

厚生労働科学研究費補助金

感覚器障害研究事業

多機能高精度自動点訳エンジンの研究

平成20年度 総括研究報告書

研究代表者 石川 准

平成21(2009)年 4月

目 次

I. 総括研究報告		
多機能高精度自動点訳エンジンの開発	-----	1
石川 准		
(資料) Open XML 元文書例	-----	3
(資料) 構造化点訳 XML(SBF) 変換例	-----	3
(資料) 自動点訳出力ファイル例	-----	4
(資料) 住所辞書 例	-----	5
(資料) 日本史辞書 例	-----	6
(資料) 生薬辞書 例	-----	7
(資料) SBF 仕様書	-----	8
(資料) SBF の DTD (document type definition) ver.1.0	-----	17
(資料) 形態素解析の例	-----	18
(資料) 自動点訳ソフトウェア スクリーンショット	-----	19

厚生労働科学研究費補助金（感覚器障害研究事業）
総括研究報告書

多機能高精度自動点訳エンジンの開発

研究代表者 石川 准 静岡県立大学 国際関係学部 教授

研究要旨

XML 文書等からの構造情報、レイアウト情報、テキスト情報の抽出、自動点訳に特化した形態素解析の開発、専門辞書の整備、文書種別自動判別アルゴリズムの改善等により、多機能高精度自動点訳エンジンを実現する。

A. 研究目的

今日、自動点訳ソフトウェアは学校、自治体、企業、点字出版社、点字図書館、ボランティアグループ、視覚障害者個人等により幅広く利用されている。この自動点訳ソフトウェアの核となるのが自動点訳エンジンであるが、その利用範囲は自動点訳にとどまらず、スクリーンリーダーの点字表示、点字携帯端末、視覚障害者用ソフトウェアなどでも多く利用されている。

本研究は、近年の以下のような新しいニーズの高まりに対応する多機能高精度点訳エンジンの開発を目的とする。

1. 各種ドキュメントが有する構造情報、レイアウト情報、テキスト情報等を生かした高精度自動点訳の実現
2. 医学、法学等専門分野の文献の自動点訳の改善
3. 固有名詞と一般名詞の判別及び人名

の変換精度の向上

4. 誤りリスク、候補一覧、ふりがな、文節情報等の提示機能
5. Unicode 文字情報の自動点訳

B. 研究方法

上記目的を達成するために研究初年度である本年度は以下の研究を進めた。

- ・はじめに構造化点訳 XML (Structured Braille Format) の規格策定を行った。(SBF 仕様書と DTD は報告書の末尾に添付する。)
- 次いで、Office Open XML から取得できる構造情報、レイアウト情報、テキスト情報のうち、点訳において意味を持つ情報を抽出した。その上で、Open XML から構造化点訳 XML への変換を実現するコンバータを開発した。
- ・これまで開発してきた自動点訳エンジンの形態素解析アルゴリズムを分析し、

弱点を克服するための改善を行い、形態素解析の改善により点訳精度が向上することを確認した。

・医学・理療辞書、町名大字名、丁目・字名等の住所辞書、社会科、理科等の学校教科書用辞書を整備し、それにより点訳精度が向上することを確認した。

(倫理面への配慮)

研究の過程で知り得た個人情報の守秘義務を遵守した。

C. 研究結果

Open XML等の文書から構造情報、各種レイアウト情報を抽出することで、自動点訳の精度を改善できることが確認できた。

形態素解析の改善(助詞、助動詞の判別精度改善)により、点訳精度は一定程度向上したが、一般名詞と固有名詞の判別等、なお改善を要する。

専門辞書は当該分野の文書の点訳では効果を発揮するが、他分野の文書に適用すると、むしろ変換精度の低下を招くので、分野種別を自動判別するか、判別精度が悪い場合には、ユーザが明示的に指定できるようにする必要がある。

D. 考察

多機能高精度自動点訳エンジンの利用範囲は、自動点訳デスクトップパブリッシング、スクリーンリーダー、点字携帯端末、DAISYプレイヤー、携帯電話、電子読書プレイヤー、放送、ウェブサービス等と幅広い。

多機能高精度自動点訳エンジンを実装

した情報機器、情報サービスは、視覚障害者の就労支援、高等教育支援等に資する。

E. 結論

初年度の研究は多機能高精度自動点訳エンジン開発の核となる構造情報・レイアウト情報、テキスト情報の効果的活用、点訳に特化した形態素解析、専門分野辞書の構築において、それぞれ成果があった。

F. 健康危険情報

該当しない。

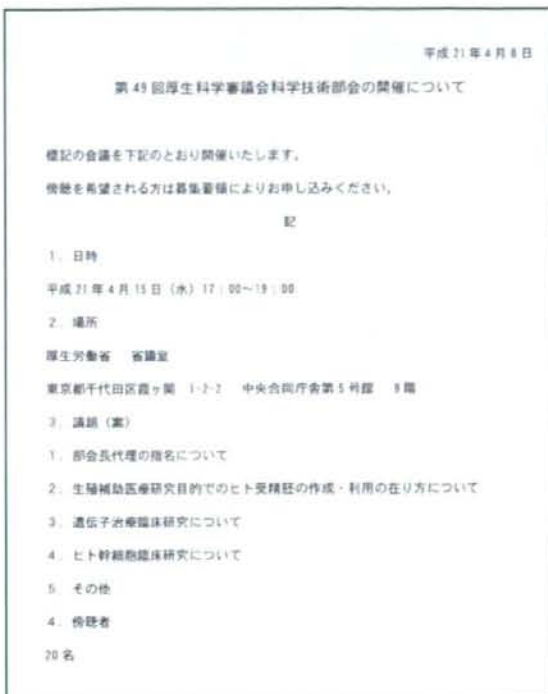
G. 研究発表

なし

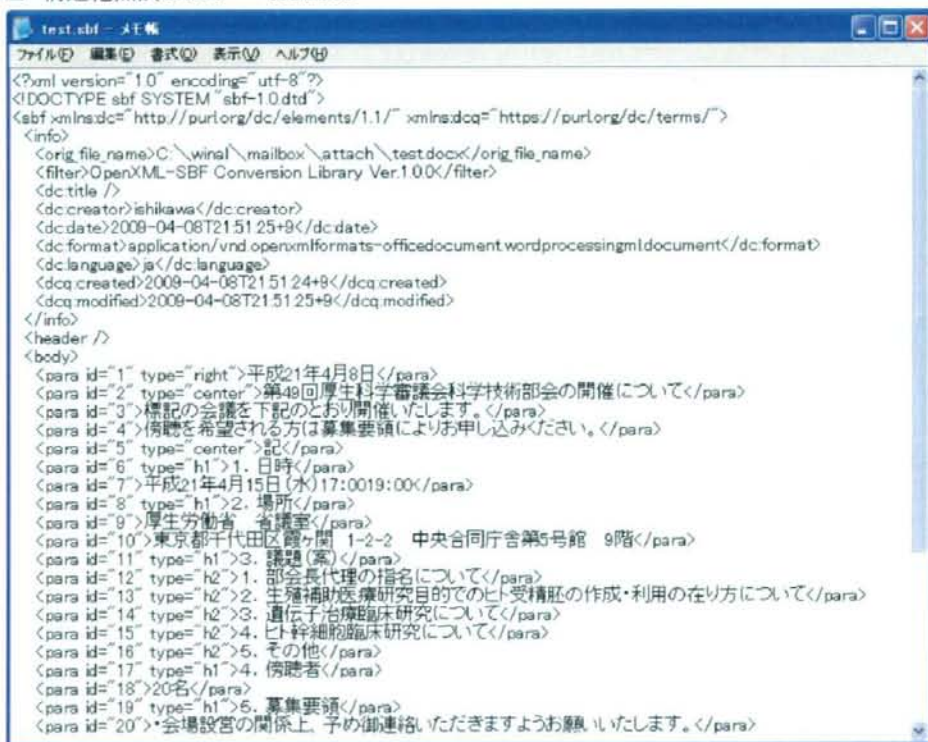
H. 知的財産権の出願・登録状況

なし

■ Open XML 元文書
test.docx



■ 構造化点訳 XML test.sbf



■ 自動点訳出力ファイル (Extra5 形式) test.exf2

```

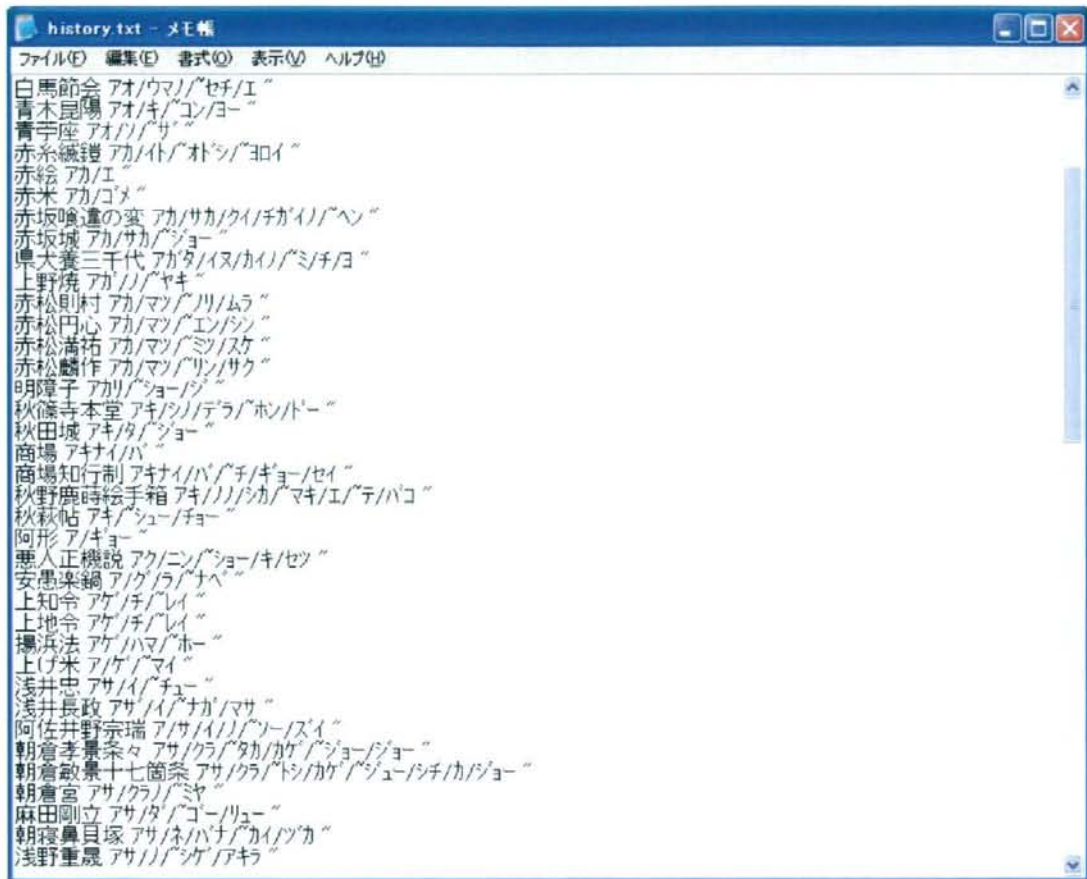
test.exf2 - メモ帳
ファイル(F) 編集(E) 書式(O) 表示(V) ヘルプ(H)
<para id="14" intro="0" lbind="0" lhind="0" pind="0" style="hr"/>
<para id="15" intro="2" lbind="0" lhind="0" pind="0" startnum="2" style="index"/>
</defstyle>
</info>
<header>
<para id="2" page="global">
<k pagenum="right" show="yes">
</k>
<h pagenum="right" show="yes">
</h>
<b pagenum="right" show="yes">
</b>
</para>
</header>
<footer>
<para id="3" page="global">
<k pagenum="right">
</k>
<h pagenum="right">
</h>
<b pagenum="right">
</b>
</para>
</footer>
<body lang="2" startpage="1" tab="8">
<para id="5">
<k src="yes">
</k>
<h>
</h>
<b ekp="000">
</b>
</para>
<para id="6" style="right">
<k src="yes">平成21年4月8日</k>
<h>へいせい21ねん4がつよーか</h>
<b ekp="000">&amp;B]B #BAP0 #D&quot;#N &gt;3*</b>
</para>
<para id="7">
<k src="yes">
</k>
<h>
</h>
<b ekp="000">
</b>
</para>
<para id="8" style="title">
<k src="yes">第49回厚生科学審議会科学技術部会の開催について</k>
<h>だい49かいこーせいかかしくんぎかいかがくきしゅつぶかいかいさいについて</h>
<b ekp="000">&quot;OB#DI*B [3]B *&quot;,*% ¥0&quot;&lt;,*B *&quot;,*% &quot;&lt;,*N&quot;X*BS *B BL
NBQ</b>
</para>

```

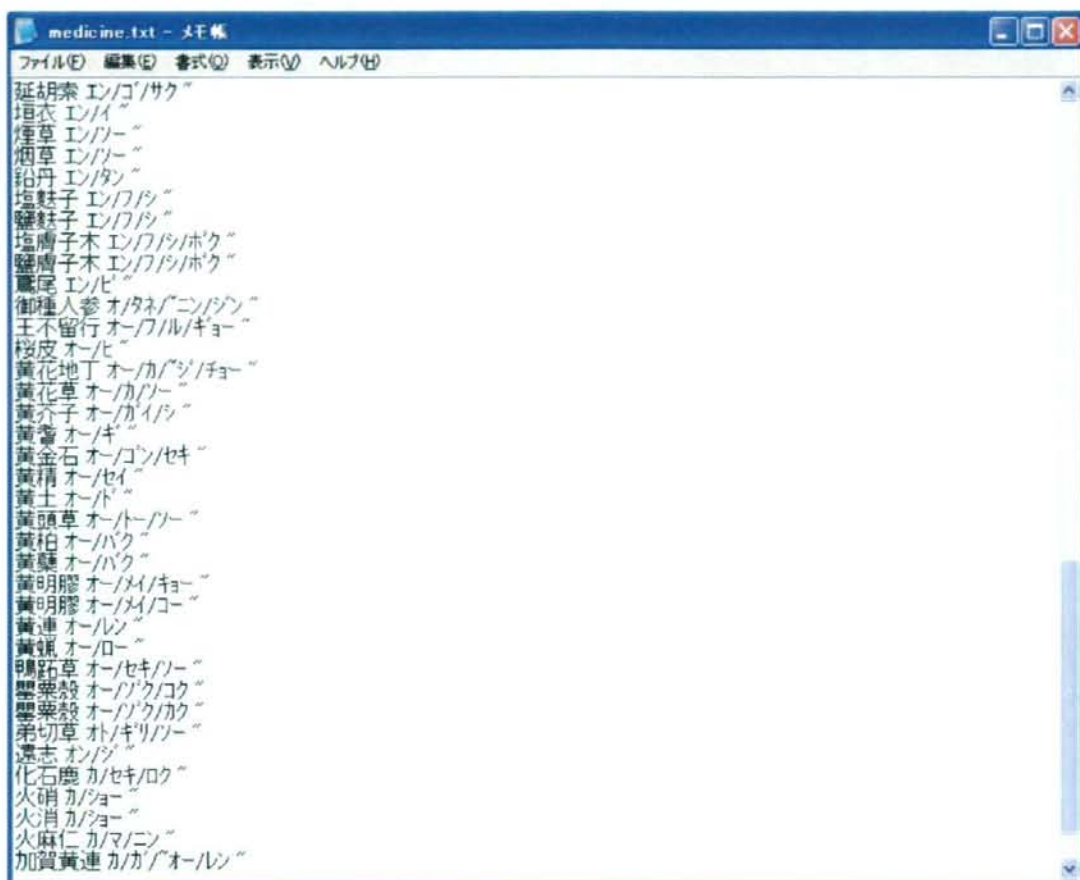
■ 住所辞書



■ 日本史辞書



■ 生薬辞書



■ SBF 仕様書

構造化点訳 XML Structured Braille Format: SBF 仕様

2008年7月14日

作成者：石川准

□ 構造化点訳 XML とは

次世代 Extra 自動点訳エンジンでは、文書ファイルの見出しなどの構造情報を自動点訳に利用する。機能の実現には、構造情報付きの文書情報を入力ファイルとして扱えなければならない。構造情報付きの文書としては、Microsoft Word/Excel/PowerPoint, PDF, RSS, DAISY などの様々なファイルを想定している。

特定の文書ファイル形式に依存した形で直接点訳エンジン内で処理を行う実装を行うと、新たに他の文書ファイル形式に対応するにはまた同様の開発を行わなければならない。特定の文書形式を直接点訳エンジンが扱うのは汎用性の面で開発効率が悪いので、いったん実際の文書ファイルから構造情報付きのテキスト情報を XML ファイルの形式で抽出したのに対して Extra 自動点訳エンジンが点訳を行う構造とする。

本仕様書は、この中間形式の構造情報付き点訳入力 XML(Structured Braille Format: 以降 SBF)の仕様を規定するものである。

□ ファイルフォーマット

SBF ファイルは、「JIS X 4159:拡張可能なマーク付け言語(XML)」で規定される XML の規格に準拠するものとする。

ファイルの拡張子には .sbf を用いる。

□ XML 宣言、文字コード、DTD

ファイルの文字コードは、BOM 付きの UTF-8 とする。ファイルの開始は 16 進数表現で EF BB BF 3C 3F 78 6D となる。

XML 宣言は次のようになる。

```
<?xml version="1.0" encoding="utf-8" ?>
```

DTD 宣言は次のような記載となる。

```
<!DOCTYPE sbf SYSTEM "sbf-1.0.dtd" >
```

□ 要素名、属性値、XML 構成の概要

SBF ファイルのサンプル例は次の通りである。

```
<?xml version="1.0" encoding="utf-8" ?>
<!DOCTYPE sbf SYSTEM "sbf-1.0.dtd" >
<sbf
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:dcq="https://purl.org/dc/terms/" >
  <info>
    <orig_file_name>元文書ファイル名</orig_file_name>
```

```

<filter>sbf ファイルを生成したプログラム名</filter>
...略...
<dc:title>タイトル</dc:title>
<dc:language>ja</dc:language>
<dc:creator>作者名</dc:creator>
<dc:description>要約</dc:description>
...略...
</info>
<header>ヘッダー内容</header>
<body>
  <para id="1" type="h1">第 1 章</para>
  <para id="2">本文</para>
  <para id="3" type="h2">第 1 節</para>
  <para id="4">本文</para>
  <para id="5" type="h3">第 1 項</para>
  <para>本文</para>
  ...略...
</body>
</sbf>

```

このように SBF ファイルの要素名、属性名は全て小文字を用いる。

また、SBF ファイルの要素名、属性値のうちいくつかは Dublin Core を利用する。

基本構造として、sbf 要素(html 要素ではない)が最上位に配置され、この下に info 要素、header 要素、body 要素の順で並ぶ。

info 要素の下には、文書全般に関する情報が格納され、body 要素の下には本文の内容情報が格納される。

要素名	意味
info	文書全体に関する情報
header	ヘッダー情報
body	本文情報

□ info

info 要素の下には、文書全般に関する情報が格納される。info 要素の子ノードとしては、独自の要素と、Dublin Core で規定されるメタデータの 2 種類が配置される。子ノードの出現順序は任意である。

◆ 独自要素

要素名	意味
orig_file_name	変換元の文書ファイル名
filter	sbf ファイルを生成したプログラム名

◆ Dublin Core 要素

Dublin Core で規定される文書のメタデータを記述する。

sbf 内での基本 Dublin Core 要素は dc:で修飾した名前空間を使用する。

また、拡張 Dublin Core 要素は dcq: で修飾した名前空間を使用する。

要素名	意味
dc:title	タイトル、リソースに与えられた名前
dc:creator	リソースの内容に主たる責任を持つ人や組織
dc:subject	リソースの内容に含まれるトピック(キーワード、分類コード)
dc:description	要約や説明文
dc:publisher	リソースを利用可能にする責任者(個人、組織、サービスなど)
dc:contributor	リソースの内容に協力、貢献している人や組織、サービス
dc:date	リソースの作成日、公開日など主要な出来事に関する日
dc:type	分野コード
dc:format	元文書ファイルの MIME Type
dc:identifier	リソースへの曖昧さのない参照(URI, ISBN)
dc:source	リソースが派生作品であるときの元リソースへの参照
dc:language	リソースの言語(ISO 639 の言語コードを用いる)
dc:relation	関連するリソースへの参照
dc:coverage	リソースの内容がカバーする範囲もしくは対象
dc:rights	権利に関する情報
dcq:created	作成日
dcq:issued	正式発行日
dcq:modified	更新日

日付のフォーマットは ISO-8601/W3C-DTF 形式

(<http://www.w3.org/TR/1998/NOTE-datetime-19980827>) を使用する。

年のみ

YYYY (例: 1997)

年月

YYYY-MM (例: 1997-07)

年月日

YYYY-MM-DD (例: 1997-07-16)

年月日+時分+タイムゾーン

YYYY-MM-DDThh:mmTZD (例: 1997-07-16T19:20+01:00)

年月日+時分秒+タイムゾーン

YYYY-MM-DDThh:mm:ssTZD (例: 1997-07-16T19:20:30+01:00)

年月日+時分秒+1/100 秒+タイムゾーン

YYYY-MM-DDThh:mm:ss.sTZD (例: 1997-07-16T19:20:30.45+01:00)

dc:format の値には次に示すような MIME Type を指定する。

拡張子	MIME Type
.doc	application/msword
.xls	application/msexcel
.ppt	application/mspowerpoint
.pdf	application/pdf

.rtf	application/rtf
.docx	application/vnd.openxmlformats-officedocument.wordprocessingml.document
.xlsx	application/vnd.openxmlformats-officedocument.spreadsheetml.sheet
.pptx	application/vnd.openxmlformats-officedocument.presentationml.presentation

dc:language の値には ISO 639 で規定される次のような 2 文字の言語コードを指定する。

コード	言語
en	英語
ja	日本語

dc:language の情報は、点字に変換する際に点訳モードの自動判定に利用される。

dc:type の情報は、点訳を行う際の専門辞書の自動判定に利用される。dc:type の分野コード情報は SBF の変換プログラムに対する引数、オプションなどの形式で指定できることが奨励される。現在計画している分野コードは次の通りである。

分野コード	分野名
medicine	医学
jurisprudence	法学
sociology	社会学

◆ 必須 info 子要素

info 子要素として必須なものは次の通りである。

要素名	意味
orig_file_name	変換元の文書ファイル名
filter	sbf ファイルを生成したプログラム名
dc:title	タイトル、リソースに与えられた名前
dc:language	リソースの言語(ISO 639 の言語コードを用いる)
dc:format	元文書ファイルの MIME Type

これ以外の要素については省略が可能である。

□ body

body 要素の下には、文書内での出現順に段落、表のようなブロック要素が複数個配置される。ブロック要素として利用できるものは次の通りである。

要素名	意味
para	段落
table	表

◆ para 要素

次の例のように para 要素は段落を表現する。

<para id="1" type="h1">詩の書き方</para>

<para id="2" >七・五調などの定型詩は<em type="under">五マス目から書き出す。</para>

<para id="3" >定型詩の表題は書き出し五マス目と区別するため<em type="under">七マス目から書き出す。</para>

para 要素の子要素には段落内容が記述される。段落の内容にはテキストの他、後述するインライン要素が含まれる。

para 要素は必ず id 属性を持つ。id 属性値にはユニークな数字(重複の無い数字)が設定される。

段落には type 属性を付与できる。type 属性の無い段落要素は、通常の本文として扱われる。

type 属性の値として指定できるのは以下の通りである。

属性値	意味・役割
h1	見だし 1
h2	見だし 2
h3	見だし 3
h4	見だし 4
li1	箇条書き 1
li2	箇条書き 2
li3	箇条書き 3
title	タイトル
center	センタリング
right	右寄せ
hr	水平線
index	目次

h1 は一番大きな単位の見だしである。h2 は h1 より一段階小さな単位の見だし、h3 は h2 よりさらに小さな単位の見だし、h4 は h3 よりさらに小さい単位の見だしである。

li1 は箇条書きに用いる。箇条書きの中に、箇条書きを入れ子にするには li2 を用いる。li2 の下にさらに箇条書きを入れ子にするには li3 を用いる。

h1~h4、li1~li3、index の type 属性を持つ para 要素には、次の属性が付与される。

属性	意味・役割
intro	前置文字列
startnum	開始数字番号

startnum 属性で、章節などの番号や、番号付きリストの開始番号を指定できる。

startnum 属性が明示されていない場合には、直前の同じ属性の番号をインクリメントした数値が使用される。

intro 属性値は段落の前置文字列を意味する。前置文字列中の数字は、各レベルの番号に置き換わる。

番号付き箇条書きの例:

-
- 1 大会概要:東京マラソン 2009
 - 2 主催:(財)日本陸上競技連盟、東京都
 - 3 開催日時:2009年3月22日(日)
 - 4 制限時間:マラソン:7時間、10km:1時間40分
-

上記のような番号付きの箇条書きを para 要素で表現すると、次のようになる。

```
<para id="10" type="li1" intro="1 " startnum="1">大会概要:東京マラソン 2009</para>
<para id="11" type="li1" intro="1 ">主催:(財)日本陸上競技連盟、東京都</para>
<para id="12" type="li1" intro="1 ">開催日時:2009年3月22日(日)</para>
<para id="13" type="li1" intro="1 ">制限時間:マラソン:7時間、10km:1時間40分</para>
```

番号無し箇条書きの例:

- ・大会概要:東京マラソン 2009
- ・主催:(財)日本陸上競技連盟、東京都
- ・開催日時:2009年3月22日(日)
- ・制限時間:マラソン:7時間、10km:1時間40分

上記のような番号無しの箇条書きを para 要素で表現すると、次のようになる。

```
<para id="10" type="li1" intro="・ ">大会概要:東京マラソン 2009</para>
<para id="11" type="li1" intro="・ ">主催:(財)日本陸上競技連盟、東京都</para>
<para id="12" type="li1" intro="・ ">開催日時:2009年3月22日(日)</para>
<para id="13" type="li1" intro="・ ">制限時間:マラソン:7時間、10km:1時間40分</para>
```

◇ 目次

type 属性値に index が指定される目次に、どの段落に対する目次なのかという情報が重要である。参照先の段落を示すために、次のような idref の属性を使用する。

属性	意味・役割
idref	段落番号

idref の属性値には、参照先段落の id 値を指定する。

例:

```
<para id="10" type="index" idref="20">第一章：都市の生活</para>
<para id="11" type="index" idref="34">早朝の風景</para>
<para id="12" type="index" idref="36">昼間の風景</para>
<para id="13" type="index" idref="50">第二章：町中探検</para>
<para id="14" type="index" idref="55">銭湯めぐり</para>
<para id="15" type="index" idref="62">公園めぐり</para>
```

◆ table 要素

表を記述するには table 要素を使用する。

table 要素の構成は、XHTML の table にほぼ準拠したものになる。

```

<table id="58">
<tr><th>セル見だし 1</th><th>セル見だし</th></tr>
<tr><td>セル内容</td><td>セル内容</td></tr>
<tr><td>セル内容</td><td>セル内容</td></tr>
<tr><td>セル内容</td><td>セル内容</td></tr>
</table>

```

table 要素に対して id 属性が必ず付与される点が XHTML とは異なる点である。
id 属性値は、para 要素と同様にユニークな数字が設定される。この id 属性値は、para 要素の id 属性値とは重複しない。

テーブルで使われる要素は次の通りである。

要素	意味・役割
tr	テーブルの行
th	テーブルのセル見だし
td	テーブルのセルデータ

◆ インライン要素

元文書の段落中に何らかの属性情報が付いている場合には点訳時にこの情報を生かせる可能性がある。このような、段落中の属性情報をインライン要素と呼ぶ。
インライン要素には次のような要素がある。

要素	意味・役割
em	段落のデフォルト属性とは異なる属性が付いている
a	ハイパーリンク
ruby	ルビ
page	ページ区切り

◇ em 要素

元文書の段落中で、特定の箇所だけ特殊な属性が付いている場合には該当箇所を em 要素の子要素として表現する。

なお、段落全体に同じ属性が付与されている場合は、構造情報としての意味合いが薄いので、em 要素を用いての属性を表現は行わない。

em 要素には、必ず type 属性が指定され、元文書のどのような表現であったかを示す。

type 属性値	意味・役割
italic	イタリック
bold	ボールド
under	アンダーライン
color	文字色/背景色の変更
font	フォント・サイズの変更
other	その他の属性変更（網掛け、打ち消し線など）

例：

```
<para id="201"><em type="under">マラソン:</em>東京都庁～飯田橋～皇居前～日比谷～品川～銀座～日本橋～浅草雷門～築地～豊洲～東京ビッグサイト(日本陸上競技連盟/AIMS 公認コース) </para>
```

```
<para id="202"><em type="under">10km:</em>東京都庁～飯田橋～皇居前～日比谷公園(公認条件に適合せず記録は公認されない)</para>
```

この例では、段落頭の「マラソン」と「10Km」の2箇所がアンダーラインの属性が付与されていることを示す。

◇ a 要素

ハイパーリンクを示すのに a 要素を用いる。

ハイパーリンクの例:

```
<para id="133">皇居マラソン参加のお申し込みは、インターネットの<a href="http://www.foo.bar/">登録サイト</a>よりお願いします。</para>
```

◇ ruby 要素

ruby 要素はルビを表現するのに使用する。

ruby 要素の下にルビの対象範囲を示す rb 要素と、ルビ文字列を示す rt 要素を用いて表現する。

(XHTML ではルビのタグを解釈できないブラウザ用に rp 要素を用いるが、SBF では使用しない。)

ルビの例:

```
<para id="203"><ruby><rb>米国</rb><rt>アメリカ</rt></ruby>は映画産業が盛んである。</para>
```

◇ page 要素

点訳では必要に応じて原文のページ番号を付加することがある。これを実現するため、新しいページが始まったことを page 要素を使って示す。

page 要素には、ページ番号を示す pagenum 属性が必須である。

```
<para id="433">組織委員会を行い、2009 年の<page pagenum="10" />大会要項が決定しましたのでお知らせします。</para>
```

□ header 要素

header 要素には元文書のヘッダーの内容を記載する。

例:

```
<header>マラソン大会参加概要</header>
```

header 要素の子要素にはインライン要素を含められる。

□ Microsoft Word の Open XML ファイル(.docx)からの情報抽出

Microsoft Word の Open XML ファイル(.docx)からの情報抽出に関する注意事項は次の通りである。

◆ 修正記録

修正記録付きの文書ファイルの場合には最後の確定内容のみが抽出され、過去に削除された文字列情報、スタイル情報などは抽出しないものとする。

◆ 見だしレベル・箇条書きレベル

Microsoft Word の見だしレベルは 1~9 までが定義されている。SBF に見だしレベルの情報を変換するにあたっては、1~4 のみを対象とし、これより大きな見だしレベル 5~9 は通常の本文と同様として扱う。

また、箇条書きレベルも Microsoft Word では 1~5 までが定義されている。こちらも 4~5 の箇条書きレベルは本文と同様として扱う。

◆ プロパティ情報

Word の文書プロパティ情報は、次のように Dublin Core 要素に対応させる。

要素名	Word の文書プロパティ情報
dc:title	ファイル概要-タイトル
dc:creator	ファイル概要-作成者
dc:subject	ファイル概要-分類
dc:description	ファイル概要-サブタイトル
dc:publisher	ファイル概要-管理者
dc:contributor	
dc:date	ファイル概要-更新日時
dc:type	(変換プログラム側で指定)
dc:format	application/vnd.openxmlformats-officedocument.wordprocessingml.document
dc:identifier	
dc:source	
dc:language	(ja もしくは en を指定)
dc:relation	ユーザ設定-リファレンス
dc:coverage	
dc:rights	
dcq:created	ファイル概要-作成日時
dcq:issued	ユーザ設定-完了日
dcq:modified	ファイル概要-更新日時

◆ セクション区切り

Open XML ではセクション区切り情報を用いて、文書レイアウトを変更している。

Word の SBF 化にあたっては、セクション区切りはなにも情報を抽出しない。

◆ カスタム XML マークアップ

Open XML ではカスタム XML マークアップというしくみで、ユーザ定義の XML 情報を Open XML 中に埋め込める。

Word の SBF 化にあたっては、カスタム XML マークアップに関してはなにも情報を抽出しない。

■ SBF の DTD (document type definition) ver.1.0

<!-- a Element -->

<!ELEMENT a (#PCDATA | em | a | ruby | page | blockquote | footnote | annotation)*>

<!ATTLIST a href CDATA #REQUIRED>

<!-- ruby Element -->

<!ELEMENT ruby (rb, rt)>

<!ELEMENT rb (#PCDATA | em | a | ruby | page | blockquote | footnote | annotation)*>

<!ELEMENT rt (#PCDATA | em | a | ruby | page | blockquote | footnote | annotation)*>

<!-- page Element -->

<!ELEMENT page (#PCDATA | em | a | ruby | page | blockquote | footnote | annotation)*>

<!ATTLIST page pagenum CDATA #REQUIRED>

<!-- blockquote Element -->

<!ELEMENT blockquote (#PCDATA | em | a | ruby | page | blockquote | footnote | annotation)*>

<!-- footnote Element -->

<!ELEMENT footnote (#PCDATA | em | a | ruby | page | blockquote | footnote | annotation)*>

<!ATTLIST footnote notenum CDATA #REQUIRED>

<!-- annotation Element -->

<!ELEMENT annotation (#PCDATA | em | a | ruby | page | blockquote | footnote | annotation)*>

<!ATTLIST annotation notenum CDATA #REQUIRED>

■ 形態素解析の例

<sentence><phrase><word type="名詞">当時</word><word type="名詞">私</word><word type="助詞">は</word><word type="名詞">図書館</word><word type="カタカナ名詞">サービス</word><word type="助詞">と</word><word type="助詞">は</word><word type="副詞">全く</word><word type="名詞">関係</word><word type="助詞">の</word><word type="形容詞">ない</word><word type="名詞">仕事</word><word type="助詞">を</word><word type="サ変動詞連用形">し</word><word type="助動詞">て</word><word type="1 段動詞語幹">い</word><word type="助動詞">た</word><word type="助詞">ので</word><word type="読点">、</word></phrase><phrase><word type="連体詞">その</word><word type="名詞">方</word><word type="助詞">の</word><word type="名詞">訴え</word><word type="助詞">を</word><word type="5 段動詞連用形">聞き</word><word type="助動詞">ながら</word><word type="読点">、</word></phrase><phrase><word type="開きかっこ">「</word><word type="市町村名">日野市</word><word type="助詞">の</word><word type="名詞">図書館</word><word type="助詞">にも</word><word type="名詞">点字</word><word type="助詞">の</word><word type="1 段動詞連体形">できる</word><word type="名詞">視覚</word><word type="名詞">障害</word><word type="名詞">職員</word><word type="助詞">が</word><word type="1 段動詞語幹">い</word><word type="助動詞">て</word><word type="読点">、</word></phrase><phrase><word type="名詞">点字</word><word type="名詞">指導</word><word type="助詞">を</word><word type="1 段動詞連用形">やっ</word><word type="助動詞">て</word><word type="1 段動詞語幹">くれ</word><word type="助動詞">たら</word><word type="形容詞">いい</word><word type="助詞">のに</word><word type="閉じかっこ">」</word><word type="助詞">と</word><word type="5 段動詞未然">思わ</word><word type="助動詞">ず</word><word type="助詞">には</word><word type="1 段動詞未然形">いられ</word><word type="助動詞">ません</word><word type="助動詞">でした</word><word type="句点">。</word></phrase></sentence>