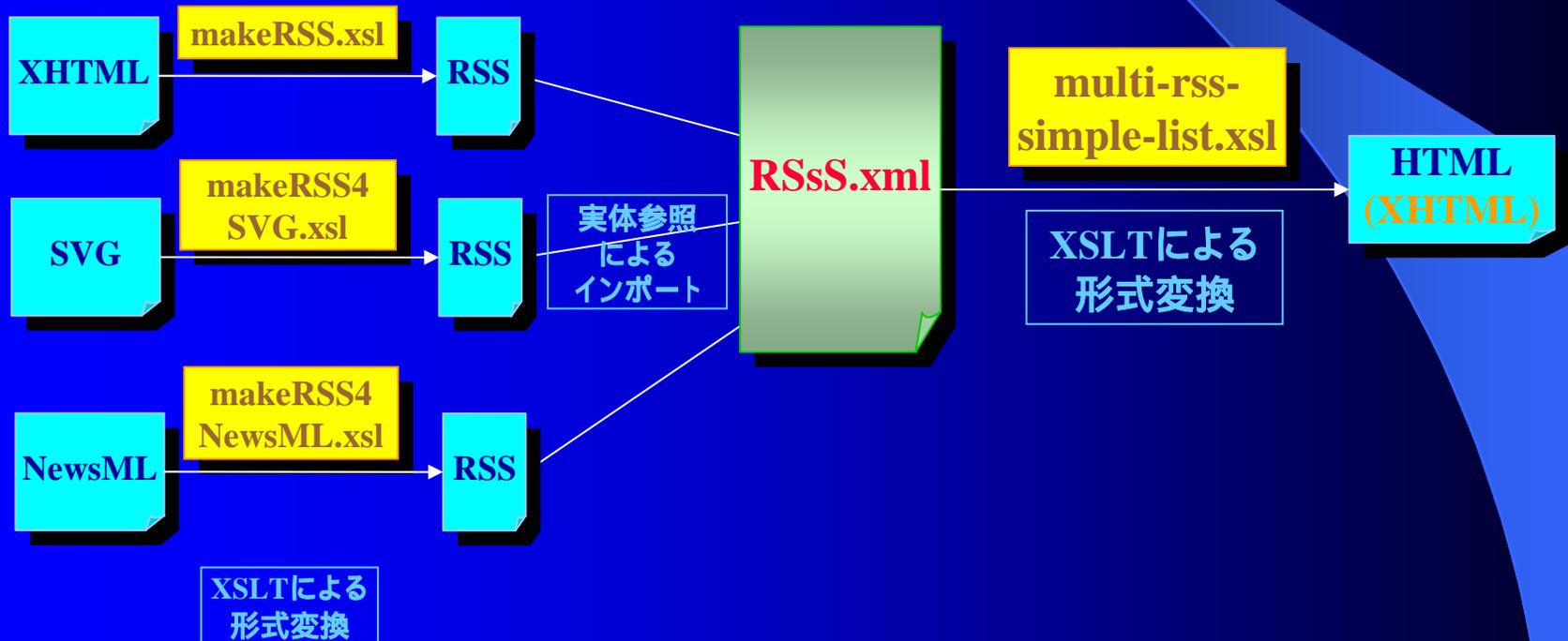
スタイルシート
「RSsS-simple-list.xml」

```
<xsl:template match="rss:RSsS">
  <table border="1">
    <xsl:for-each select="rdf:RDF">
      <tr>
        <td>
          <xsl:apply-templates select="."/>
        </td>
      </tr>
    </xsl:for-each>
  </table>
</xsl:template>
```

RSsSの表示例



RSsS生成と表示



RSsSからデータベースに

Microsoft Excel - SitesInfoByRSsS.xls

RSsS ファイルを読み込み、エクセル表に設定

読み 消去

ファイル名 D:\2003-xml\Week\demo\importer\RSsS.xml

番号	タイトル	主題	提供者	作成者	日付	内容
1	抗菌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	経済産業省報道発表	SemanticWeb WG	経済産業省	2003/2/24	経済産業省では、抗菌加工製品のガイドラインのフォローアップ結果として、抗菌加工製品ガイドラインに基づくルール変更状況や抗菌効果の国際標準化の取組、また抗菌加工製品に対する国際的関心の活発化について公表する。
2	イラク情勢	外務省トピックス	SemanticWeb WG	外務省	2003/2/24	大量破壊兵器の脅威、イラク情勢に関するニュースが増えています。イラク情勢を巡る米イ小と日本政府の取組みや取りまとめました。中央環境審議会自然環境部会自然公園のあり方検討小委員会「自然公園のあり方に関する中間答申(HI4.1.29)」において継続審議事項とされた課題について意見交換等を行うために開催した、第2回「自然公園のあり方懇話会」の概要についてお知らせします。
3	第2回自然公園のあり方懇話会(HI5.2.14)議事要旨	環境省新着情報	SemanticWeb WG	環境省	2003/2/24	平成15年6月1日から、新たな小型船舶の免許制度がスタートします。
4	●平成15年6月1日から、新たな小型船舶の免許制度がスタートします。	国土交通省トピックス	SemanticWeb WG	国土交通省	2003/2/24	新しい免許制度の概要は次のとおりです。
5	日本統計月報	総務省新着情報	SemanticWeb WG	総務省	2003/2/24	日本統計月報は、我が国の人口、経済、社会、文化などの動向を総合的に把握できるようにまとめたものです。
6	「国政」 財政の現状と	財務省トピックス	SemanticWeb WG	財務省	2003/2/24	平成14年度一般会計年度における歳入歳出決算は、そのうち「国政」の歳入歳出は、
7	今週の官庁のタイトルロゴ	セマンティックWebとメタデータ	SemanticWeb WG	小林 茂	2003/3/4	今週の官庁にて利用するSVGで記述したタイトルロゴ。XMLコンソーシアムDayでの発表用に作成したものです。
8	Multimedia Integration Language (SMIL 2.0)仕様	Multimedia Integration Language (SMIL 2.0)仕様	g	3c.org	28T00.00.00	記述するためのXMLベースの言語である。SMIL2.0は他の言語(HTML/SVGなど)で再利用されることが考えられている。製品ごとにモジュール化され提供される。
9	Synchronized Multimedia Integration Language (SMIL 2.0)仕様	Synchronized Multimedia Integration Language (SMIL 2.0)仕様	http://www.w3c.org	http://www.w3c.org	2001-03-28T00.00.00	This is a W3C Working Draft of the Synchronized Multimedia Integration Language (SMIL) 2.0 specification produced by the SYMM Working Group (members only). This specification

HTMLから

SVGから

NewsMLから

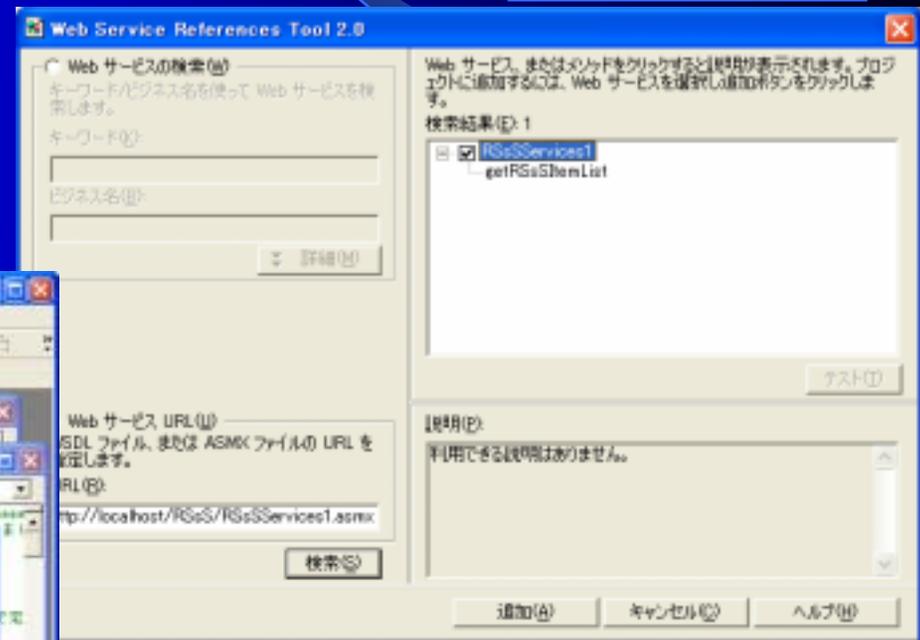
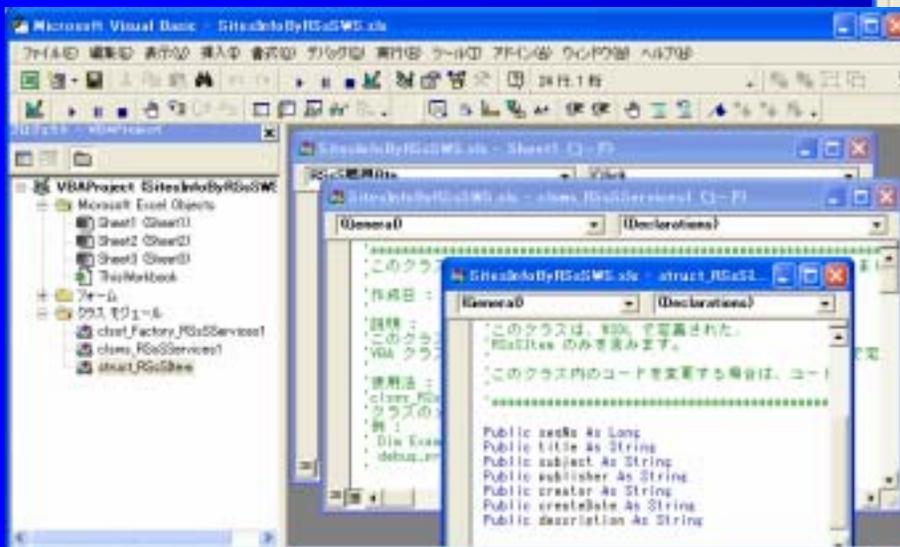
EXCEL VBAでの利用

```
r = startRow  
c = startCol  
Set root = doc.documentElement  
Set itemList = doc.getElementsByTagName("rdf: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
```


EXCEL VBAから Webサービスを呼び出す

- Office XP Web Services Toolkit 2.0 を利用しプロキシクラスを自動生成
 - clws_RSsSServices1
 - struct_RSsSItem

WSDLの参照



プロキシクラス自動生成

- プロキシクラスを利用することにより, XML, SOAP, WSDLと言った知識が無くても利用可能

VBAによるWebサービスの呼出し

```
Dim svcRSsS As New clsWS_RSsSServices1
```

```
Dim item() As struct_RSsSItem
```

```
Dim itemCount As Integer
```

```
itemCount = svcRSsS.wsm_getRSsSItemList( item )
```

```
r = startRow
```

```
For i = 0 To itemCount - 1
```

```
  c = startCol
```

```
  Cells(r, c + 0).Value = i + 1
```

```
  Cells(r, c + 1).Value = item(i).title
```

```
  Cells(r, c + 2).Value = item(i).subject
```

```
  Cells(r, c + 3).Value = item(i).publisher
```

```
  .....
```

```
  r = r + 1
```

```
Next
```

Webサービスプロキシーを呼出し戻り値をitemCountに設定

出力されたitemの内容をあるデータをセルに設定

RSsSからPDFへの変換

● 変換の手順

– RSsSからJepaXへの変換

- XSLTによる変換。スタイルシートを作成

– JepaXからXSL-FOへの変換

- XSLTによる変換。2001年度の成果を利用

– XSL-FOからPDFへの変換

- Apache XMLプロジェクトのFOPを利用。
- 2001年度の成果を利用

注記

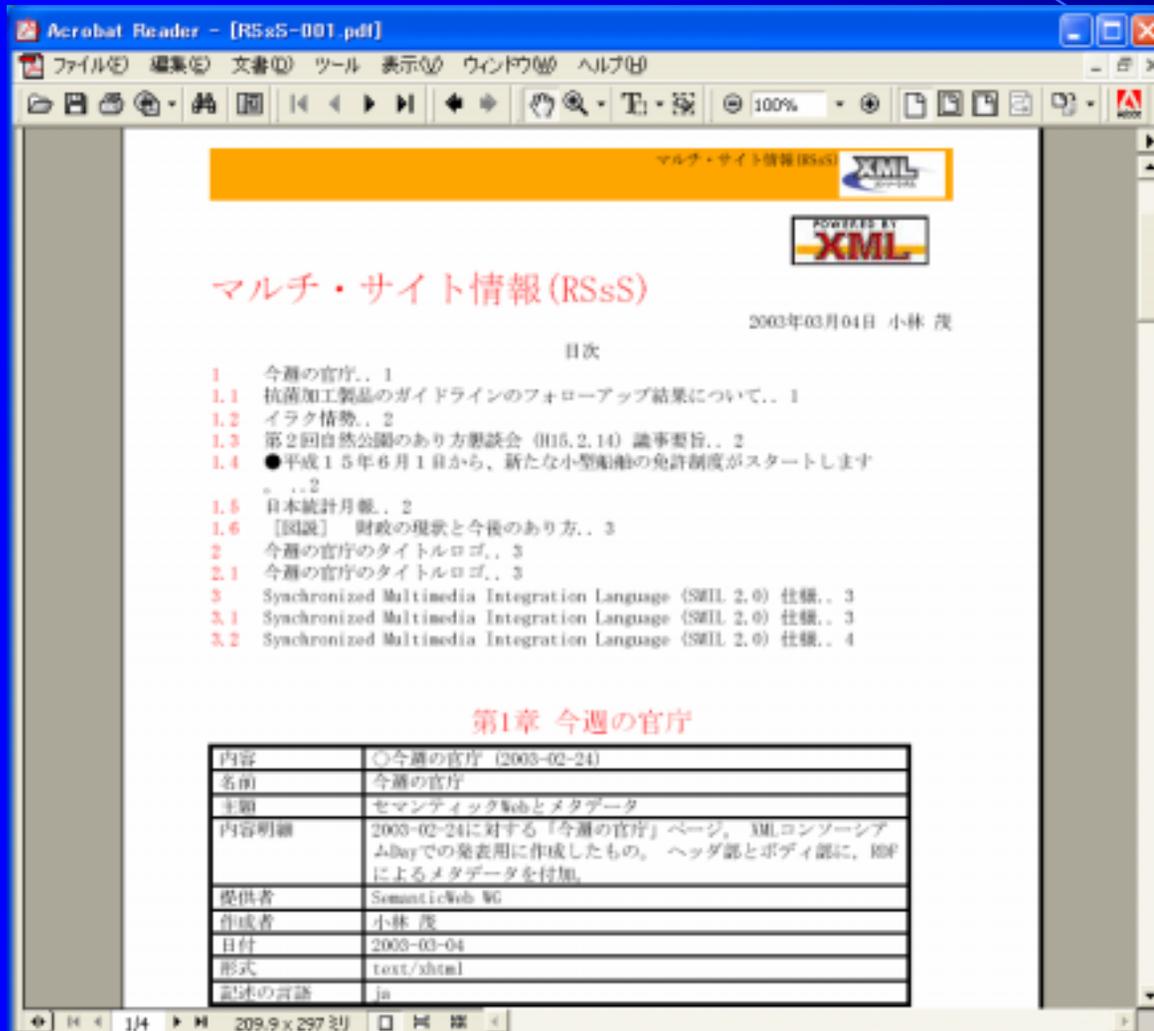
2002年6月XMLコンソーシアムWeekで発表(加藤, 小林)
基盤技術部会 共通基盤WG XSL / XSLT SWG

セマンティックWebとメタデータ

RSsSからJepaXへ変換する スタイルシート

```
<xsl:template match="rdf:RDF">
  <div type="章">
    <head><title><xsl:value-of select="rss:channel/rss:title"/></title></head>
    <table>
      <tr><th>内容</th><td><xsl:value-of select="rss:channel/rss:description"/></td></tr>
      <xsl:apply-templates select="rss:channel/*" mode="supplementModule"/>
    </table>
    <xsl:for-each select="rss:item">
      <div type="節">
        <head><title><xsl:value-of select="rss:title"/></title></head>
        <table>
          <tr><th>内容</th><td><xsl:value-of select="rss:description"/></td></tr>
          <xsl:apply-templates select="*" mode="supplementModule"/>
        </table>
      </div>
    </xsl:for-each>
  </div>
</xsl:template>
```

RSsSのドキュメント(PDF)化

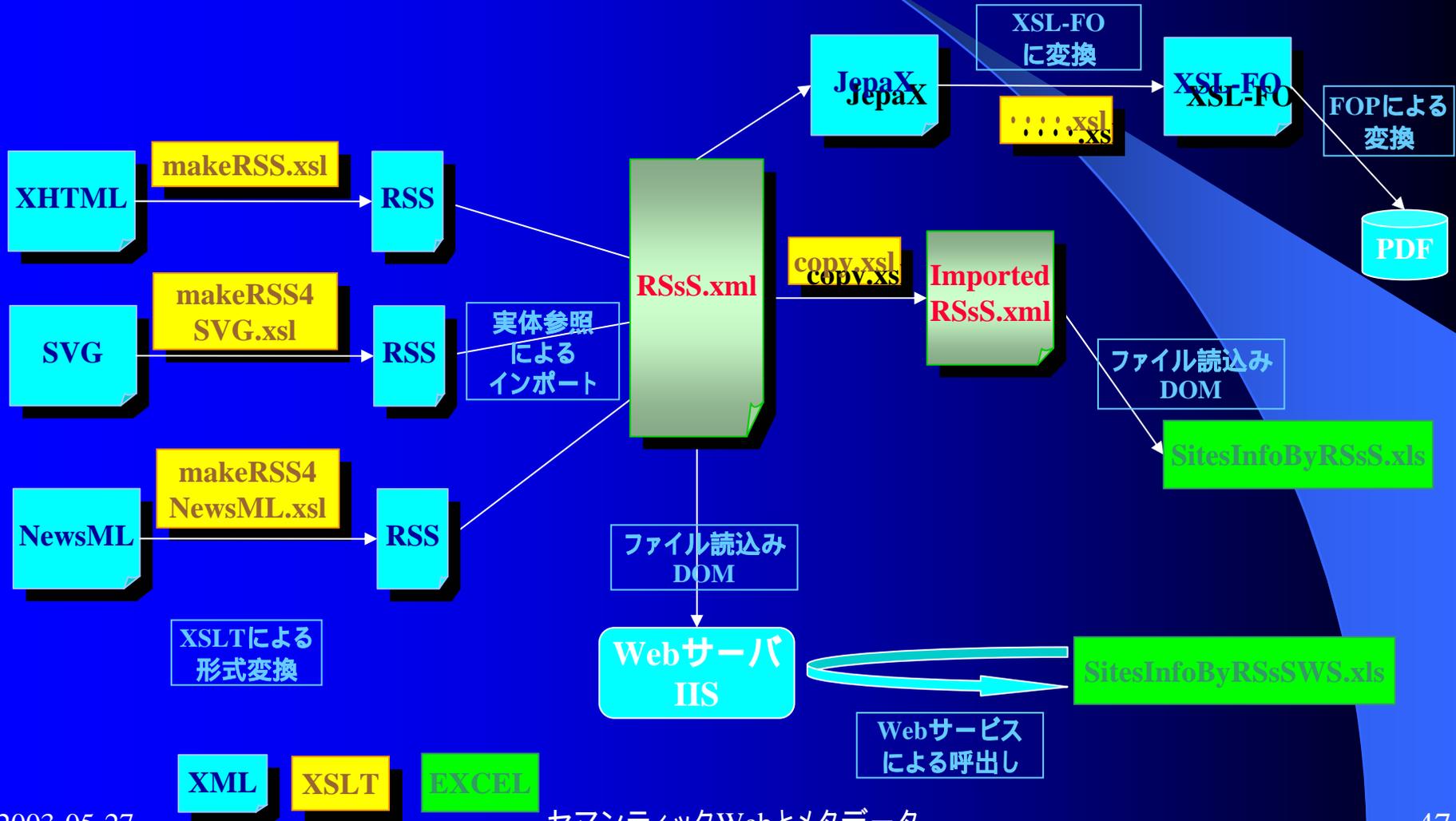


目次の生成

目次から本文
へのリンク付け

本文を表形式
でレイアウト

RSsS生成と利用の概念図



まとめ

- **セマンティックWebはまずメタデータから**
 - まずはお近くのサイトからメタデータ記述を付加しよう
 - ダブリンコアを有効に活用しよう
- **RSS**
 - サイトのメタ情報を表現するものが既に存在するので利用しよう
- **RSsS**
 - 複数のサイト情報をまとめて必要なデータを自動収集しDB登録, Webサービス等での利用も, XMLだから簡単にできそう
- **RDF**
 - メタデータの記述方法を新たに考えるのはちょっと待って
 - RDFにより記述すれば異なるものも統一的な扱いが可能となる
- **そして, 来るべきセマンティックWebの概念がもたらす様々な恩恵にあずかる準備をしよう。**
- **どんな利益を内在しているかを我々のワーキンググループに参加して一緒に考えよう!**

付録:RSSについて

- RSS 0.9
- RSS 0.91
- RSS 1.0

RSS 0.9 (RDF Site Summary)

- ネットスケープ コミュニケーション社が1999年3月に、自社のポータルにチャンネルを登録するための手段として公開したもの
- Webサイトの情報をRDFを利用した簡単なXML形式で記述する。10個の要素だけ。
- channel , item(15個まで) , title , link
- These elements include the notion of a "channel," that contains up to 15 "item"s, each of which consist of a "title" and a "link." Webmasters could use RSS 0.9 files to syndicate their content, using links (no descriptions) to their stories. The format is still quite popular, with about 45% of RSS files surveyed in May, 2001.

RSS 0.9 DTD

```
<!ELEMENT rdf:RDF (channel | image? | item+ | textinput?)*>
<!ATTLIST rdf:RDF
xmlns:rdf CDATA #FIXED "http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns    CDATA #REQUIRED>
    <!-- must be "http://my.netscape.com/rdf/simple/0.9/" --> -->
<!ELEMENT channel (title | description | link)*>
<!ELEMENT title (#PCDATA)>
<!ELEMENT description (#PCDATA)>
<!ELEMENT link (#PCDATA)>
<!ELEMENT image (title | url | link)*>
<!ELEMENT url (#PCDATA)>
<!ELEMENT item (title | link)*>
<!ELEMENT textinput (title | description | name | link)*>
<!ELEMENT name (#PCDATA)>
```

例

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
          xmlns="http://my.netscape.com/rdf/simple/0.9/"
>
  <channel>
    <title>チャンネルの見出し</title>
    <description>内容記述</description>
    <link>リンク先</link>
  </channel>
  <item>
    <title>項目の見出し</title>
    <link>リンク先</link>
  </item>
  …… item要素は最大15個まで……
</rdf:RDF>
```

RSS 0.91 (Rich Site Summary)

- In July 1999 Netscape introduced RSS 0.91 (re-dubbed "Rich Site Summary"), incorporating most of the features of UserLand's <scriptingNews> 2.0b1 format.
- 0.91 added 14 elements to better describe channels, items, and update frequency, including a new "description" element for items (what some call "leads" or "blurbs").

RSS 0.91 DTD

```
<!ELEMENT rss (channel)>
<!ATTLIST rss version CDATA #REQUIRED> <!-- must be "0.91"> -->
<!ELEMENT channel (title | description | link | language | item+ | rating? | image? | textinput? | copyright? | pubDate? |
  lastBuildDate? | docs? | managingEditor? | webMaster? | skipHours? | skipDays?)*>
<!ELEMENT title (#PCDATA)>
<!ELEMENT description (#PCDATA)>
<!ELEMENT link (#PCDATA)>
<!ELEMENT image (title | url | link | width? | height? | description?)*>
<!ELEMENT item (title | link | description)*>
<!ELEMENT textinput (title | description | name | link)*>

<!ELEMENT url (#PCDATA)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT rating (#PCDATA)>
<!ELEMENT language (#PCDATA)>
<!ELEMENT width (#PCDATA)>
<!ELEMENT height (#PCDATA)>
<!ELEMENT copyright (#PCDATA)>
<!ELEMENT pubDate (#PCDATA)>

<!ELEMENT lastBuildDate (#PCDATA)>
<!ELEMENT docs (#PCDATA)>
<!ELEMENT managingEditor (#PCDATA)>
<!ELEMENT webMaster (#PCDATA)>
<!ELEMENT hour (#PCDATA)>
<!ELEMENT day (#PCDATA)>
<!ELEMENT skipHours (hour+)>
<!ELEMENT skipDays (day+)>
```

RSS 0.91の構造

```
<rss version="0.91">
  <channel>
    <title>チャンネルの見出し</title>
    <description>チャンネルの内容</description>
    <link>リンク先</link>
    <pubDate>日付</pubDate> ・・・その他, language, rating等が追加された
    <item>
      <title>項目の見出し</title>
      <description>項目の内容</description>
      <link>リンク先</link>
    </item>
    ・・・ item要素は最大15個まで・・・
  </channel>
</rss>
```

RSS 1.0

- 開発者グループ(RSS-Devワーキンググループ)に引き継がれ, 2000-12-06に, 1.0を公開
 - サイトサマリとしてのコア部分をRSSとして定義
 - RSS 0.9と後方互換性を持つ(0.91ではなく)
 - RDFをベースにする。名前空間を用い, 他のモジュールを利用する
 - Dublin Coreモジュール (`xmlns:dc="http://purl.org/dc/elements/1.1/"`)
 - Syndicationモジュール (`xmlns:sy="http://purl.org/rss/1.0/modules/syndication/"`)
 - Contentモジュール (`xmlns:content="http://purl.org/rss/1.0/modules/content/"`)
 - 他に推奨モジュールがある

関連サイト

- **W3C 各種規格の公開**
<http://www.w3.org/TR/>
- **XMLコンソーシアム**
<http://www.xmlconsortium.org/>
- **メタ情報とセマンティック・ウェブ (RDF,RSS関連)**
 - **W3C RDF Primer**
<http://www.w3.org/TR/rdf-primer/>
 - **RSS 1.0**
<http://web.resource.org/rss/1.0/spec>
 - **神崎正英 氏のホームページ**
<http://www.kanzaki.com/docs/sw/>
 - **What is RSS? (XML.com) rss**
<http://www.xml.com/pub/a/2002/12/18/dive-into-xml.html>