XML を利用した『日本語話し言葉コーパス』の検証と検索

菊池　英明

早稲田大学人間科学部／国立国語研究所

〒359-1192 埼玉県所沢市三ヶ島 2-579-15

E-mail: kikuchi@waseda.jp

あらまし　『日本語話し言葉コーパス』(Corpus of Spontaneous Japanese;以下 CSJ と呼ぶ)には、転記、形態論情報、分節音情報、韻律情報、係り受け情報など様々な研究用の情報が含まれる。同一のデータに付加される、種類の異なる情報の間に依存関係がある場合、情報の作成過程において情報間に矛盾が生じることがある。コーパス全体において整合性の高い研究用情報を提供するために、多様な構造の記述に向きかつ構造の妥当性を検証する仕組みを持つ XML を利用する。具体的には、全ての CSJ 研究用情報を統合して XML によって記述し、XML の関連規格である XML Schema や XSLT を用いて型や構造の検証を行う。本稿では CSJ の各種情報を XML で記述する方法を述べたうえで、XML を利用して行う CSJ の整合性検証方法を説明する。また XML によって各種情報間の依存関係を構造として表現することにより可能になる、他階層に跨る情報検索についても述べる。

キーワード　日本語話し言葉コーパス、整合性検証、検索、XML

Verification and Retrieval of the *Corpus of Spontaneous Japanese* by XML Technologies

KIKUCHI Hideaki

School of Human Sciences, Waseda University / National Institute for Japanese Language

2-579-15, Mikashima, Tokorozawa-shi, Saitama, 359-1192 Japan

E-mail: kikuchi@waseda.jp

Abstract　The '*Corpus of Spontaneous Japanese (CSJ)*' will contain various data such as transcription, POS information, segmental labels, intonation labels, and so on. There are dependencies between those data and it is important to keep consistency of them high. This paper discusses the use of XML technologies for verification of data consistency and retrieval of data in the *CSJ*. Basic principle of XML design for the *CSJ* and some problems in designing will be described. Also, a method of verification and retrieval of the *CSJ* will be described with some examples using relevant standards such as XML Schema, XSLT and XPath.

Keyword　Corpus of Spontaneous Japanese, consistency, retrieval, XML

1. はじめに

『日本語話し言葉コーパス』(Corpus of Spontaneous Japanese;以下 CSJ と呼ぶ)には、転記、形態論情報、分節音情報、韻律情報、係り受け情報など様々な研究用の情報が含まれる[1]。これらの情報が同一のデータに対して付加された場合、種類の異なる情報の間で依存関係を持つことがある。例えば、形態論情報や係り受け情報は転記のテキストに対して付加される情報であるし、韻律情報におけるアクセント位置などは形態論情報から得られる語単位に付加する情報である。したがって、それぞれの情報付加作業の出発点で、依存関係のある他の種類の情報を利用することにより、各種情報間に矛盾のない、整合性の高いデータの構築が可能になる。

ところが、情報の作成過程においては、それぞれの種類の情報について独立に作業を進めるために、各種情報間に矛盾が生じてくる。常に各種の情報そのものあるいは情報付加作業者の間で連携をとれば発生した矛盾を即座に解消することも可能だが（伝[2]はこの種の問題の所在を明確にし、これを解決する一つのアイデアを提案している。)、CSJ の開発においてはそれぞれの情報付加の基準を定めながら作業を進めざるを得ないため、そのような方法は得策とはいえない。そこで我々は、各種の情報付加作業は専門のグループ毎に行い、それらが終了した段階で各種情報間の整合性を検証して矛盾のある箇所を修正するアプローチを取

ることにした。

各種情報の表現および整合性検証には、XML と呼ばれるマークアップ言語を用いる。XML は、構造化されたデータをテキスト形式で記述することのできる柔軟性の高い記述言語であり、多様な構造を持った情報を記述するのに向き、かつデータ構造の妥当性を検証する仕組みを持つ。そこで、CSJ の研究用情報を XML で記述し、各種情報間の整合性を構造の妥当性として検証する。

ところで、XML を用いれば各種情報間の依存関係が構造として表現される。例えば、先に記した転記のテキストと形態論情報の関係は、転記における発話単位の下位階層に形態論情報における語（CSJ では短単位）を位置させ、語の属性情報として転記のテキストを持つ形で表現できる。あるいは、韻律情報と形態論情報の関係は、韻律情報のうち語境界に付加する情報を形態論情報における語の属性値として持つ形で表現できる。このように構造化して表現することにより、例えばある特定の語におけるアクセント位置の分布を調べるなど、種類の異なる情報に跨った情報検索も容易になる。

本稿では、まず XML についての簡単な解説をしたうえで CSJ の各種情報を XML で記述する方法を述べる。さらに、XML を利用して行う CSJ の整合性検証と検索について説明する。

2. XML

2.1. XML とは

XML(Extensible Markup Language)は、SGML (Standard Generalized Markup Language)を祖とした記述言語であり、WWW で用いられる技術の標準化を行う団体 W3C(World Wide Web Consortium)によって 1998 年に勧告として発表された。データを構造化して記述できる利点により、HTML に代わる次世代の WWW コンテンツ記述言語として注目されてきたが、さらにここに来て様々な関連規格が整備され、単に WWW コンテンツ記述言語としてのみならず、汎用的なデータ記述言語としてその活躍の場を大きく広げている。最近では言語コーパスの記述言語としても使われるようになっている[3][4][5]。

2.2. XML の基本

XML で記述する基本単位は、開始タグと終了タグで内容を囲んだものであり、これを要素という。例えば”*<author>*菊池*</author>*”のように記述すれば、「菊池」という内容を持つ author 要素を記したことになる。また、このようにしてタグ名を設定することで任意の要素名を自由に導入することができる。

各要素には属性とその値を表現することができる。例えば先の author 要素に値が 1 である属性 id を持たせるには、”*<author id="1">*菊池*</author>*と記述する。

さらに、タグで囲まれる範囲に別の要素を記すことにより、木構造を表現でき親子関係や兄弟関係を表すことができる。例えば、author 要素の子供にあたる affiliation 要素を記す場合、以下のように記述できる。

```
<author id="1">菊池英明
  <affiliation id="1">早稲田大学</affiliation>
  <affiliation id="2">国立国語研究所</affiliation>
</author>
```

このように、XML は階層構造の表現が容易であり、基本的には階層構造で表現できる言語学的情報の記述に向いている。次節には、CSJ の研究用情報を XML で記述する方法とその際の問題について説明する。

3. XML による CSJ 各種情報の記述

3.1. 基本方針

まず始めに、前述の目的を実現するために XML で記述すべき研究用情報を以下に列挙する。なお、これ以外にも CSJ には多様な研究用情報が含まれ、CSJ 公開時にはそれらの情報も XML の形で記述して用意する予定だが、本稿の範囲では下記の情報に限定して説明する。

- 転記（転記基本単位とその時間位置、基本形表記、発音形表記、各種タグ情報等）
- 形態論情報（短単位、長単位、代表形、品詞、活用の種類等）
- 分節音情報（分節音ラベルとその時間位置）(*1)
- 韻律情報（トーンラベル、BI ラベルとその時間位置、単語アクセント位置等）(*1)

(＊1 これらはコアと呼ばれる約 200 講演にのみ与えられる)

これらの情報には、言語学的情報と音声学的情報が混在しており、例えば語の境界をはさんで音声的に融合しているために分節音ラベルが語境界をまたぐような場合など、階層構造を逸脱するものもある。しかしながら、整合性検証や検索などの多くの用途においては言語学的な階層構造を想定することが予想されるため、CSJ の研究用情報を記述する際にはあくまで階層構造を基本とすることにする。

CSJ における XML の要素について、関連図を図 1 に、各要素の内容を表 1 に示す。なお、紙面の都合上、一部の要素、属性については省略している。

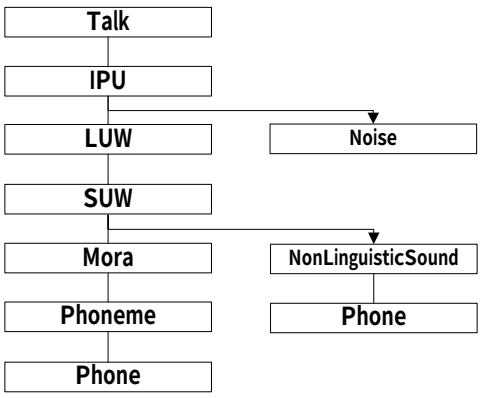

図 1: XML 要素関連図

表 1: XML 要素の内容

要素	属性	説明
Talk		講演
	RecordingDate	録音日
	SpeakerID	話者 ID
	BirthDate	誕生日
	WaveFilePath	音声ファイルパス
IPU		転記基本単位
	Channel	音声録音チャンネル
	IPUStartTime	開始時刻
	IPUEndTime	終了時刻
LUW		長単位
	LUWPOS	品詞 (長単位)
	LUWConjugateType	活用型 (長単位)
	LUWConjugateForm	活用形 (長単位)
	LUWDictionaryForm	代表形 (長単位)
	LUWLemma	代表表記 (長単位)
SUW		短単位
	SUWDictionaryForm	代表形 (短単位)
	SUWLemma	代表表記 (短単位)
	SUWPhoneTrans	発音形 (短単位)
	SUWPOS	品詞 (短単位)
	SUWConjugateType	活用の種類 (短単位)
	SUWConjugateForm	活用形 (短単位)
	LexicalAccPos	語彙アクセント位置
	TagDisfluency	語断片 (D)
	TagFiller	フィラー (F)
	TagIncorrect	言い誤り (W)
	TagIncorrectNorm	言い誤り正式発音
	TagForeign	外国語 (O)
Mora		
	MoraEntity	モーラ記号
	Uncertain	発音不明瞭 (?)
	Whisper	ささやき声 (L)
	PerceivedAccPos	知覚アクセント位置
Phoneme		音素
	PhonemeEntity	音素記号
Phone		分節音
	PhoneEntity	分節音記号
	Devoiced	無声化の有無
	PhoneStartTime	開始時刻
	PhoneEndTime	終了時刻
	StartPosUncertain	開始位置不明
	EndPosUncertain	終了位置不明
Noise		言語音と独立した非言語音
NonLinguisticSound		言語音と共起する非言語音

図 1 に示すように、講演を表す Talk 要素から分節音を表す Phone 要素までを階層構造で表現する。各要素について以下に説明する。

(1) Talk 要素

CSJ では主に講演を単位として収録した音声をファイル化している。Talk 要素はこの単位についての情報を記述するために、録音日や音声ファイルのパスなどを記述するための属性を持つ。また、講演の話者についての情報も記録する。なお、二人の話者が存在するインタビューについては、話し手（インタビュイー）の話者情報を記す。Talk 要素は複数の IPU 要素を子要素として持つ。

(2) IPU 要素

転記において、原則として 200[ms]以上のポーズで区切られた単位を転記基本単位として認定する。この単位を IPU(Inter-Pausal Unit)要素とし、講演における各単位の開始時刻、終了時刻を属性として記録する。なお、二人の話者が存在するインタビューにおいては、2 チャンネルで格納された音声ファイル中のチャンネル名も属性として記す。IPU 要素は一つ以上の LUW 要素を子要素として持つ。

(3) LUW 要素

形態論情報のうちの長単位を LUW(Long-Unit Word)要素とし、長単位の品詞、活用型などの情報を属性として持つ。LUW 要素は一つ以上の SUW 要素を子要素として持つ。

(4) SUW 要素

形態論情報のうちの短単位を SUW(Short-Unit Word)要素とする。この要素は、いわば言語学的情報と音声学的情報の接点に相当する。短単位の品詞、活用の種類などの言語学的情報はこの要素の属性として記録される。そして、対応する音声における発音は子要素である Mora 要素の属性として記録される。音声に関する情報のうち、例えば言い誤りの有無を示すタグなど、語単位で与えられるものは SUW 要素の属性として記録する。また、語彙的に決定されるアクセント位置をアクセント辞書から求めて属性として記す予定である。SUW 要素は一つ以上の Mora 要素あるいは NonLinguisticSound 要素を子要素として持つ。

(5) Mora 要素

(4)で述べたように、転記の発音形に記される発音表記をモーラ単位に分割した単位を Mora 要素とし、モーラ記号（カナ）を属性として記す。なお、転記におけるタグのうち、モーラ単位で記されるものはこの要素の属性として記録する。さらに、韻律ラベルにおいて記録された知覚アクセントの位置もこの要素の属性として記録する。Mora 要素は一つ以上の Phoneme 要素を子要素として持つ。

(6) Phoneme 要素

Mora 要素の属性 MoraEntity の値から自動的に音素単位を生成して Phoneme 要素とし、音素記号を属性として記す。Phoneme 要素は一つ以上の Phone 要素を子要素として持つ。

(7) Phone 要素

分節音ラベルの単位を Phone 要素として、ラベルやラベル区間の開始・終了時刻などを属性として記す。また、分節音ラベルに記された母音無声化の情報も属性として記録する。Phone 要素は韻律情報の各ラベル情報を要素としたものを子要素として持つ。その際、ラベルは必ず時間情報を持つため、ラベルの時間を区間に含む Phone 要素に機械的に所属させることにしている。ただし、そのように物理的な時間関係だけを元に構成するのでは、後に述べる検索用途において韻律情報に関わる検索が困難になる恐れがある。したがって、検索用途のために、ラベルの持つ音韻論的な意味を解釈した結果を反映させた XML を派生させる。

(8) Noise 要素

転記の発音形に記される、言語音とは独立に発生した息や咳などの情報を記すために、Noise 要素を設ける。Noise 要素は子要素に NonLinguistic 要素を持つ。

(9) NonLinguisticSound 要素

転記の発音形に記される、言語音区間中に発生した息や咳などの情報を記すために、NonLinguisticSound 要素を設ける。NonLinguisticSound 要素は子要素として Phone 要素を持つ場合がある。

なお、CSJ のうち、分節音情報や韻律情報が作成されるのはコアと呼ばれる約 50 万語相当のデータに限定される。したがって、上記の要素のうち(6),(7)と他の要素の一部の属性については一部のデータでのみ用いられる。

3.2. 階層構造からの逸脱に対する措置

話し言葉音声においては、言語学的な階層構造を逸脱する現象がたびたび生じる。ここでは、幾つかのそうした現象に対する措置を説明する。

3.2.1. 分節音ラベルの複合化

分節音ラベル作成において､母音の無声化､子音の弱化､母音連鎖や融合などによって分節音境界を定めにくい場合に、複数のラベルを一つに複合化して付与する処置を許している。これにより、ラベルの区間が明確でない分節音の単位が生じることになる。

一方、前述した階層構造から、分節音ラベルの情報を記録する Phone 要素の時間情報を参照すれば、上位階層にあたる Mora 要素や SUW 要素の、音声における開始・終了時間がわかる。したがって、大まかでも Phone 要素の時間情報を記して欲しいという要求が想定される。そこで、複合化された分節音ラベルについては、便宜上区間を均等に分割した結果の区間開始・終了時刻を各 Phone 要素の属性として記し、恣意的に与えた境界であることを別途属性で記録することにした。例えば、母音の無声化によって複合化された分節音ラベルを以下のように記述する。

分節音ラベル（左からラベル区間の終端時間、ラベル）

```
244.647519  a
244.785809  s,U        #/u/の無声化により複合
```

XML による分節音ラベルの複合化の記述

```
<Phone PhoneStartTime="244.564289" PhoneEndTime=
"244.647519“ PhoneEntity="a"></Phone>
<Phone PhoneStartTime="244.647519" PhoneEndTime=
"244.716664“ PhoneEntity="s" EndPosUncertain="1"></Phone>
<Phone PhoneStartTime="244.716664" PhoneEndTime=
"244.785809“ PhoneEntity="u" StartPosUncertain="1"
Devoiced="1"></Phone>
```

3.2.2. 語の発音レベルでの融合

話者の言い誤りによって、複数の語が発音レベルで融合しているように観察されるケースが頻繁に見られる。例えば、「この」と「よう（な）」の発音が融合して「コニョー(ナ)」となるようなケースである。このようなケースに対して、CSJ の転記では発音形テキストにおいて、関係する語全体に(W)タグを記し「(W コニョー;コノヨウ)のように“正しいであろう発音”とともに実際の発音を表記する。一方、形態論情報は転記の基本形テキストに対して付加されるため語の融合としては扱わない。この矛盾に XML の構造上で対応するために、形態論情報は SUW 要素の属性として表わし、モーラ以下の音声学的情報は融合している範囲の先頭の SUW 要素の下位階層に記録する。残りの SUW 要素の下位階層には MoraEntity の値が”φ”である Mora 要素を置く。上記の例は XML により以下のように記述される。

XML による語の融合の記述

```
<SUW SUWDictionaryForm="コノ" SUWLemma="此の"
SUWPOS="連体詞" TagIncorrectNorm="コノ">
  <Mora MoraEntity="コ"></Mora>
  <Mora MoraEntity="ニョ"></Mora>
  <Mora MoraEntity="ー"></Mora>
</SUW>
<SUW SUWDictionaryForm="ヨウ" SUWLemma="様"
SUWPOS="形状詞" TagIncorrectNorm="ヨウ">
  <Mora MoraEntity="φ"></Mora>
</SUW>
```

3.3. CSJ 研究用情報の XML による記述例

これまでに示した設計方針にしたがって、CSJ の研究用情報を XML で記述する例を示す。まず、同一音声に対して作成された転記、形態論情報、分節音情報、韻律情報のそれぞれを図 2 に示す。

これらの情報の XML 化にあたっては、まず始めに短単位情報を XML の形に変換する。なお、転記の情報は余すことなく短単位情報に記されているため、転記との照合は不要である。この段階で、**3.1**で示した Phoneme 要素までが記述された XML が作成される。次の段階では分節音情報から作成した Phone 要素を Phoneme 要素の子要素として配置する。さらに韻律情報から、知覚されたアクセントの位置の情報を Mora 要素の属性として記録し、トーンラベルの情報を Phone 要素の子要素として位置付けるなどの処理を行う。

以上の処理によって自動的に生成される XML の例を図 3 に示す。

4. XML を利用した検証

各種の情報付加作業の出発点では、依存関係を持つ別の情報を利用するため、各種情報間の矛盾はない。しかし、情報の作成過程においては、それぞれの種類毎に独立に作業を進めるため、各種情報間に矛盾が生じることがある。この矛盾を解消するために行う検証に、要素や属性間の関係の妥当性を検証する仕組みを備え持つ XML を利用する。具体的には、XML Schema および XSLT の規格を用いる。

転記

```
0091 00244.050-00245.009 L:
いつもの                    & イツモノ
場所で                      & バショデ
0092 00245.270-00245.581 L:
(D ねろ)                    & (D ネロ)
0093 00245.800-00247.076 L:
寝転がっていますと          & ネ<Q>コロガッテイマスト
```

形態論情報（「いつもの」の部分に対して）

```
S03f0119 0091 00244.050-00245.009 L:-001-001          いつ
         いつ         イツ         何時         イツ         代名詞
                                   2001-05-14 12:01:18+09
S03f0119 0091 00244.050-00245.009 L:-001-005          も
         も           モ           も           モ           助詞
                      副助詞       2001-04-09 10:59:11+09
S03f0119 0091 00244.050-00245.009 L:-001-007          の
         の           ノ           の           ノ           助詞
                      格助詞       2001-04-09 10:59:11+09
```

分節音情報（「いつもの」の部分に対して）

```
  244.073871  121 #
  244.154540  121 i
  244.187874  121 <cl>
  244.240331  121 c
  244.268501  121 u
  244.328683  121 m
  244.372218  121 o
  244.418315  121 n
  244.493862  121 o
```

韻律情報（「いつもの」の部分に対して）

```
  244.092331  115 %L
  244.114585  115 A
  244.494613  115 L%
```

図 2: CSJ の研究用情報の例

4.1. XML Schema を用いた型や構造の検証

XML Schema とは、XML データの記述内容を規定するスキーマ言語の一つであり、XML 同様、W3C によって規格化されている。XML Schema では、要素や属性の名前と型を対応付けたり、ある要素について下位要素の出現順序や出現回数を規定したりすることができる。例えば、Talk 要素のスキーマ定義は以下のように記述できる。

XML Schema による Talk 要素のスキーマ定義

```
<xs:element name="Talk">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="IPU" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="RecordingDate" type="xs:date"/>
    <xs:attribute name="SpeakerID" type="xs:string"/>
    <xs:attribute name="BirthDate" type="xs:date"/>
    <xs:attribute name="WaveFilePath" type="xs:string"/>
  </xs:complexType>
</xs:element>
```

この場合、Talk 要素は IPU 要素を子要素として持ち、その数は無限であることを規定し、さらに属性として date 型の RecordingDate、string 型の SpeakerID、date 型の BirthDate、string 型の WaveFilePath を持つことを規定している。

```
<Talk RecordingDate="2000-01-01" SpeakerID="0001"
BirthDate="1950-01-01" WaveFilePath="wav/S03f0119.wav">
  <IPU Channel="L" IPUStartTime="244.050"
  IPUEndTime="245.009">
    <LUW LUWPOS="代名詞" LUWDictionaryForm="イツ"
    LUWLemma="何時">
      <SUW SUWPOS="代名詞" SUWDictionaryForm="イツ"
      SUWLemma="何時" LexicalAccPos="1" >
        <Mora MoraEntity="イ" PerceivedAccPos="1">
          <Phoneme PhonemeEntity="i">
            <Phone PhoneEntity="i" PhoneStartTime="244.073871"
            PhoneEndTime="244.154540">
              <XJToBITone XJToBIToneEntity="%L"
              LabelTimePos="244.092331"/>
              <XJToBITone XJToBIToneEntity="A"
              LabelTimePos="244.114585"/>
            </Phone>
          </Phoneme>
        </Mora>
        <Mora MoraEntity="ツ">
          <Phoneme PhonemeEntity="c">
            <Phone PhoneEntity="cl" PhoneStartTime="244. 154540"
            PhoneEndTime="244.187874"/>
            <Phone PhoneEntity="c" PhoneStartTime="244. 187874"
             PhoneEndTime="244.240331"/>
          </Phoneme>
          <Phoneme PhonemeEntity="u">
            <Phone PhoneEntity="u" PhoneStartTime="244.
            244.240331" PhoneEndTime="244.268501"/>
          </Phoneme>
        </Mora>
      </SUW>
    </LUW>
    <LUW LUWPOS="助詞" LUWDictionaryForm="モ"
    LUWLemma="も">
      <SUW SUWPOS="助詞" SUWDictionaryForm="モ"
           SUWLemma="も">
        <Mora MoraEntity="モ">
          <Phoneme PhonemeEntity="m">
            <Phone PhoneEntity="m" PhoneStartTime=" 244.268501"
            PhoneEndTime="244.328683"/>
          </Phoneme>
          <Phoneme PhonemeEntity="o">
            <Phone PhoneEntity="o" PhoneStartTime="244. 328683"
            PhoneEndTime="244.372218"/>
          </Phoneme>
        </Mora>
      </SUW>
    </LUW>
    <LUW LUWPOS="助詞" LUWDictionaryForm="ノ" LUWLemma="
    の">
      <SUW SUWPOS="助詞" SUWDictionaryForm="ノ"
      SUWLemma="の">
        <Mora MoraEntity="ノ">
          <Phoneme PhonemeEntity="n">
            <Phone PhoneEntity="n" PhoneStartTime=" 244.372218"
            PhoneEndTime="244.418315"/>
          </Phoneme>
          <Phoneme PhonemeEntity="o">
            <Phone PhoneEntity="o" PhoneStartTime="244. 418315"
            PhoneEndTime="244.943113">
              <XJToBITone XJToBIToneEntity="L%"
              LabelTimePos="244.494613"/>
            </Phone>
          </Phoneme>
        </Mora>
      </SUW>
    </LUW>
  </IPU>
</Talk>
```

図 3: XML 化した CSJ の研究用情報の例

このようにして XML Schema を用いれば、各要素についてのスキーマ定義を記述でき、スキーマのプロセッサによって XML における型や構造の検証ができる。また定義したスキーマ自体は各種情報の構造や型を標準規格によって定義することにもなるため、複雑な情報群の共有に役立つ。

4.2. XSLT を用いた整合性検証

XSLT(XSL Transformation)とは、XML の構造を変換するための言語に関する規格であり、W3C によって規格化されている。XSLT はデータの変換や加工、検索、抽出に利用できるため、この規格を用いれば複雑な構造を持った XML から対象とする部分だけを処理に都合の良い形で抽出することができる。これを利用して、CSJ の研究用情報を記述した XML における各種情報間の整合性を検証する。

CSJ の研究用情報において各種情報間に不整合が生じるケースとしては主に以下のようなものがあげられる。

(A) 転記発音形テキストと分節音ラベル、韻律ラベルの不整合
(B) 形態論情報短単位境界と韻律ラベルの不整合
(C) 転記タグと韻律ラベルの不整合

このうち、(A)について、具体的に転記発音形の発音表記「ノ」に対して分節音ラベル”n”,”o”の対応を検出するために記述した XSL(Extensible Stylesheet Language)を以下に示す。

XSLT による転記発音形と分節音ラベルの整合性検証例

```
<xsl:template match="/">
  <xsl:apply-templates select="//Mora[@MoraEntity='ノ']"/>
</xsl:template>
<xsl:template match="Mora">
  <xsl:apply-templates select="./Phoneme/Phone" />
</xsl:template>
<xsl:template match="Phone">
  <xsl:choose>
    <xsl:when test="@PhoneEntity='n'">
      <xsl:variable name="npos" select="position() + 1" />
      <xsl:if test="../Phone [position() = $npos]/@PhoneEntity != 'o'">
        <xsl:message>'o' not found</xsl:message>
      </xsl:if>
    </xsl:when>
    <xsl:otherwise>
      <xsl:message>'n' not found</xsl:message>
    </xsl:otherwise>
  </xsl:choose>
</xsl:template>
```

この XSL では、検証の対象とする XML における Mora 要素を検索し、その下位階層にある Phone 要素について、「ノ」対”n”,”o”の対応関係を調べ問題があれば警告を出力するように記述している。このように、各種情報間の制約を記述して XSLT のプロセッサを実行することにより制約を満たさない不整合箇所を検出することが可能になる。また記述された XSL は標準規格によって表現された制約として位置付けられる。

5. XML を利用した検索

各種情報間の依存関係を構造として表現した XML を用いて、種類の異なる情報に跨った情報検索が容易になる。XML の検索には、XPath および XQuery と呼ばれる検索問い合わせ言語の規格が利用できる。また、DOM と呼ばれる規格を用いて、高度な検索を実施するプログラムを作成することも可能である。ここでは、XPath を用いた検索方法を説明する。

XPath とは、XML に階層的に記述された要素に対して、データを特定するための経路を表わす方法の規格である。これを用いれば、指定した条件を満たす属性値を持つ要素を検索することができ、XSL と組み合わせて用いることによって任意の形式で検索結果を出力することができる。以下に、無声化した母音を含む短単位をあらわす SUW 要素を検索する XPath を示す。

無声化母音を含む短単位を検索する XPath

```
/Talk/descendant::SUW[Mora/Phoneme/Phone/@Devoiced
="1"]
```

この場合、Talk 要素の下位階層にある SUW 要素のうち、Phone 要素の属性 Devoiced の値が”1”であるものを特定している。このような XPath の表現を XSL 内に記して用いることにより、所望の情報を抽出することが可能になる。

6. おわりに

CSJ の多様な研究用情報を効率よくメンテナンスし、また公開後の検索の利便性を上げるために導入した、XML およびその関連規格とその利用方法について解説した。CSJ の公開に際しては、全ての情報を余すことなく記述した XML と、主な研究目的に応じて派生させた XML 群を提供する予定である。本稿において示した XML の構造設計やスキーマなどについては、公開に際して変更される可能性があるので注意されたい。

謝辞 CSJ の XML 化全般を担当し本稿執筆にご協力いただいた塚原渉氏(国立国語研究所)に感謝します。また、日頃熱心に議論いただく「話し言葉工学」プロジェクトのメンバーに感謝します。

参 考 文 献

[1] 前川: “『日本語話し言葉コーパス』の設計と実装”, 平成 15 年度国立国語研究所公開研究発表会予稿集, (2003).

[2] 伝: “言語学的対象のオントロジー”, 人工知能学会研究会資料, SIG-SLUD-A202-07, pp.39-44 (2002).

[3] Bird, Liberman: “A formal framework for linguistic annotation”, *Speech Communication*, 33, pp.23-60 (2001).

[4] McKelvie: “The MATE workbench – An annotation tool for XML coded speech corpora”, *Speech Communication*, 33, pp.97-112 (2001).

[5] Asahara, Yoneda, Yamashita, Den, Matsumoto: “Use of XML and relational databases for consistent development and maintenance of lexicons and annotated corpora”, *proc. of LREC*, pp.1372-1378 (2002).