多言語組版研究会第3回資料
StarSuite6.0のXMLと多言語機能について

2003年4月21日
アンテナハウス株式会社
小林　徳滋

要約

1. StarSuite6.0とは何かを説明し、主にStarSuiteWriterのXMLファイルの取得法を説明する。
2. StarSuiteWriterのXMLファイルは再利用が簡単か？例えば、XSLTで簡単にHTML等に変換できるだろうか？この点は、現状では、再利用性が高いとはいえない。
3. StarSuiteWriterの売りである国際化を多言語XML作成という点から見る。多言語XML作成への活用という点でも検討の余地が大きい。
4. 2,3はAutomatic-style機能の問題とそのXML表現の解読の難しさに起因しそうだ。

はじめに

StarSuite6.0は、SunMicrosystemsが販売するオフィス製品である。プログラムはWindowsとLinux、Solarisで動作する。この資料では、Windows版StarSuite6.0について主に取り上げる。ちなみに、この文書はStarSuite Writerで作成して、AcrobatDistillerでPDF化したものである。

StarSuite6.0には下記の機能が含まれる。

1) StarSuite Writer　　ワープドプロセサ
2) StarSuite Calc　　表計算
3) StarSuite Impress　　プレゼンテーション
4) StarSuite Draw　　図形描画ソフト

OpenOfficeとStarSuite6.0について

StarSuiteは、元々は、ドイツのStarDivision（StarOfficeのオリジナル）をSunMicrosystemsが買収してStarOffice5.2として2000年6月に発売したものがベースである。SunMicrosystemsは、StarOfficeの強化にあたり、OpenOffice.orgを立ち上げオープン・ソース運動を利用する形で開発を進めてきた。

http://www.openoffice.org/
http://ja.openoffice.org/　（日本語情報）

OpenOfficeプロジェクトの目的は、次のように述べられている。

「コミュニティとして、すべての主要なプラットフォーム（OS）上で動き、オープンでコンポーネント・ベースのAPIとXMLベースのファイル形式によって、すべての機能とデータへのアクセスを提供するような、主導的かつ国際的なオフィス・スイートを作り出す。」

現在、OpenOfficeのWebページからWindows、Linux、Solaris、MacOSX（ベータ）用のOpenOffice実行形式プログラム、およびソースプログラムやドキュメント、SDKを入手できる。StarSuite6.0は、OpenOffice.orgのOpenOfficeに対してSunMicrosystems独自の付加価値をつけた市販製品である。

上のように書くといかにもオープン・ソース運動らしく聞こえるが、実態はOpenOfficeプロジェクトはSunMicrosystemsが中核スポンサーとなって運営している。別の言い方をすればOpenOfficeプロジェクトはSunMicrosystemsの別働隊である。たとえば、OpenOffice.orgには、11個のサブ・プロジェクトが掲げられているが、各プロジェクトのリーダは全部Sunのメールアドレスを持っている。OpenOfficeプロジェクトは、オープン・ソース・プロジェクトの姿を借りたSunMicrosystemsのStarSuiteマーケティング活動なのである。このようなマーケティング手法が成功するかどうか、ソフトウエア業界者という観点からは興味深い。

現時点では、StarSuite6.0/OpenOffice は MicrosoftOffice よりも機能においてかなりレベルが低い。しかし、最近、IBM 等も OpenOffice プロジェクトのスポンサーになるという噂もあり、Linux に続いて注目をあつめるオープン・ソース・プロジェクトに成長する可能性も秘めている。

StarSuite6.0 の XML 形式

StarSuite6.0 の大きな特徴は、編集ファイルの保存形式として XML を採用していることである。マニュアルなどのドキュメントを XML で作成する、ということを考えたとき、WYSIWYG で文書を編集できるワープロソフトの文書保存形式が XML になっているということは大きな意味をもつ。そこで、まず、StarSuite6.0 の文書保存形式としての XML について検討してみる。

XML のファイル形式については、「OpenOffice.org XML File Format 1.0 Technical Reference Manual Version 2」2002 年 12 月（SunMicrosystems）として公開されており下記より入手できる。500 ページ強の厚い仕様書である。

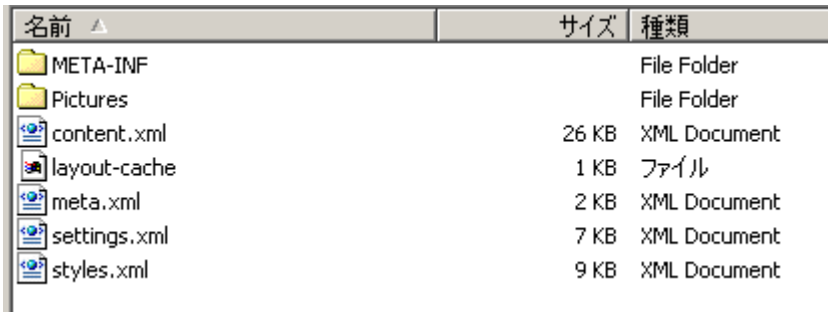
http://xml.openoffice.org/xml_specification.pdf

さらに、SunMicrosystems は、2002 年 12 月に OASIS の技術委員会（OASIS Open Office XML Format TC）に、上記仕様書を提供して、これを Office 文書の標準フォーマット化する活動を始めた。本委員会の議長は Sun Microsystems の Michael Brauer であり、Corel などもメンバーに参加しているが、Microsoft、IBM のメンバーは参加していない。OpenOffice の XML 文書形式が標準フォーマットの地位を得られるかは、まだなんとも言えないが、注目すべき活動である。

保存ファイルと XML 形式

StarSuite6.0 で作成したデータは、アプリケーション毎に ZIP で圧縮したひとつのファイルとしてハードディスク上に生成される。たとえば、StarSuite Writer で作成した文書を「StarSuite6.0 文書ドキュメント」として保存すると、拡張子 sxw のついたファイルができる。このファイルは、多数の XML 形式文書を ZIP 圧縮したものである。sxw ファイルを解凍するといくつかの XML ファイルや、画像ファイルが得られる。
sxw に含まれる XML ファイルは、次のような構成になっている。

meta.xml	ドキュメントの著者、最終更新日時等の情報
styles.xml	ドキュメントで使われるスタイル情報
content.xml	テキスト、表、グラフィック要素などのドキュメントの主要な内容
settings.xml	拡大率、プリンタ指定などのアプリケーションよりの文書と表示の設定
META-INF/manifest.xml	MIME タイプや暗号化の方法などの追加情報
Pictures/	ネィティブのバイナリー形式のイメージを保存するフォルダ
Dialogs/	ドキュメントのマクロで使われるダイヤログを含むフォルダ
Basic/	StarBasic のマクロを含むフォルダ
Obj.../	チャートのような埋め込みオブジェクトを含むフォルダ



このような構成を取っているので、StarSuite6.0 では、①ひとつの文書ファイルの内容が複数の XML に分かれてしまい、また、② XML が ZIP 圧縮ファイルになってしまうということ、など XML を利用したい人にとっては不便な面がある。この点は、OpenOffice1.1 の機能強化項目に XML ファイルの入出力機能があがっているので、次期バージョンでは改良されるだろう。

ドキュメントの内容

content.xml の内容がドキュメント本文である。content.xml ファイルの先頭には、文書型宣言（<!DOCTYPE　..>）があり、DTD として office.dtd が指定されている。content.xml を XML パーサで読むには office.dtd ファイルが必要である。このため、content.xml と同じフォルダに office.dtd ファイルがないとインターネット・エクスプローラ（IE）で開いて見ようとすると、エラーになる。

```
1: <?xml
 :   version="1.0" encoding="UTF-8"
 :  ?>
2: <!DOCTYPE
 :   office:document-content PUBLIC "-
 :   //OpenOffice.org//DTD OfficeDocument 1.0//EN"
 :   "office.dtd"
 :  >
3:
 : <office:document-content
```

Office.dtd はプログラムのソース・ファイルと一緒に配布されているようだが、最新版の DTD は入手していないので、パーサで妥当性の検証ができるかは不明である。ただし、次の図で示すように content.xml では名前空間を使って、W3C の XSL-FO 仕様、XLINK 仕様、SVG 仕様、MathML 仕様で定義している要素型名を取り込んでいる。DTD では名前空間をうまく取り扱うことができないので DTD を入手して検証してもあまり意味がない。

整形式の XML として処理するには、文書型宣言は必要ないので、この行（上の画像の 2 行目）を削除すると IE で開いて見ることができる（次の図）。

```
<?xml version="1.0" encoding="UTF-8" ?>
- <office:document-content xmlns:office="http://openoffice.org/2000/office"
    xmlns:style="http://openoffice.org/2000/style"
    xmlns:text="http://openoffice.org/2000/text"
    xmlns:table="http://openoffice.org/2000/table"
    xmlns:draw="http://openoffice.org/2000/drawing"
    xmlns:fo="http://www.w3.org/1999/XSL/Format"
    xmlns:xlink="http://www.w3.org/1999/xlink"
    xmlns:number="http://openoffice.org/2000/datastyle"
    xmlns:svg="http://www.w3.org/2000/svg"
    xmlns:chart="http://openoffice.org/2000/chart"
    xmlns:dr3d="http://openoffice.org/2000/dr3d"
    xmlns:math="http://www.w3.org/1998/Math/MathML"
    xmlns:form="http://openoffice.org/2000/form"
    xmlns:script="http://openoffice.org/2000/script" office:class="text"
    office:version="1.0">
    <office:script />
  + <office:font-decls>
  + <office:automatic-styles>
  + <office:body>
  </office:document-content>
```

ルート要素型は office:document-content で、その子供に office:body がある。ドキュメント本文のテキスト内容は、office:body の中に入っている。office:body を開いてみると次のようになる。text:p の内容がテキストである。

```
- <office:body>
  - <text:sequence-decls>
      <text:sequence-decl text:display-outline-level="0" text:name="Illustration" />
      <text:sequence-decl text:display-outline-level="0" text:name="Table" />
      <text:sequence-decl text:display-outline-level="0" text:name="Text" />
      <text:sequence-decl text:display-outline-level="0" text:name="Drawing" />
    </text:sequence-decls>
    <text:p text:style-name="P1">多言語組版研究会第3回資料</text:p>
    <text:p text:style-name="P1">StarSuite6.0のXMLと多言語機能について</text:p>
    <text:p text:style-name="P2">2003年4月21日</text:p>
    <text:p text:style-name="P2">アンテナハウス株式会社</text:p>
    <text:p text:style-name="P2">小林　徳滋</text:p>
    <text:p text:style-name="Heading">はじめに</text:p>
    <text:p text:style-name="Text body">StarSuite6.0は、SunMicrosystemsが販売す
      るオフィス製品である。プログラムはWindowsとLinux、Solarisで動作する。この資料では、
      Windows版StarSuite6.0について主に取り上げる。ちなみに、この文書はStarSuite6.0
      で作成して、AcrobatDistillerでPDF化したものである。</text:p>
    <text:p text:style-name="P3">StarSuite6.0には下記の機能が含まれる。</text:p>
```

箇条書きは、次のように text:ordered-list、text:list-item としてタグがついている（次図）。表も表としてタグがついている（図は省略）。

```
- <text:ordered-list text:style-name="L1">
  - <text:list-item>
    - <text:p text:style-name="P4">
        StarSuite Writer
        <text:tab-stop />
        ワープドプロセサ
      </text:p>
    </text:list-item>
```

BMP 形式でペーストした画像は次の図のように PNG 形式のファイルとして出力されている。XML を Web などで配布するために使うならこれで十分だろう。品質の高い印刷物を作ろうとした場合は、画像の取り扱いが十分かどうかは更に調査する必要がある。

```
- <text:p text:style-name="Text body">
    <draw:image draw:style-name="fr1" draw:name="図1" text:anchor-type="paragraph"
      svg:x="1.453cm" svg:y="0.106cm" svg:width="13.679cm" svg:height="4.842cm"
      draw:z-index="0"
      xlink:href="#Pictures/10000000000002050000000B71C694C43.png"
      xlink:type="simple" xlink:show="embed" xlink:actuate="onLoad" />
  </text:p>
```

StarSuite6.0 の XML コンテンツの再利用性

ドキュメントを XML で表現する一般的なメリットは、再利用性、および XSLT プロセサなどで加工を簡単にできることである。単に XML で保存されているというだけでは、バイナリより多少良いという程度である。StarSuite6.0 の XML コンテンツが簡単に再利用できる形式になっているかどうかが問題である。

例えば、content.xml の内容を、簡単に（高度なプログラミング開発を必要とすることな

しに）HTML や XSL-FO に変換できるかどうか？これは XSLT プロセサに詳しいエンジニアによる検討が必要である。ただし、一見したところでは、難しそうだ。一番大きな問題は、テキストを修飾するスタイル情報を text:p 要素の style-name 属性の値として設定しており、さらに、この style-name の値は、StarSuiteWriter が自動的に作成したものが、office:automatic-style の内部に保存されていることである。

たとえば、前の図の番号付き箇条の中で出現している style-name="P4"の値 P4 は、automatic-style の中で次のように定義されている。

```
- <style:style style:name="P4" style:family="paragraph" style:parent-style-
    name="Standard" style:list-style-name="L1">
    <style:properties fo:margin-left="1.111cm" fo:margin-right="0cm" style:font-
      name-asian="MS 明朝" fo:text-indent="0.82cm" style:auto-text-
      indent="false" />
  </style:style>
```

StarSuiteWriter の出力する content.xml を活用する人は、たいていの場合、テキストへの修飾情報を解釈したいだろう。このためには automatic-style の中の各 style:style を解釈する、すなわち、例えば style:name=”P4”というスタイル設定値を、フォント・ファミリー名、フォントのサイズ、などの具体的な項目とその値に展開しなければならない。しかし、XSLT プロセサの標準機能のみでは、このような処理はできないので、専用のプログラムの開発が必要になりそうだ。この観点から見ると、一般のユーザが StarSuiteWriter で作成した XML ファイルを、XSLT などを使って、自在に高度活用できるかという点には疑問がある。

もちろん、OpenOffice.org はオープン・ソースなので公開されているソース・プログラムを調べて、automatic-style を解釈してスタイル情報を展開するプログラム・モジュールを開発することはできるだろう。

StarSuite6.0 の多言語機能

次に多言語機能を調べてみる。StarOffice は、Sun が買収した時点では、ラテン文字セットを扱うローカルなアプリケーションであったが Sun が Unicode 化した。

1.文字の入力

文字の入力は、Windows 版では、Windows の入力方法に依存しているので StarSuiteWriter 独自の工夫はない。

2.コピー・ペースト

StarSuiteWriter は Unicode アプリケーションとして、タイ語、アラビア語、ヘブライ語の文字および文字列を扱うことはできる。たとえば、① Unipad から日本語、ヘブライ語、アラビア語、タイ語、簡体字中国語、繁体字中国語、韓国語をコピーして、② StarSuiteWriter の文書に貼り付けてから文書を保存し、③ content.xml のテキストを取り出す。これを元の Unipad の文字列と比較すると、文字列は同じになる。したがって、クリップボード経由のコピー＆ペーストのレベルであれば、多言語の文字列が混在してもよい。

3.日本語、中国語の混在

Unicode の問題のひとつに「日本語、中国語（簡体字、繁体字）の文字の中で同一のコードポイントに割り振られているものがある」ということがある。これが StarSuiteWriter でどのように取り扱われるかを確認する。StarSuiteWriter の上で日本語、中国語を混在させることはできる。次の行は日本語（下線・実線）、繁体中国語（下線・二重線）、簡体中国語（下線・点線）が混在する。言語は、使用フォントを日本語は MS 明朝、繁体中国語は MingLiU、簡体中国語は SimSun を指定することで区別している。すなわち、表示上のみの指定を行っている。

海に沈む島沈下大海的島嶼沉下大海的岛屿温暖化防止対策溫暖化防止措施温暖化防止措施

上の例で、温暖化防止＊＊の部分のテキストの Unicode のコード値は、次表の通りである。「温」という文字だけ、中国語繁体字は、日本語や中国語簡体字とは異なるコード値が割り振られている。しかし「暖」、「化」、「防」、「仕」の文字コードは 3 言語で同じである。

日本語		繁体字中国語		簡体字中国語	
文字	コード	文字	コード	文字	コード
温	U+6E29	溫	U+6EAB	温	U+6E29
暖	U+6696	暖	U+6696	暖	U+6696
化	U+5316	化	U+5316	化	U+5316
防	U+9632	防	U+9632	防	U+9632
止	U+6B62	止	U+6B62	止	U+6B62
体	U+5BFE	措	U+63AA	措	U+63AA
策	U+7B56	施	U+65BD	施	U+65BD

従って、プレーン・テキスト上では、日本語、繁体字中国語、簡体字中国語は識別不可能である。

この部分は、XML では次のようになっている。

```
- <text:p text:style-name="P13">
    <text:span text:style-name="T2">海に沈む島</text:span>
    <text:span text:style-name="T3">沈下大海的島嶼</text:span>
    <text:span text:style-name="T4">沉下大海的岛屿</text:span>
    <text:span text:style-name="T5">温暖化防止対策</text:span>
    <text:span text:style-name="T6">溫暖化防止措施</text:span>
    <text:span text:style-name="T4">温暖化防止措施</text:span>
  </text:p>
```

P13 には段落の全体のスタイル（フォント・ファミリ名、フォントのサイズ、等）が指定されている。T2...T6 には、文字列に対するスタイル（フォント・ファミリ名、フォントのサイズ、下線の種類等）が指定されている。この XML ファイルから言語情報を取り出すには P13、T2、などのスタイルを解読して、フォント・ファミリ名の指定情報を取り出し、フォント・ファミリ名から言語を判定することになる。ただし、必ずしもフォント・ファミリー名から言語判定ができるとは限らない。これは結構難しそうだ。

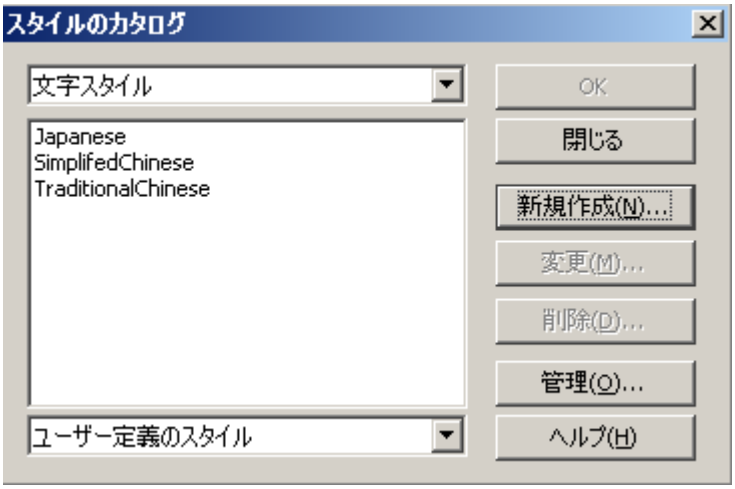
4.スタイル指定で言語を区別

上の問題は日本語、中国語（簡体字、繁体字）の場合だけではなく、ラテン・アルファベットとラテン拡張文字を使う欧文の多言語処理でも同じだろう。もし、西欧語の多言語混在の文書で、言語別にハイフネーション辞書を切り替えようとしたら、フォント・ファミリ名ではなく言語名の情報が必要になる。多言語混在の文書では、文字列の言語名を得るのは重要な課題なのである。

そこで、StarSuiteWriter の機能を使った、ひとつの解決方法を試してみる。

日本語、中国語繁体字、中国語簡体字と 1 対 1 対応する文字スタイルを定義して、そのスタイルを日本語、中国語繁体字、中国語簡体字の部分に、それぞれ適用してみる。

StarSuiteWriter のユーザ定義スタイル機能を利用して、文字のスタイルを次のように 3 種類定義する。文字列に対して、フォントではなく、新たに定義したユーザ定義スタイルを設定する（次頁の図を参照）。



海に沈む島沈下大海的島嶼沉下大海的岛屿温暖化防止対策溫暖化防止措施温暖化防止措施

上の部分の XML の構造は保存した文書の中で次の図のようになっている。これをみると、StarSuiteWriter の「スタイルのカタログ」ダイヤログで現れるスタイル名が、必ずしも XML コンテンツの当該箇所のスタイル名と 1 対 1 対応しているわけではなく、StarSuiteWriter が自動的に新しいスタイル定義を作り出してしまっている。XML の automatic-style には、StarSuiteWriter が作ったスタイル値が出力されているため、「ユーザ定義のスタイル」機能を使って言語を区別するという方法は簡単には使えないようだ。

```
- <text:p text:style-name="P22">
    <text:span text:style-name="T8">海に沈む島</text:span>
    <text:span text:style-name="TraditionalChinese">沈下大海的島嶼</text:span>
    <text:span text:style-name="T9">沉下大海的岛屿</text:span>
    <text:span text:style-name="T10">温暖化防止対策</text:span>
    <text:span text:style-name="TraditionalChinese">溫暖化防止措施</text:span>
    <text:span text:style-name="T11">温暖化防止措施</text:span>
  </text:p>
```

5.日本語画面表示

画面上で正しく組版して表示できるかどうかという点では、StarSuiteWriter の日本語組版機能はあまり優れているとはいえない。というよりも明らかに不完全である。たとえば、この文書の本文は両端揃えを指定しているが、行末がそろっていない。したがって、両端揃え機能が完成しているとはいえない。

6.その他の言語画面表示

タイ語はとりあえず表示はできる。しかし、アラビア語、ヘブライ語の文章は正しく表示できない。文字の進行方向と単語の進行方向が矛盾し、また、双方向性の処理が正しくできていない。これらの言語は、正式には未サポートである。

OpenOffice は現在、V1.1 のβ版が公開されており、追加機能の一覧をみると V1.1 で多言語機能の強化が予定されている。具体的には、タイ語、ヒンディ語、アラビア語、ヘブライ語のような複雑な言語のサポートが追加されると書かれている。今後の対応に期待したい。