

OpenDocument Format (ODF) を 活用した文書の作成と管理

岸 和孝

JAGAT 客員研究員

ODFを活用した文書の作成と管理

- ⇒ OpenOfficeの概要
- ⇒ OpenOfficeの活用
- ⇒ OpenOfficeのケース・スタディ
- ⇒ OpenOfficeの課題

パラダイム・シフト

開放型システムという理想的な情報基盤へのパラダイム・シフト

● 開放型ソフトウェア

Microsoft社の独占による弊害を解消

第1段階 基本ソフトウェアの登場

MS Windows

⇒ Linux

第2段階 応用ソフトウェアの登場

MS Office

⇒ OpenOffice

● XML対応

実行環境に依存しないデータ形式

膨大化・複雑化するデータへの対応

データへの意味付け

内容と体裁・割付けの分離

⇒ 相互運用性の向上

⇒ 可塑性の向上

OpenOfficeの概要

- ⇒ OpenOfficeとは
- ⇒ OpenOfficeとMicrosoft Officeの対比
- ⇒ OpenOfficeのXML対応
- ⇒ OpenOfficeの文書ファイル
- ⇒ ODFの仕様
- ⇒ OpenOfficeの入手

OpenOfficeとは

● 完成されたオフィス・スイート

テキスト



Writer

スプレッドシート



Calc

プレゼンテーション



Impress

ドローイング



Draw

データベース



Base

● 豊富な説明資料（入門書から専門書までサイトで公開）

- 既存の文書データからの変換可能 ⇒ 連携性・移行性の向上
Microsoft Office文書, 一太郎文書, WordPerfect文書, Lotus 1-2-3文書
- 無償で利用可能 ⇒ 費用の削減, デジタル・デバイドの解消
- 各種の実行環境で実現 ⇒ 相互運用性の向上
Windows 98/2000/XP/Vista, Linux, Sun Solaris, FreeBSD, Mac OS X
- オープン・ソース ⇒ 継続性・透明性の向上
Sun Microsystems社のStarOfficeが原型
- 利用者は世界中, 各国の行政府が支持表明 ⇒ 公共性の向上
- Microsoft Officeを代替できるオフィス・スイート
オフィス・スイートの寡占状況における弊害を解消

OpenOfficeとMicrosoft Officeの対比

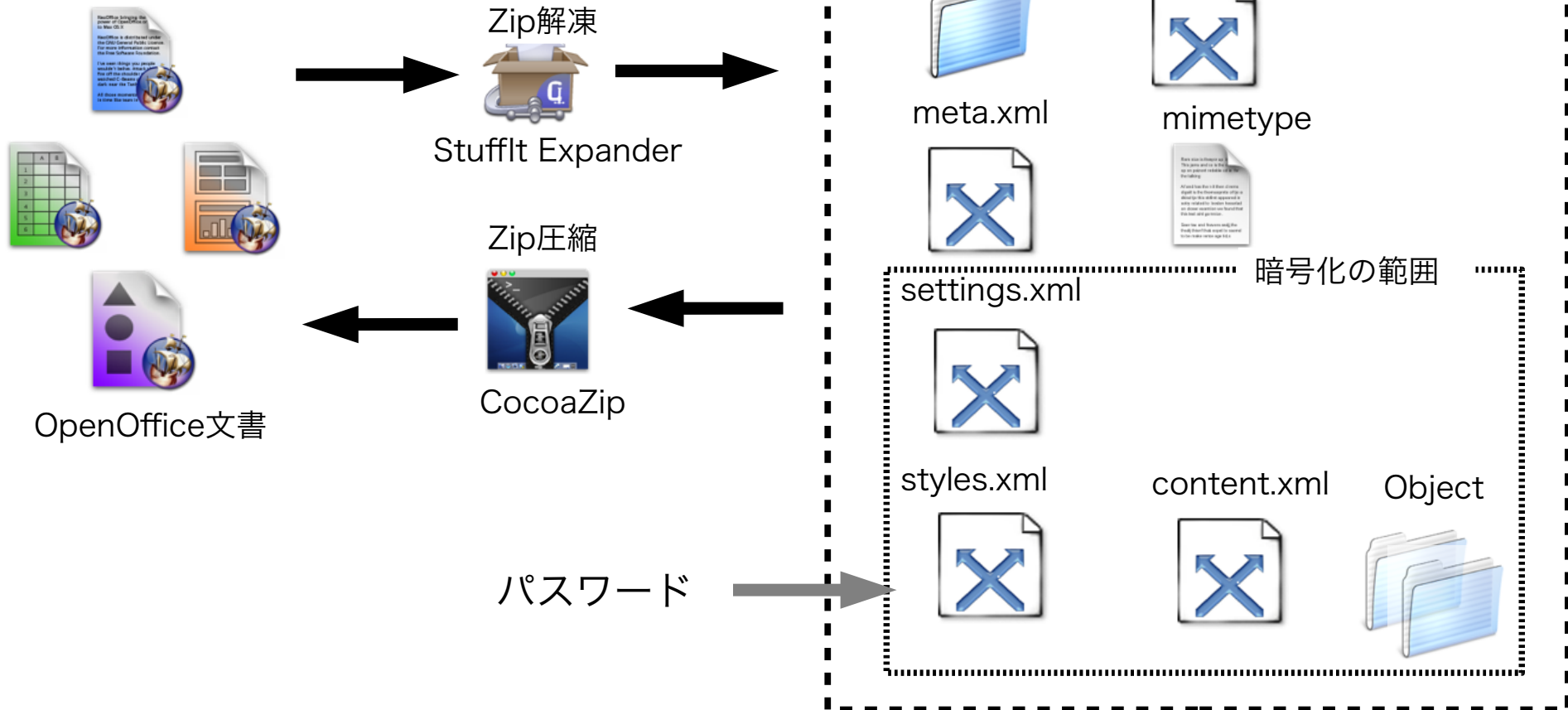
オフィス・スイート		Microsoft Office	OpenOffice.org
機能	テキスト	Word	Writer
	スプレッドシート	Excel	Calc
	プレゼンテーション	PowerPoint	Impress
	電子メール	Outlook	
	流れ図	Visio	Draw
	ドローイング	PhotoDraw	Draw
	数式, 化学式	Equation Editor	Math
	データ管理	Access	Base
	プロジェクト管理	Project	
	DTP	Publisher	Draw
	ウェブ・デザイン	SharePoint Designer	Writer
	コラボレーション	Groove	
	実行環境	Windows	✓
Mac OS X		✓	✓
Linux			✓
BSD			✓
Unix			✓
XML対応		OOXML	ODF

OpenOfficeのXML対応

- OpenOffice文書ファイルそのものがXML対応
Zip形式でパッケージ、小容量化、伝送の高速化
- XML形式で文書交換が可能 ⇒ 再処理可能性の向上
OpenDocument Format (ODF)
ISO/IEC 26300:2006として標準化された
標準のボキャブラリー (FO, SVG, MathML) を包含
ODFは独立した文書ファイル様式であって、OpenOfficeの一部ではない。
- XSLTまたはマクロ (SAX, DOM) によるXMLデータの輸出入
- 様々な表現形式の文書
テキスト, スプレッドシート, プレゼンテーション, ドローイング
内容とスタイルの分離
- Microsoft社のOffice Open XML File Format (OOXML) と競合
ECMA 376 (2006) として標準化された。
その後, 二重標準の批判の中でISO/IEC 29500:2008としても標準化された。

OpenOfficeの文書ファイル

ODFで表された文書データの構成部品



ODFの仕様

Open Document Format for Office Applications (OpenDocument) v1.1

- | | | | |
|----|----------|----|-------------|
| 1 | 概要 | 11 | フォーム内容 |
| 2 | 文書構造 | 12 | 共通内容 |
| 3 | メタデータ要素 | 13 | SMILアニメーション |
| 4 | テキスト内容 | 14 | 体裁 |
| 5 | 段落要素内容 | 15 | 書式付け特性 |
| 6 | テキスト項目 | 16 | データ型とスキーマ定義 |
| 7 | テキスト索引 | 17 | パッケージ |
| 8 | 表 | | |
| 9 | グラフィック内容 | | |
| 10 | チャート内容 | | |

- 1 概要
- 2 文書構造
 - 2.1 文書根
 - 2.2 文書メタデータ
 - 2.3 本文要素と文書型
 - 2.3.1 テキスト文書
 - 2.3.2 ドローイング文書
 - 2.3.3 プレゼンテーション文書
 - 2.3.4 スプレッドシート文書
 - 2.3.5 チャート文書
 - 2.3.6 画像文書
 - 2.4 応用ソフトウェア設定
 - 2.5 スクリプト
 - 2.6 書体の宣言
 - 2.7 体裁
 - 2.8 ページ体裁と割付け
- 3 メタデータ要素
 - 3.1 定義済みのメタデータ要素
 - 3.2 利用者定義のメタデータ
 - 3.3 注文メタデータ
- 4 テキスト内容
 - 4.1 見出し, 段落, 基本的テキスト構造
 - 4.2 ページ列
 - 4.3 箇条書き
 - 4.4 テキスト節
 - 4.5 ページに固定されたグラフィック内容
 - 4.6 変更履歴
 - 4.7 テキスト宣言
- 5 段落要素内容
 - 5.1 基本的テキスト内容
 - 5.2 しおりと参照
 - 5.3 注
 - 5.4 ルビ
 - 5.5 テキスト注釈
 - 5.6 索引マーク
 - 5.7 変更履歴と変更マーク
 - 5.8 行内の画像とテキスト枠

- 6 テキスト欄
 - 6.1 欄要素における共通の特徴
 - 6.2 文書欄
 - 6.3 変数欄
 - 6.4 メタデータ欄
 - 6.5 データベース欄
 - 6.6 その他の欄
 - 6.7 共通欄の属性
- 7 テキスト索引
 - 7.1 索引マーク
 - 7.2 索引構造
 - 7.3 目次
 - 7.4 図表の索引
 - 7.5 表の索引
 - 7.6 オブジェクトの索引
 - 7.7 利用者定義の索引
 - 7.8 アルファベット順の索引
 - 7.9 参考文献
 - 7.10 索引ソースの体裁
 - 7.11 索引表題のテンプレート
 - 7.12 索引テンプレートの項目
- 8 表
 - 8.1 基本の表モデル
 - 8.2 拡張された表モデル
 - 8.3 拡張された表
 - 8.4 拡張された表のセル
 - 8.5 スプレッドシート文書内容
 - 8.6 データベースの範囲
 - 8.7 フィルター
 - 8.8 データ案内表
 - 8.9 併合
 - 8.10 DDE連結
 - 8.11 スプレッドシートにおける変更履歴
- 9 グラフィック内容
 - 9.1 グラフィック応用ソフトウェアのために強化されたページ機能
 - 9.2 図形
 - 9.3 枠
 - 9.4 3次元図形
 - 9.5 注文図形
 - 9.6 プレゼンテーション図形
 - 9.7 プレゼンテーション・アニメーション
 - 9.8 SMILプレゼンテーション・アニメーション
 - 9.9 プレゼンテーション・イベント
 - 9.10 プレゼンテーション・テキスト欄
 - 9.11 プレゼンテーション文書内容

10 チャート内容

- 10.1 チャート文書の概要
- 10.2 チャート
- 10.3 表題, 副題, フッター
- 10.4 凡例
- 10.5 プロット領域
- 10.6 チャートの壁
- 10.7 チャートの床
- 10.8 座標軸
- 10.9 系列
- 10.10 カテゴリー
- 10.11 データ点
- 10.12 平均値
- 10.13 エラー指示子
- 10.14 回帰曲線

11 フォーム内容

- 11.1 フォーム
- 11.2 XFormsモデル
- 11.3 制御
- 11.4 共通のフォーム属性と制御属性
- 11.5 共通の制御属性
- 11.6 イベント
- 11.7 特性

12 共通内容

- 12.1 注釈
- 12.2 数値の形式
- 12.3 変更履歴のメタデータ
- 12.4 イベント・リスナーの表
- 12.5 数学的内容
- 12.6 DDE接続

13 SMILアニメーション

- 13.1 基本のアニメーション要素
- 13.2 アニメーション・モデル属性
- 13.3 共通のアニメーション属性
- 13.4 アニメーション・タイミング
- 13.5 メディア要素
- 13.6 特殊な要素

14 体裁

- 14.1 体裁要素
- 14.2 省略時体裁
- 14.3 ページ割付け
- 14.4 マスター・ページ
- 14.5 表テンプレート
- 14.6 書体宣言
- 14.7 データの体裁
- 14.8 テキストの体裁
- 14.9 強化されたテキストの体裁
- 14.10 箇条書きの体裁
- 14.11 略述の体裁
- 14.12 表の体裁
- 14.13 グラフィックの体裁
- 14.14 強化されたグラフィックの体裁要素
- 14.15 プレゼンテーション・ページの割付け
- 14.16 チャートの体裁

15 書式付け特性

- 15.1 単純な書式付け特性と複雑な書式付け特性
- 15.2 ページ割付け書式付け特性
- 15.3 ページの上部見出し、下部見出し書式付け特性
- 15.4 テキスト書式付け特性
- 15.5 段落書式付け特性
- 15.6 ルビ・テキスト書式付け特性
- 15.7 節書式付け特性
- 15.8 表書式付け特性

- 15.9 表の欄書式付け特性
- 15.10 表の行書式付け特性
- 15.11 表のセル・書式付け特性
- 15.12 箇条書きの体裁特性
- 15.13 線特性
- 15.14 塗りつぶし特性
- 15.15 テキスト・アニメーション特性
- 15.16 テキスト特性とテキスト揃え特性
- 15.17 カラー特性
- 15.18 影特性
- 15.19 線接続部特性
- 15.20 寸法特性
- 15.21 キャプション特性
- 15.22 3次元幾何模様特性
- 15.23 3次元照明特性
- 15.24 3次元外見特性
- 15.25 3次元素材特性
- 15.26 3次元影特性
- 15.27 枠書式付け特性
- 15.28 浮動枠書式付け特性
- 15.29 チャート書式付け特性
- 15.30 チャート・サブタイプ書式付け特性
- 15.31 チャート軸特性
- 15.32 共通のチャート特性
- 15.33 統計的特性
- 15.34 プロット領域特性
- 15.35 回帰曲線の特性
- 15.36 プレゼンテーション・ページ属性

16 データ型とスキーマ定義

17 パッケージ

- 17.1 概要
- 17.2 ZIPファイル構造
- 17.3 暗号化
- 17.4 MIMEタイプのスキーマ
- 17.5 パッケージ内の国際化識別子（IRI）の用法
- 17.6 寸描画像
- 17.7 目録ファイル

OpenOfficeの入手

OpenOffice 3.2.1～

<http://www.openoffice.org/>



NeoOffice 3.1.2～ (Mac OS X)

<http://www.neooffice.org/neojava/ja/>



拡張機能

<http://extensions.services.openoffice.org/ja/>

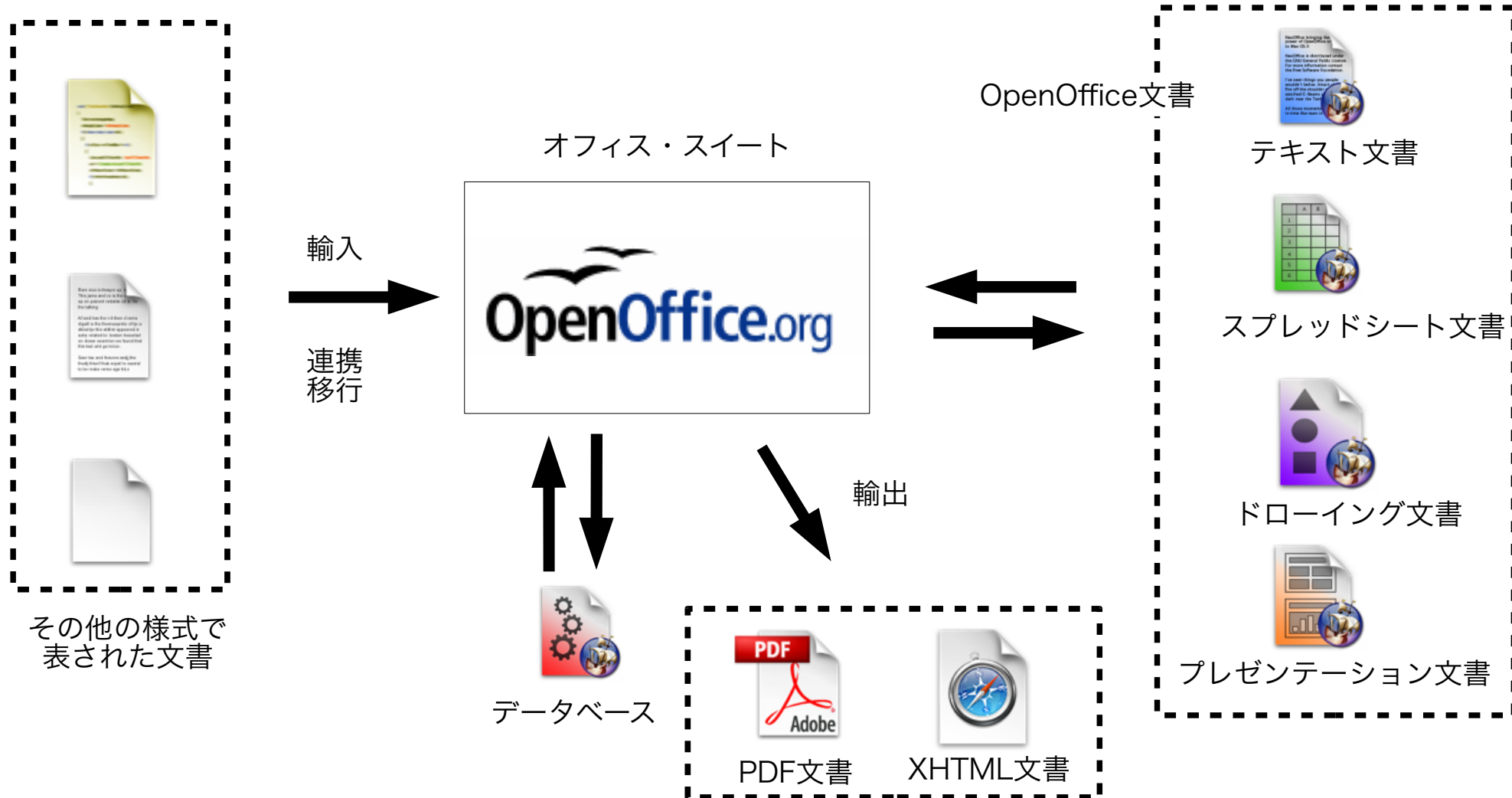
テンプレート

<http://templates.services.openoffice.org/ja/>

OpenOfficeの活用

- ⇒ 新世代オフィス・スイートとしてのOpenOffice
- ⇒ XMLパブリッシング・ツールとしてのOpenOffice

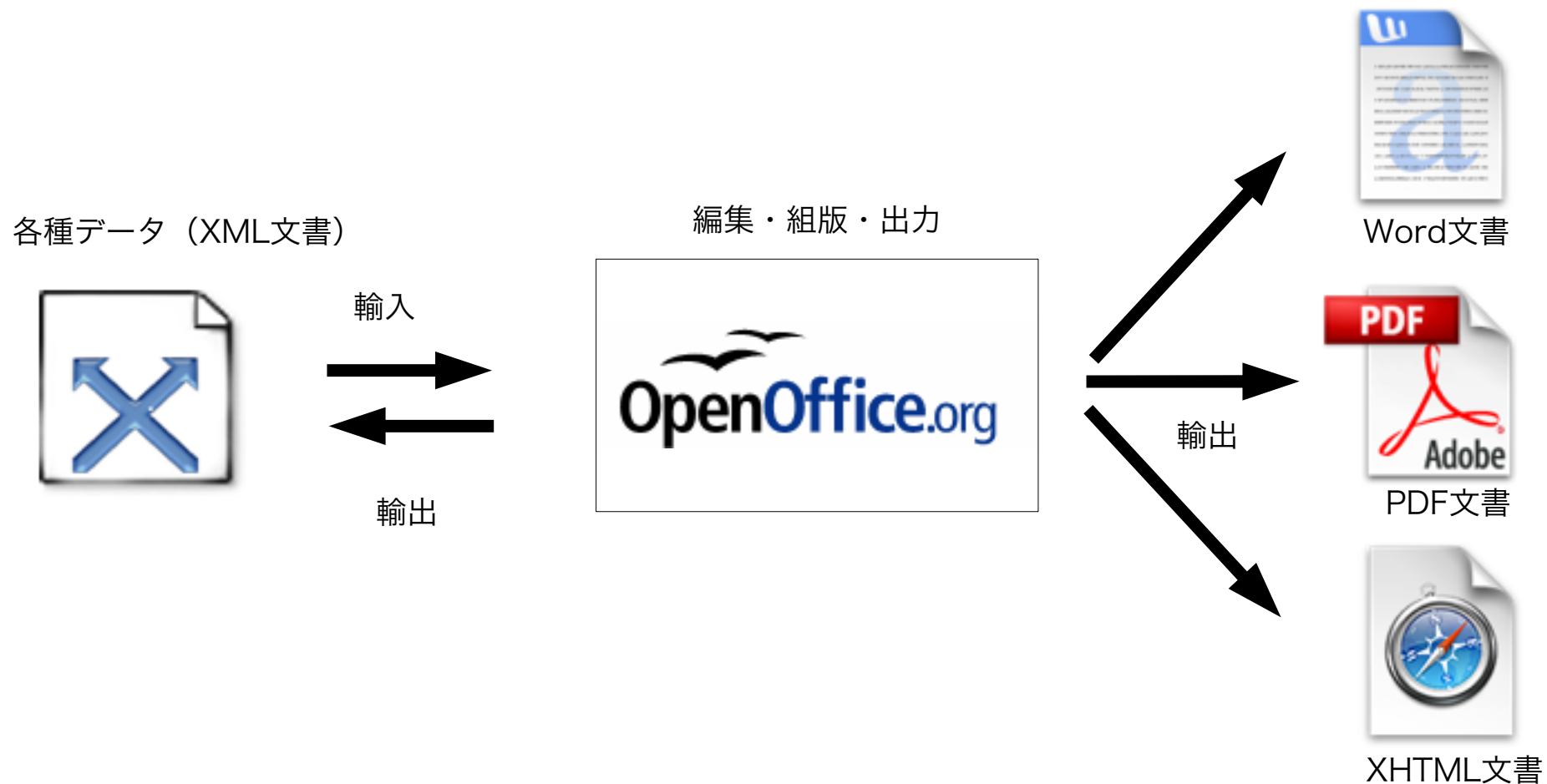
新世代オフィス・スイートとしてのOpenOffice



OpenOfficeが新世代オフィス・スイートに至る想定シナリオ

- OpenOfficeの効用が喧伝される
 - 無償であること
 - Microsoft Officeと互換性があること
 - 中央省庁がOpenOfficeを支持していること
 - XMLによる新しいデータ活用が可能になること
- Microsoft OfficeをOpenOfficeへ代える個人や機関が増える
 - 関心を持つ個人がOpenOfficeを採用する
 - 財政難を解消するために自治体、学校などがOpenOfficeを採用する
 - Microsoft Officeの費用負担を解消するために企業がOpenOfficeを採用する
 - OpenOfficeを採用した大企業系列下にある中小企業がOpenOfficeを採用する
- 既存の文書データとの連携や移行が進む
 - Microsoft Office文書が必要に応じて次第にOpenOfficeへ移行される
 - Microsoft Office文書が必要に応じてOpenOfficeと連携される

XMLパブリッシング・ツールとしてのOpenOffice



OpenOfficeがXMLパブリッシング・ツールに至る想定シナリオ

- OpenOfficeによるXML応用事例が喧伝される
 - XML形式によるデータ交換のメリットが強調される
 - 公文書はXML文書（ODF）であることが強調される
 - データベースへのデータ入力（名簿, 契約書, 届出書）分野から始まる
- OpenOfficeの輸出入用のテンプレートが出回るようになる
 - 情報処理サービス業者がユーザーへテンプレートやフィルターを提供する
 - テンプレートやフィルターの開発・提供サービスが始まる
- ユーザーがOpenOfficeのテンプレートを開発できるようになる
 - XMLへの関心が広まる
 - テンプレートやフィルター開発に関わる書籍が発行される
 - テンプレートやフィルター開発に関わるサイトが増える
- XML形式によるデータ交換が普及する

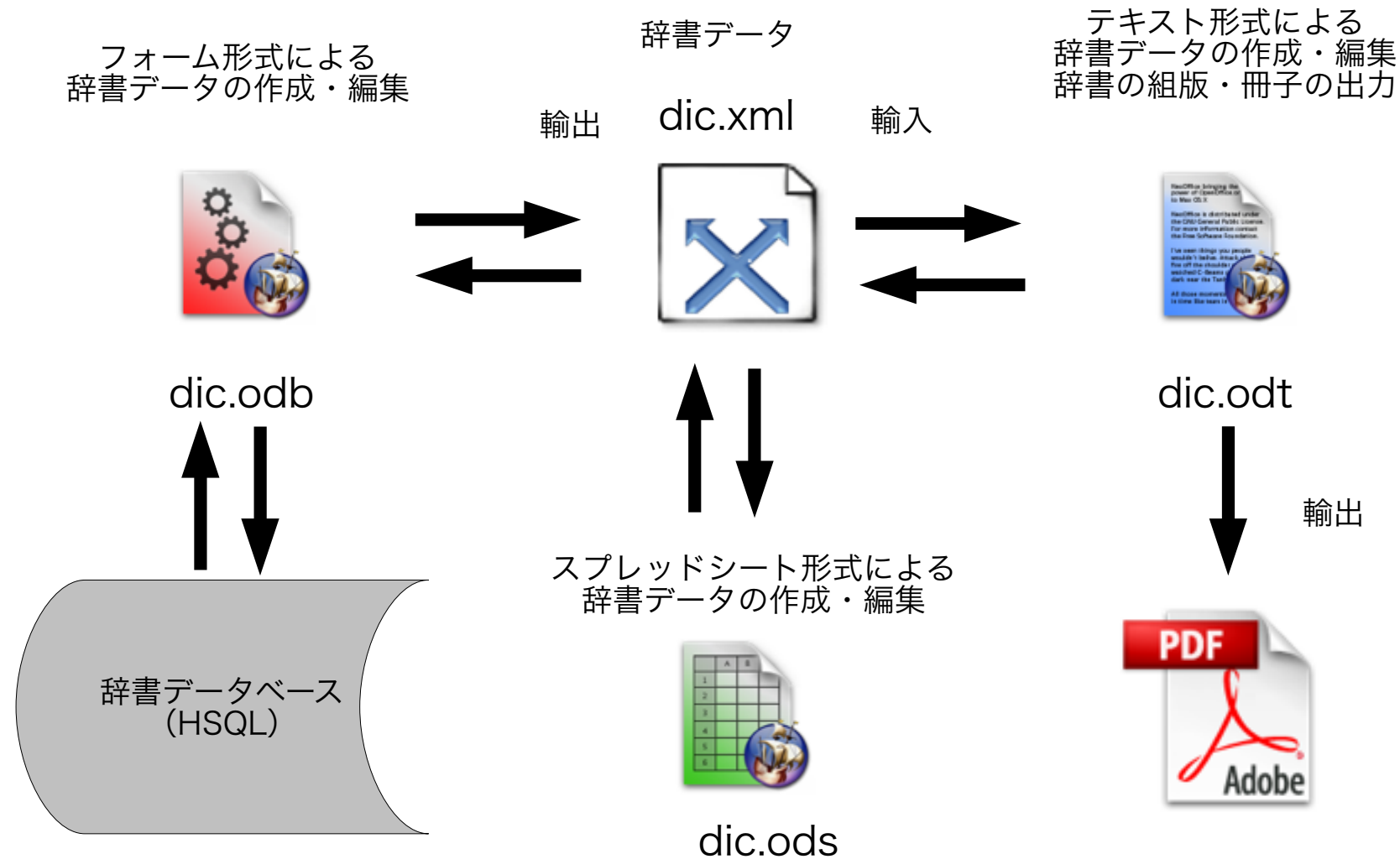
OpenOfficeのケース・スタディ

- ⇒ ケース・スタディの目的
- ⇒ ワーク・フロー
- ⇒ XMLデータ輸出入の仕組み
- ⇒ ケース・スタディの例

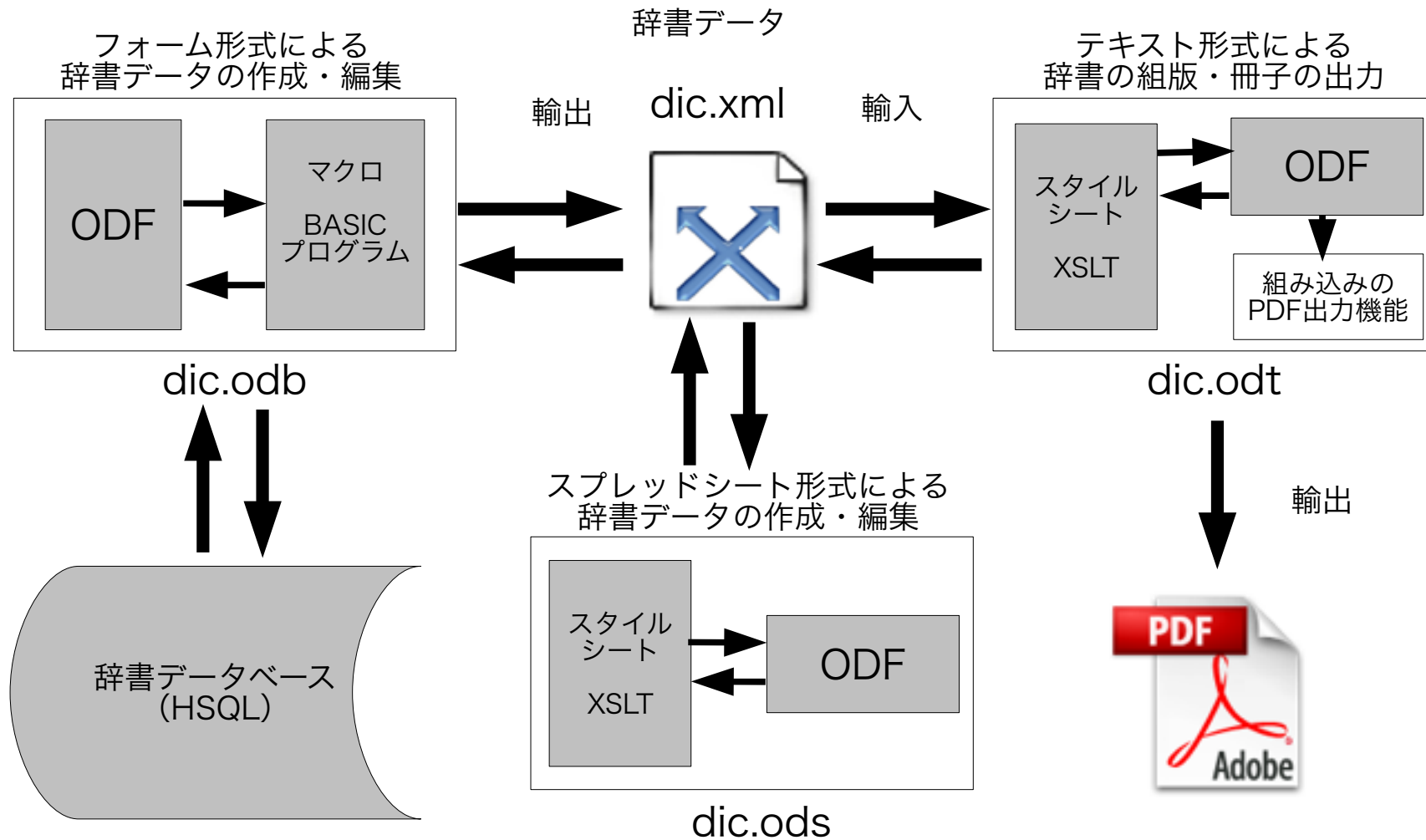
ケース・スタディの目的

- OpenOfficeは実際の仕事に使えるか
このプレゼンテーションとそのPDF
ケース・スタディで用いた辞書データの例
ODF仕様書の和訳
- OpenOfficeは多様な文書形式で表した文書を交換できるか
テキスト, スプレッドシート, フォームを介したデータベース
意味を体裁で表した後, その体裁から意味を再び引き出せるか
- OpenOfficeの輸出入 (export / import) はどのように働くのか
- OpenOfficeにおいてどのような表現が可能か
- OpenOfficeにおいて文字実体の扱いはどうなるか
- OpenOfficeとNeoOfficeは同じか

ワーク・フロー



XMLデータ輸出入の仕組み



開発したフィルターとマクロ

フィルター (スタイルシート)	フィルターの種類	アプリケーション
dictionary to calc	import filter	OpenOffice Calc (.ods)
calc to dictionary	export filter	OpenOffice Calc (.ods)
dictionary to writer	import filter	OpenOffice Writer (.odt)
writer to dictionary	export filter	OpenOffice Writer (.odt)
footnotes to dictionary	export filter	OpenOffice Writer (.odt)

マクロの機能	記述言語	アプリケーション
import	Basic (UNO)	OpenOffice Base (.odb)
export	Basic (UNO)	OpenOffice Base (.odb)

XML文書の変換 (XMLからOpenOffice Writerへ)

XML文書インスタンス

```
<?xml version="1.0" encoding="Shift_JIS"?>
<?xml-stylesheet type="text/xsl"
href="xml2writer.xsl"?>
<名刺>
<名前><姓>桃</姓><名>太郎</名></名前>
<会社名>有限会社 オタカラ商事</会社名>
<役職>社長</役職>
<住所郵便番号="700">瀬戸内海鬼ガ島</住所>
<電話番号>123-4567</電話番号>
<ホームページ>www.onigashima.com</ホーム
ページ>
<ロゴ画像="momo.jpg"/>
</名刺>
```

ODF文書インスタンス

```
<?xml version="1.0"?>
(略)
```

xml2writer.xsl

XSLスタイルシート

```
<?xml version="1.0"?>
(略)
```

XSL変換
(XSLT)OpenOffice
Writer

桃 太郎
社長
有限会社 オタカラ商事
電話：123-4567
ホームページ：www.onigashima.com
〒700 瀬戸内海鬼ガ島

変換を指示するXSLスタイルシート (xml2writer.xsl)

```

<?xml version="1.0"?>
<xsl:stylesheet version="1.0" (略) office:version="1.0">
<xsl:template match="/">
<office:document-content>
<office:font-face-decls> (略) </office:font-face-decls>
<office:styles> (略) </office:styles>
<office:body>
<office:text>
<xsl:apply-templates />
</office:text>
</office:body>
</office:document-content>
</xsl:template>
<xsl:template match="名刺">
<draw:frame draw:style-name="名刺枠体裁"
  text:anchor-type="paragraph" svg:width="9cm"
  svg:height="5.5cm" draw:z-index="0">
<draw:text-box>
<text:p text:style-name="ロゴマーク体裁">
<draw:frame draw:style-name="ロゴマーク画像体裁" text:anchor-
type="as-char" svg:width="{ロゴマーク/@幅}"
  svg:height="{ロゴマーク/@高さ}" draw:z-index="1">
<draw:image xlink:href="{ロゴマーク/@画像}"
  xlink:type="simple" xlink:show="embed"
  xlink:actuate="onLoad"
  draw:filter-name="JPEG - Joint Photographic Experts
Group"/></draw:frame>
</text:p>

```

```

<text:p text:style-name="名前体裁">
<xsl:value-of select="名前/姓"/>&#12288;
<xsl:value-of select="名前/名"/>
</text:p>
<text:p text:style-name="役職体裁">
<xsl:value-of select="役職"/>
</text:p>
<text:p text:style-name="会社名体裁">
<xsl:value-of select="会社名"/>
</text:p>
<text:p text:style-name="基本体裁">
&#x260e;&#12288;
<text:span text:style-name="電話番号体裁">
<xsl:value-of select="電話番号"/>
</text:span>
</text:p>
<text:p text:style-name="ホームページ体裁">
<xsl:value-of select="ホームページ"/>
</text:p>
<text:p text:style-name="基本体裁">
〒<text:span text:style-name="郵便番号体裁">
<xsl:value-of select="住所/@郵便番号"/>
</text:span>&#12288;
<text:span text:style-name="住所体裁">
<xsl:value-of select="住所"/>
</text:span>
</text:p>
</draw:text-box>
</draw:frame>
</xsl:template>
</xsl:stylesheet>

```

変換結果のODF文書インスタンス

```
<?xmlversion="1.0" encoding="Shift_JIS"?>
<fo:root>
<fo:layout-master-set>
<fo:simple-page-sequence-master
  master-name="CARD"
  page-width="90mm" page-height="55mm"
  margin-top="2mm" margin-bottom="2mm"
  margin-left="5mm" margin-right="5mm">
</fo:simple-page-sequence-master>
</fo:layout-master-set>
<fo:page-sequence>
<fo:flow>
<fo:block text-align="right">
<fo:external-graphic href="momo.jpg"/>
</fo:block>
<fo:block text-align="center"
  font-family="平成角ゴシック" font-size="14pt">
桃&#12288;太郎
</fo:block>
<fo:block text-align="center" space-before="3pt"
  font-family="平成明朝" font-size="12pt">
有限会社 オタカラ商事
</fo:block>
<fo:block text-align="center" space-before="3pt"
  font-family="平成明朝" font-size="10pt">
社長
</fo:block>
<fo:block text-align="center" space-before="3pt"
  font-family="平成明朝" font-size="10pt">
〒700&#12288;瀬戸内海鬼ガ島
</fo:block>
```

```
<fo:block text-align="center" space-before="3pt"
  font-family="平成明朝" font-size="10pt">
電話：123-4567
</fo:block>
<fo:block text-align="center" space-before="3pt"
  font-family="平成明朝"font-size="10pt">
ホームページ：www.onigashima.com
</fo:block>
</fo:flow>
</fo:page-sequence>
</fo:root>
```

意味と体裁の対応付け (1)

入力されるXML文書インスタンス

…<a>荒海や…佐渡に…<c>横たふ</c>…<d>天の川</d>…

要素名に対応する体裁名の体裁で可視化する。

同じように見える体裁を持つ
ODFに対応させるが、それぞ
れ体裁名は異なる。

…荒海や…佐渡に…横たふ…天の川…

体裁名に対応する要素名へ変換する。

輸出されるXML文書インスタンス

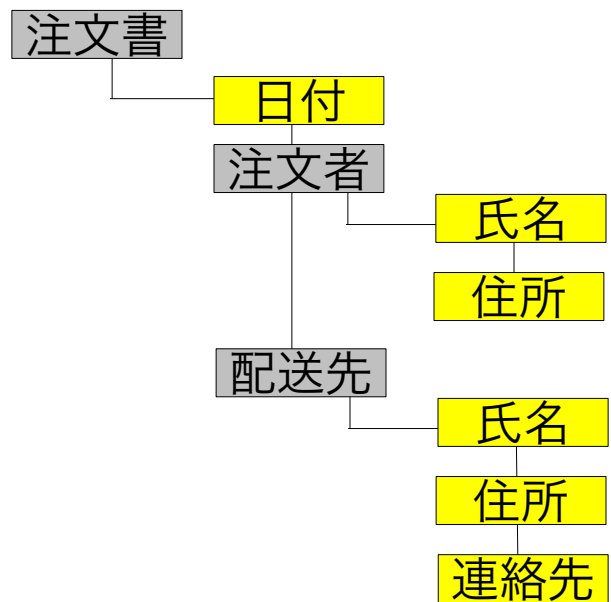
…<a>荒海や…佐渡に…<c>横たふ</c>…<d>天の川</d>…

意味と体裁の対応付け (2)

利用者定義の文書構造

意味を表す要素名を持つ要素から構成される。

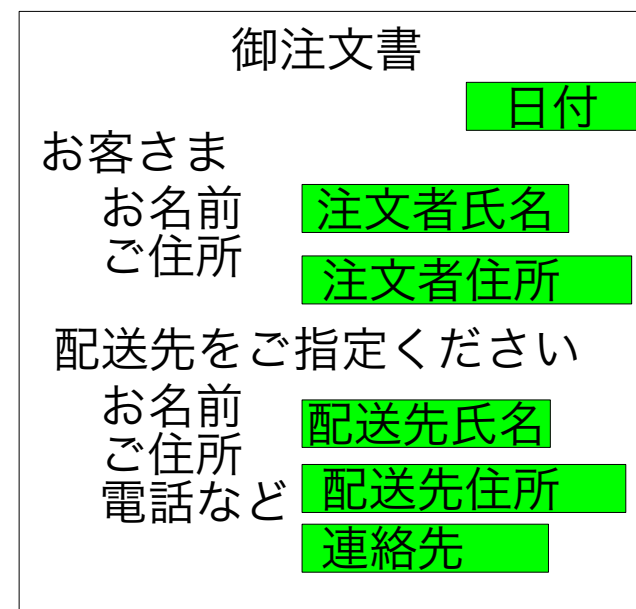
ここで、「氏名」と「住所」が親要素を別とする子要素であることに注目されたい。



ODFの文書構造

利用者定義の要素に一意に対応する体裁名を持つ要素から構成する。

ここで、「注文者」の子要素「氏名」と「住所」に対応する「注文者氏名」と「注文者住所」と、「配送先」の子要素「氏名」と「住所」に対応する「配送先氏名」と「配送先住所」が区別されなければならないことに注目されたい。



辞書データの例

```
<?xml version='1.0'?>
<dictionary title="XML+SGML用語辞典" author="Neil Bradley" translator="関西SGMLビジ
ネス懇談会" date="1999年12月1日" version="初版">
<preface title="はじめに"> (略) </preface>
<entry>
<headword>マーク宣言</headword>
<synonym>markup declaration</synonym>
<syllable>まーくせんげん</syllable>
<description><p>文書にマーク付けするために用いられるのではなく、文書構造規則
(<r>DTD</r>)を作ったり、それぞれの<r>実体</r>を識別して見つけ出したり、代替文書
の部分を定義したりするといった、別の多くの目的のために用いられる特殊文字タグ。</p>
<p>省略時は、“<example><![CDATA[<![></example>”と“<example>></example>”に
よって区切られる。</p>
<p>(<r>MDO</r>, <r>MDC</r>を参照。) </p>
</description>
</entry>
(略)
</dictionary>
```

フォーム形式による辞書データの表現

辞典データベース

(C)2007 KISI

見出し語

Frac要素

同義語

引用

ふりがな

記述

分数を囲む9573 math要素。

分母に対して分子を位置揃えするAlign属性を含む。

▼
`<frac align="left"><numer>x</numer><over>y+z</over></frac>`

▲
 その表示：

`<オブジェクト 場所="50-辞典-図/frac.odg" 幅="1.69cm" 高さ="1.13cm" />`

初期化

入力

輸出

辞書データの輸出入はBaseに組込んだマクロによる。

要素ごとに固定的な項目欄を設定できるので、入力しやすいが、前後関係を見るには不便である。

オブジェクトの挿入はできない。

意味を表す体裁の表現に制約がある。

内外へ連結できない。

スプレッドシート形式による辞書データの表現

	A	B	C	D	E
1	辞書データ				
2	見出し語	同義語	ふりがな	記述	引用
3	DSSSL (文書スタイル意味指定言語)	Document Style Semantics and Specification Language	DSSSL	[“dee-sell” と発音] より限定されたFOS/アプローチに代えて、SGML構造化文書に関係する変換と様式情報を指定するために用いられるISO標準 (ISO/IEC 10179:1995)。1996年4月に公開された。その言語はSchemeプログラミング言語に基づく。最近、XMLスタイル指定の基準として用いられた。 http://occam.sjf.novell.com:8080/dsssl/dsssl96 を参照。 簡潔な手引については http://itrc.uwaterloo.co:80/~papresco/dsssl/tutorial.html を参照。 DSSSL Onlineを参照。	
4	DTD		DTD	[1] 少なくとも文書型定義の部分を含む実体として公的公開識別子によって識別される外部実体を識別するキーワードの予約名。 [2] 文書の個別の型に関する規則を成文化する命令。 タグを付けられた文書が定義済みの文書構造規則に適合するかどうかの検査のためにパーザーによって用いられる。 前書きにおいて、幾つかの要素宣言と属性宣言コードを用いて作り上げられる。 文書を組み立て、ページ組みするように設計された応用。 通常はWYSIWYGインターフェイスを使うデスクトップコンピューター上で操作される。	
5	DTP (デスクトップパブリッシング)	DeskTop Publishing	DTP		

辞書データの輸出入はCalcに組込んだフィルターによる。

一覧性があり、データ入力はやりやすい。

オブジェクトの挿入には無理がある。

意味を表す体裁の表現には余り制約はないが、段落（セル内の改行）の入力が不便である。

内外へ連結できない。

スプレッドシート形式のXML表現

```

<?xml version="1.0" encoding="UTF-8"?>
<office:document-content ... >
  (略)
<office:body>
<office:spreadsheet>
<table:table table:name="表1" table:style-name="ta1"
table:print="false">
<office:forms form:automatic-focus="false"
form:apply-design-mode="false"/>
<table:table-column table:style-name="co1"
table:default-cell-style-name="ENTRY_5f_HEADWORD"/>
<table:table-column table:style-name="co2"
table:default-cell-style-name="ENTRY_5f_SYNONYM"/>
<table:table-column table:style-name="co2"
table:default-cell-style-name="ENTRY_5f_SYLLABLE"/>
<table:table-column table:style-name="co3"
table:default-cell-style-name="ENTRY_5f_DESCRIPTION"/>
<table:table-column table:style-name="co2"
table:default-cell-style-name="ENTRY_5f_REFERENCES"/>
  (略)
<table:table-row table:style-name="ro2">
<table:table-cell office:value-type="string">
<text:p>DSSSL (文書スタイル意味指定言語) </text:p>
</table:table-cell>
<table:table-cell office:value-type="string">
<text:p>Document Style Semantics and Specification
Language</text:p>
</table:table-cell>
<table:table-cell office:value-type="string">
<text:p>DSSSL</text:p>
</table:table-cell>

```

```

<table:table-cell office:value-type="string">
<text:p> [ “dee-sell” と発音] </text:p>
<text:p>より限定されたFOSIアプローチに代えて，SGML構造化文書に関する変換と様式情報を指定するために用いられるISO標準 (ISO/IEC 10179:1995)。</text:p>
<text:p>1996年4月に公開された。</text:p>
<text:p>その言語はSchemeプログラミング言語に基づく。</text:p>
<text:p>最近，XMLスタイル指定の基準として用いられた。</text:p>
<text:p>
<text:a
xlink:href="http://occam.sjf.novell.com:8080/dsssl/dsssl96">http://occam.sjf.novell.com:8080/dsssl/dsssl96</text:a>を参照。</text:p>
<text:p>簡潔な手引については<text:a
xlink:href="http://itrc.uwaterloo.co:80/~papresco/dsssl/tutorial.html">http://itrc.uwaterloo.co:80/~papresco/dsssl/tutorial.html</text:a>を参照。</text:p>
<text:p>DSSSL Onlineを参照。</text:p>
</table:table-cell>
<table:table-cell/>
</table:table-row>
  (略)
</office:spreadsheet>
</office:body>
</office:document-content>

```

テキスト形式による辞書データの表現

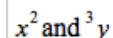
Sup 要素

[1] [上付きテキスト](#)を含んでいる [9573 math](#) 要素。

`Pos` 属性は、基底文字の（後の代わりに）前か上に再位置付けされる上付きテキストを可能にする。

```
x<sup>2</sup> and y<sup pos="pre">3</sup>
```

その表示：



[2] [上付きテキスト](#)を含んでいる [HTML 3.2](#) 要素。

XPath (XML パス言語)

XML Path Language

[XSLT](#) と [XPath](#) によって用いられるように設計された、XML 文書の部分を場所指定するための言語（案）。

<http://www.w3.org/TR/xpath> を参照。

マーク宣言

markup declaration

文書にマーク付けするために用いられるのではなく、文書構造規則 (DTD) を作ったりそれぞれの実体を識別して見つけ出したり、代替文書の部分を定義したりするといった、別の多くの目的のために用いられる特殊文字タグ。

省略時は、“<!” と “>” によって区切られる。

辞書データの輸出入は Writer に組込んだフィルターによる。

大量のデータ入力には向かない。

オブジェクトの挿入は自由である。

冊子にふさわしい体裁がとれる。

内外へ連結できる。

テキスト形式のXML表現

```
<?xml version="1.0" encoding="UTF-8"?>
<office:document-content ... >
  (略)
<office:body>
<office:text>
  (略)
<table:table table:name="表1" table:style-name="表1">
<table:table-column table:style-name="表1.A"/>
<table:table-column table:style-name="表1.B"/>
<table:table-row>
<table:table-cell office:value-type="string">
<text:h text:style-name="ENTRY_5f_HEADWORD"
text:outline-level="1">
<text:bookmark text:name="Sup要素"/>
<text:s/>Sup要素 </text:h>
</table:table-cell>
<table:table-cell office:value-type="string">
<text:p text:style-name="ENTRY_5f_SYNONYM"/>
</table:table-cell>
</table:table-row>
<table:table-row>
<table:table-cell table:number-columns-spanned="2"
office:value-type="string">
<text:p text:style-name="P">[1] <text:a xlink:type="simple"
xlink:href="#上付き">上付き</text:a>テキストを含んでいる<text:a
xlink:type="simple" xlink:href="#9573 math">9573 math</text:a>
要素。</text:p>

<text:p text:style-name="P">その表示 : </text:p>
<text:p text:style-name="P">
  (略)
```

```
<draw:frame draw:style-name="fr1" draw:name="1"
text:anchor-type="paragraph" svg:x="2cm"
svg:y="0cm" svg:width="1.829cm"
svg:height="1.221cm" draw:z-index="0">
<draw:object xlink:href=" ../objects/sup.odg"
xlink:type="simple" xlink:show="embed"
xlink:actuate="onLoad"/>
<draw:image
xlink:href=" ./ObjectReplacements/Object 1"
xlink:type="simple" xlink:show="embed"
xlink:actuate="onLoad"/>
</draw:frame>
</text:p>
  (略)
</table:table-cell>
<table:covered-table-cell/>
</table:table-row>
<table:table-row>
<table:table-cell office:value-type="string">
<text:p text:style-name="ENTRY_5f_SYLLABLE">Supようそ
</text:p>
</table:table-cell>
<table:table-cell office:value-type="string">
<text:p text:style-name="Standard"/>
</table:table-cell>
</table:table-row>
</table:table>
  (略)
</office:text>
</office:body>
</office:document-content>
```

PDF形式による辞書データの表現

Sup 要素

[1] [上付きテキスト](#)を含んでいる [9573_math](#) 要素。

`Pos` 属性は、基底文字の（後の代わりに）前か上に再位置付けされる上付きテキストを可能にする。

```
x<sup>2</sup> and y<sup pos="pre">3</sup>
```

その表示：

$$x^2 \text{ and } ^3y$$

[2] [上付きテキスト](#)を含んでいる [HTML 3.2](#) 要素。

XPath (XMLパス言語)

XML Path Language

[XSLT](#)と[XPointer](#)によって用いられるように設計された、XML文書の部分を場所指定するための言語（案）。

<http://www.w3.org/TR/xpath>を参照。

マーク宣言

markup declaration

文書にマーク付けするために用いられるのではなく、文書構造規則（[DTD](#)）を作ったり、それぞれの実体を識別して見つけ出したり、代替文書の部分を定義したりするといった、別の多くの目的のために用いられる特殊文字タグ。

省略時は、“`<!`”と“`>`”によって区切られる。

([MDO](#), [MDC](#)を参照。)

☰ Search

Sup要素

XPath (XMLパス言語)

マーク宣言

一般実体参照

辞書データの輸出はWriterに組み込み済みのフィルターによる。

Writerで組版した結果から得られる。

冊子にふさわしい体裁がとれる。

しおりを意味付けに合わせて自動的に生成できる。

内外へ連結できる。

脚注からの辞書データの抽出

変更マークが正規の段落³⁷²の内側にある場合、最初の（そして最も一般的な）事例が生じる。

```
<text:p>荒海や<text:change text:id="example"/>天の川</text:p>
```

元のテキストを再構築する³⁷³ために、二つの<text:p>要素は、変更マークを置換するために開始タグ³⁷⁴と終了タグ³⁷⁵を除き複製される。

```
<text:p>荒海や佐渡に</text:p>
<text:p>横たふ天の川</text:p>
```

変更マークが見出しの内側で現れる場合、複製されたタグが整形形式のXMLを持つことを確認するのに適していることを除き、同じ手順³⁷⁶が続けられる。

```
<text:h>荒海や<text:change text:id="example"/>天の川</text:h>
```

これは、下記のようになる。

```
<text:h>荒海や佐渡に</text:h>
<text:h>横たふ天の川</text:h>
```

辞書データの輸出はWriterに組込んだフィルターによる。

原稿作成時に脚注をスプレッドシート形式で管理（誤記や重複などの確認）できる。

370 be merged

371 word

372 regular paragraph

373 to reconstruct

374 beginning tag

375 ending tag

376 procedure

368	変更追跡	change tracking
369	変更位置要素	change position element
370	合併される	be merged
371	語	word
372	正規の段落	regular paragraph
373	再構築する	to reconstruct
374	開始タグ	beginning tag
375	終了タグ	ending tag
376	手順	procedure
377	そのまゝの形で	verbatim

その他 (1) テキスト形式における縦書き, ルビ

4	3	2	1	番付
高谷山 たかやまさん	岩本山 いわもとさん	大棚山 おほたなさん	誦経山 ずきょうさん	山号
金昌寺 きんじやうじ	常泉寺 じやうせんじ	真福寺 しんぷくじ	妙音寺 みどうおんじ	寺号
あらたかに 詣りて拝む 観世音 一世安楽と 誰もいのらん	あたらかに ひびく滝つ瀬 峰の松風 拝むべし	巡りきて 願いをかけし 大棚の 誓いもふかき 谷川の水	ありがたや 巻ならぬ 法のはな 数は四万部の 寺のいにしえ	ご詠歌
曹洞宗	曹洞宗	曹洞宗	曹洞宗	宗派
音 十面観	聖観音	聖観音	聖観音	本尊
秩父市大字山田 1803	秩父市大字山田 1392	秩父市大字山田 3095	秩父市栃谷 418	住所

秩父三十四カ所

Writerの場合, 縦書きが指定できる。

Writerの場合, (スプレッドシート形式ではない) 表においてルビが指定できる。

その他 (2) スプレッドシート形式における縦書き

秩父三十四カ所

番付	山号	寺号	ご詠歌	宗派	本尊	住所
1	誦経山	妙音寺	善ならぬ 法のはな 数は四万部の 寺のいにしえ	曹洞宗	聖観音	秩父市栃谷418
2	大棚山	真福寺	願いをかけし 大棚の 誓いもぶかき 谷川の氷	曹洞宗	聖観音	秩父市大字山田 3095
3	岩本山	常泉寺	岩本寺と 拝むべし 峰の松風 ひびく滝つ瀬	曹洞宗	聖観音	秩父市大字山田 1392
4	高谷山	金昌寺	詣りて拝む 観世音 二世安楽と 誰もいのらん	曹洞宗	十一面観音	秩父市大字山田 1803

Calcの場合、セル単位に縦書きが指定できる。

Calcの場合、ルビは指定できない。

OpenOfficeの課題

- ⇒ OpenOffice活用で求められるXML関連技術
- ⇒ OpenOfficeの活用による源流戦略
- ⇒ OpenOfficeの課題

OpenOffice活用で求められるXML関連技術

- XSLTプログラミング技術の習得
輸出入フィルターを開発するための技術
XSL <http://www.w3.org/Style/XSL/>
- Basicプログラミング技術の習得
マクロを開発するための技術
Universal Network Objects (UNO) の理解
SAX, DOM, ダイアログ, ファイル入出力
StarSuite 8 Basic プログラミング・ガイド
<http://docs.sun.com/app/docs/coll/1268.1?l=ja>
Useful Macro Information For OpenOffice By Andrew Pitonyak
<http://www.pitonyak.org/oo.php>
- ODFの理解
文書オブジェクト・モデルとしてのインターフェース
Open Document Format for Office Applications (OpenDocument) v1.1
<http://docs.oasis-open.org/office/v1.1/OS/OpenDocument-v1.1.pdf>
OASIS OpenDocument Essentials
<http://books.evc-cit.info/odbook/book.html>

OpenOfficeの活用による源流戦略

- 印刷から見た源流（データの発生現場）で情報の有り様を考える
今までの視点（原稿完成時点）の一つ先へ視点（原稿作成時点）を移す
- 顧客はどのような情報を扱っているのか
それは誰（と誰）が作り出しているものか
- 顧客はどのような情報処理を行っているのか
生産性の高いツールの提供
- 原稿としての情報の質を高めるために
文書構造を確実に伝えてもらう
内容を確実に伝えてもらう
- プリプレスを効率化するXMLによるソリューションを考える
テンプレートや輸出入フィルターの提供

OpenOfficeの課題

- ODF仕様書の和訳
- XMLフィルター（スタイルシート）設計支援ツールの実現
- 複数のマスター・ページの実現
- 電子メール機能の実現

終

多くの人々がODFを活用することを願って