

## XML と XSLT を用いた未解読長唄譜の認識

矢向 正人

九州大学芸術工学研究院

読み方や演奏方法が解明されていない楽譜、すなわち未解読楽譜を XML と XSLT を用いて認識し記述する方法を、長唄の正本に記されている胡麻点の研究を例にとりながら述べる。胡麻点の種類、形状、屈曲、方向、などを XML を用いて記述する工夫、複数の胡麻点の位置関係を記述する工夫を示し、さらに、XSLT を用いて胡麻点に解釈を記した XML データをそのつど変換出力する方法を提案する。この方法で未解読楽譜をデータベース化すれば、未解読記譜法研究の進捗をインターネットを介して研究者間で共有することができる。

## Recognition of Nagauta Notation with Undeciphered Codes by Using XML and XSLT

Masato, Yako

Faculty of Design, Kyushu University

A method using XML and XSLT is proposed for the study of musical notations with undeciphered codes, taking the case of the study on neumatic symbols in nagauta play text. First, undeciphered symbols are encoded with XML tags as <symbols undeciphered> in addition to deciphered symbols in a notation. Next, a conversion program using XSLT is designed suitable for the description of a variety of interpretations of these undeciphered codes. This method will be available for sharing with many scholars the original information of notational data with undeciphered codes and the opportunity of the choice of plural possible interpretations of these codes.

### 1. はじめに

アジアの諸地域、特に、日本、韓国、中国には、構造が明確に規定されていない記譜法や、構造がすべて知られていない記譜法、すなわち、未解読記譜法が数多く存在する。これらの記譜法の研究は近年に着実な進展を見せており、解読手続きについての知識は増えつつある。しかし、解読済みとされる記譜法であっても複数の解釈がある事例もあり、専門分野の異なる研究者がデータに共同で係わることはできないのかなど課題は多い。

さて、近年は、計算機の社会への普及により、音楽の持つ情報をデータベース化しそれを活用する試みが増えている。特に、今日、インターネットを介して容易に音楽データを交換できる状況であることが、音楽データを蓄積し視覚表示させる方法を多様にしており、この結果、音楽のこれまでにない研究方法が可能になっている。記譜法の研究にも、インターネットによるデータの交換を念頭においた楽譜情報のデータベース化が、応用されるべきであると思われる。但し、このためには、研究対象とする楽譜のデータが、インターネットの使用に適した形式で書かれていることと、オペレーティング・システ

ムやアプリケーションの違いに左右されずに利用できることが条件となる。この条件を満たす方法の一つが、XML (Extensible Markup Language) によるデータ形式の利用である。

XMLが、World Wide Web Consortium (W3C) 勧告となった1998年以降、音楽データ形式の定義にXMLを用いる動きが顕著となっており、MusicXML (Good 2001) や WEDELMUSIC など、XMLを用いた楽譜のデータ形式が普及している。この傾向に呼応して、音楽研究でもXMLの利用が増えている。2002年8月に著者が参加した国際音楽学会の計算機研究グループの会議では、発表件数の半数以上がXMLによるデータの保存を念頭においたものであった。筆者らは、五線譜法に対応するMusicXMLが公開されるのとはほぼ同時期に、長唄譜のデータ形式にXMLを用い、以後、日本の伝統的な記譜法のためのXMLによるデータ形式の研究を継続的に行ってきた (矢向 2001, 2006, 矢向; 岸田 2001, 菊地; 矢向 2002, 2005)。本研究は、未解読楽譜の研究にXMLによるデータ形式を用いる提案である。長唄正本に記された胡麻点のデータ化を題材に、未解読楽譜を認識し蓄積する新し

い考え方と方法を示す。

## 2. 記譜法の特徴の認識

記譜法の研究では、研究対象とする記譜法の持つ構造をもれなく検討することが不可欠である。しかし、インターネットの援用や研究者間でのデータの交換を研究の前提とするなら、それを他の記譜法の規則に変換するための工夫がまず必要である。一般に、記譜法の違いとして認識されるのは、

- 1) 用いられる記号の種類の違い
- 2) 用いられる記号の多寡の違い
- 3) 表音楽譜か奏法楽譜かの違い
- 4) 中心として記述される情報の違い
- 5) 付帯的な記号の関与の違い

である。現行の長唄譜には、三味線文化譜、小十郎譜、青柳譜、佐吉譜、東吉郎譜、六左衛門譜などがあるが、1)～5)から比較するなら、それぞれの記譜法の特徴と、それらに共有される長唄譜の一般的な構造とを明らかにできる。異なる記譜法のデータ形式への変換を見込んだデータ形式の策定に際しては、1)～5)の認識を踏まえたうえで、汎用性のあるデータ記述の枠組みを作る必要がある。XMLの利用はこの枠組み作りのために有効である。

```
<曲情報>
  <曲名>京鹿子娘道成寺</曲名>
</曲情報>
<譜情報 type="文化譜">
  <小節>
    <三味線>
      <調弦 name="三下り"/>
      <合図>{ハッ}</合図>
      <注釈>{早く}</注釈>
      <三味線音符>
        <奏法 type="消シ"/>
      </三味線音符>
      <三味線音符>
        <三の糸 position="1"/>
      </三味線音符>
      <唱歌>チン</唱歌>
    </三味線音符>
  </小節>
</譜情報>
```

図1：三味線文化譜のXMLデータ



図2：三味線文化譜の編集アプリケーション

## 3. XMLを用いた現行譜のデータ形式

筆者は、長唄譜のデータの入力、編集、視覚表示、分析において今後の要求に耐えうるデータ形式をXMLを用いて定義した。図1は、現行の長唄譜である三味線文化譜のデータ例である(矢向; 岸田 2001)。現在、このデータ形式は、長唄の主要な記譜法である、三味線文化譜、小十郎譜、青柳譜、佐吉譜、東吉郎譜に対応している。上述の1)～5)を検討しながら、これらの記譜法に共有される規則を同じ階層で括り定義した。それぞれのデータを異なる記譜法に変換することも可能である。筆者らはさらに、このデータ形式で記述されたデータを視覚表示させ、画面上で入力編集できるアプリケーションを開発した(菊地; 矢向 2005)。図2は、その一例であり、図1のデータに対応している。図1では、長唄譜を構成する要素のそれぞれが、XMLでマークアップされて記述されている。たとえば、長唄譜では、三味線の勘所や弾かれる糸に加えて口唱歌が書かれるが、図1では、<奏法>、<文化譜の糸>、<唱歌>、<三味線音符>の内側で同じ階層に定義した(矢向; 岸田 2001)。また、<小節>で小節線が記述されるが、小節線がない記譜法でも、そこに<小節>のレベルを認識することができる。こうして、複数の長唄譜に共有される階層構造を明確にすることができる。

XMLを長唄譜のデータ形式に用いてその構造を明確にできたので、筆者は、長唄譜以外の記譜法をXMLを用いて記述し、その有効性を検討した。たとえば、図3は、雅楽の明治選定による筆築の譜(図4)の、XMLによる簡略化したデータである。筆築の譜では、仮名譜と本譜が並記されるが、XMLによる記述では、それらを拍点とともに<音符>のレベルで認識して同じ階層で書く。図5は、真言宗豊山派声明の《教化》の譜(図6)の、XMLによるデータである。声明の博士は、その向きが五音など相対的な音高を表すが、図5では、ユリなどの唱法と五音を表す博士を同じ階層で書いている。<博士>の内側に<唱法>と<五音>が記述される。また、筆者らは、琉球古典音楽でもちいられる『声楽譜附野村流工四』のためのデータ形式を、XMLにより定義した。すでに、沖縄県立芸術大学の金城厚らにより、表計算ソフトLotus1-2-3上で利用できる工四のデータベースが作られているが(大宮 1993)、筆者らは、このデータベースを工四の形式に視覚表示できるXMLのアプリケーションを開発した(矢向 2006)。他方、2002年には、韓国人により韓国の古楽譜のためのXMLデータ形式も提案されており、XMLによる伝統的記譜法のデータ形式の有効性は国際的に認知されつつある(Lee; Downie; Renear 2002)。

## 4. XMLを用いた記譜法の解釈の記述

現行の記譜法のデータ形式にXMLを用いることによりその階層構造を明確に認識することができるが、記譜法の構造が不明確であったり、隔々まで知られていなくても、XMLを用いてデータ形式を策定し研究を推進することができる。次にそれを検討しよう。XMLのユーザは、音楽データに限らず、一般にデータを視覚表示

```

<曲情報>
  <曲名>越殿楽</曲名>
  <調子>平調</調子>
  <拍子>早四拍子</拍子>
</曲情報>
<譜情報 type="明治選定">
  <小拍子>
    <音行>
      <本譜>六</本譜>
      <仮名譜>チ</仮名譜>
      <拍点 大きさは標準 />
    </音行>
  </小拍子>
  <音行>
    <本譜>四</本譜>
    <仮名譜>ラ</仮名譜>
  </音行>
</小拍子>
</譜情報>

```

図3：筆楽譜のXMLデータ

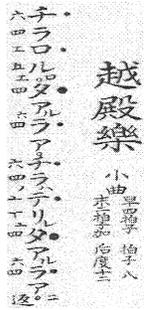


図4：明治選定による筆楽譜

```

<曲情報>
  <曲名>教化</曲名>
</曲情報>
<譜情報 type="真言宗豊山派">
  <詞章>しか
  <博士>
    <折れ目>1</折れ目>
    <仮名>カ</仮名>
    <五音>
      <方向>左下</方向>
      <方向>左上</方向>
    </五音>
  </博士>
  <詞章>ラ
  <博士>
    <唱法>ユ</唱法>
    <五音>
      <方向>左上</方向>
    </五音>
  </博士>
</詞章>
</譜情報>

```

図5：声明譜のXMLデータ



図6：声明譜《教化》

させて認識する。しかし、XMLのデータは、それへのアクセスがアプリケーションにより特定されない。XMLのデータは、数多くのXML対応のアプリケーションを用いて視覚表示させるのみならず、HTMLやXHTMLなどインターネットで普及している他のマークアップ言語にデータを変換して視覚表示させることも可能である。また、その途中段階で、公開されている他のXMLデータにアクセスし、それらとデータを比較したり混在させることも可能である。このことは、XMLのデータに対する認識方法を拡張させることになる。

たとえば、最近では、XMLをデータベース作成よりも分析のために用いる研究があるが (Haus; Ludovico 2004)、ここでXMLによるデータは、分析の実践をおとして新たな意味を獲得している。XMLの使用をこのように楽譜データに対する認識方法の拡張と捉えるなら、それは記譜法の研究にも新たな可能性をもたらす。たとえば、もとの楽譜データに、はじめに定義されなかった意味、すなわち、解釈や註釈を書き込んで視覚表示させる設定も、XMLをデータ形式に用いるなら、容易に可能である。こうして、XMLデータに解釈や註釈を書き加えていく、動的なデータ形式を構想することができる。この設定によるデータ形式は、そのつどデータに解釈を加えるるので、もともと構造が明確に規定されていない記譜法や、構造がすべて了解されていない記譜法、すな

わち、未解読記譜法のためのデータ形式には有効である。

### 5. 未解読楽譜のデータの共有

未解読楽譜の解読と復元の多くは、これまで音楽や舞踊の復元や復興と密接に関わっており、実践的な活動との往復の中で研究が進められてきた。解読されない楽譜が、現実の音符や奏法に対応づけられないままであれば、楽譜としての役割を果たさないことも事実である。しかし、楽譜に記された音が消えてしまったからには、解読はあくまで仮説であるにとどまる。まず、未解読楽譜の解読研究が、現実存在する楽譜に基づくものである以上、解読されていない未解読の楽譜データを共有することを、研究に不可欠な前提と考えたい。データの共有のためには、記号の切れ目や位置関係など、データを構成する記号間の階層関係が一意であることが好ましいが、データのこうした記述方法に合意を得るためには、未解読楽譜のデータを、インターネットでの使用を前提にXMLを用いてデータベース化し、検索し閲覧できる仕組みを作ることが有効である。XMLのこうした援用により、研究の進捗が、特定の研究者のみならず一般のユーザにも知られることになる。

### 6. 未解読楽譜とデータの解釈

本研究では、未解読楽譜を解読するプロセスを次のように認識する。まず、未解読楽譜では、記譜法の正しい解釈を一つに特定することができないと考える。従って、その解読作業は、できるだけ多くの解釈を提示したうえで、可能性の低い解釈を切り捨てていく手順が踏まれる。もし選択肢が一つになれば、そこで妥当な解釈が選択されたと考える。

本稿でXMLを用いて未解読楽譜のデータに対して試みる実践とは、解読の前提を提供するために、可能な限り多くの解釈を書き込むための工夫をすることである。このために複数の解釈を記述し、それらを対等な関係で認識できるデータ形式を策定する。XMLを用いたこの実践により、ユーザは、楽譜データの視覚化に伴うリアリティに加えて、同一の楽譜データに同一の記述形式で複数の解釈が記述されることに伴うリアリティを得ることとなる。こうして策定されたデータ形式と蓄積されたデータを、他の研究者やユーザも検索できるなら、解読のための研究手順が共有されることになる。

### 7. XMLを用いた未解読楽譜のためのデータ形式

本研究の未解読楽譜のためのデータ形式の設計方針を以下に示す。

- 1) 楽譜の未解読記号を<記号 未解読>とラベリングしてXMLデータを作る。
- 2) 未解読記号に対する複数の解釈を、1) のXMLデータに対するフィルタとしてそれぞれ1)とは別にXMLで書き、それを1)に適用する。
- 3) 2)の結果を、1) 2)とは別のXMLデータとして出力する。

1)の未解説データに対する、2)の解釈が、3)に出力される。まず、1)で、未解説楽譜のXMLによるデータを蓄積する。どのような記述かがユーザにわかりやすいデータ形式であることがその条件であり、データに対する解釈は記さない。なお、XMLによるデータ形式で入力された楽譜データであれば、校正作業に際して、入力ミスや記号の位置関係の整合性などを、ソフトウェア上でチェックできる。2)3)は、複数の解釈を記述する工夫である。1)2)3)を使用し易くするには、2)3)のデータが、1)の未解説データとできるだけ共通した構造を持つデータ形式で出力されることが好ましい。このため、2)では、1)に複数の解釈を書き込むことができる記述の雛形を作る。なお、1)がオリジナルのXMLデータとして蓄積されていくのに対して、3)はそのつど出力されるデータである。

## 8. 長唄正本と胡麻点

次に、XMLを用いた未解説楽譜の記述例として、長唄正本に記されている胡麻点を題材にして、データ形式を提案する。

長唄をはじめとする三味線音楽の詞章を記した本のうち、江戸時代に刊行された本を、正本と呼ぶ(図7)。長唄正本の他に、浄瑠璃各種の正本が知られる。長唄正本を見ると、詞章の右側に、ハル、クル、キンなどの注記の他に、さまざまな長さや向きを持つ直線および曲線が記されているのを目にする。これらの線状の記号は、胡麻点と呼ばれる。長唄正本の胡麻点は、享保16年(1731)初演の《傾城無間の鐘》の正本に見いだされ、以後の正本でその数が増加していくが、安永期に入ると減少し、天明期以後の正本からは見いだせない。寛政以後の長唄



図7：図版長唄原本集成より《白妙》正本の冒頭部分

正本からは胡麻点は消える。長唄正本に胡麻点が記されている時期と、注記が記されている時期とはほぼ一致する(矢向2005:133)。胡麻点は、能の謡本の胡麻符や平家の墨譜と類似しているため、音高や音価の変化などの楽譜情報を持っていた可能性がある。

胡麻点の研究に先だって、筆者は、長唄正本に胡麻点とともに書かれている注記について検討した。比較的古い作りの旋律と考えられている16曲の正本を研究対象とし、そのうちの8曲の正本から抽出した注記の例を比較し、それをもとに《夜鶴綱手車》の正本の注記について検討した(矢向2005:135-141)。この結果、研究対象とした注記の多くについて指示内容を特定することができた。注記の多くは詞章に即してその唄い方を示すために記されており、たとえば、ハルは、音を緊張した状態にし、かつ、音域を高くして強く唄うなどの結論を得ている(矢向2005:142,147)。

## 9. 未解説楽譜としての長唄正本

注記研究の結果を踏まえて、筆者は、胡麻点の検討を試みた。もし、胡麻点の向き、長さ、太さなどが、音高や音価を指示するならば、それらは長唄の旋律の研究を大きく前進させる。しかし、これまでの研究から、胡麻点の向き、長さ、太さなどと音高や音価との対応関係は、見いだされていない。まず、胡麻点の形状や書き方は、曲目や正本によりかなり異なり、書き方に統一性が見いだされない。よって、同一の胡麻点が、曲目や正本により異なった意味で用いられた可能性がある。このことが、胡麻点の解説を難しくしている。なお、長唄正本に胡麻点や注記の記されている享保から安永までの曲目には、現行譜あるいは演奏の伝承が存在する曲目が多くあり、胡麻点を分析した8曲についても、それらの箇所を現行の演奏と照合することが可能である(矢向2005:136)。しかし、享保-安永の初演当時から、録音資料が残る昭和初期までは、約180年間を隔てており、この間に旋律は変形してしまっている可能性がある。加えて、唄の旋律は三味線のそれよりも演奏家の個人差や演奏される状況に影響されやすいことが知られる(矢向2005:141)。よって、現行の録音資料や演奏を手がかりに、初演時の旋律や唄い方を知ることはできないと考えるべきである。この結果、現行の演奏と胡麻点との対応づけは、現段階では困難であると結論される。胡麻点のカーブの向きをとりあげても、音高やその変化に対応するのか、拍や声の使い方を示すのか、多くの解釈が可能である。

しかし、胡麻点の検討から、その解説に際しての知見もえられた。胡麻点の書き方は、曲目や正本によりさまざまであるが、謡本の胡麻符や平家の墨譜と類似する書き方がみられることも事実である。このため、未解説記号としての胡麻点を、XMLによりデータベース化して、認識しやすい形で共有することは有益である。

## 10. XMLによる胡麻点の記述

本研究では、長唄《白妙》の正本の冒頭部分を、XMLによるデータ形式で記述した。図7は、『長唄原本集成』巻十四に収録されている《白妙》の正本である(和

田他 1936)。詞章の右横に胡麻点が見られる。《白妙》は、安永元年十一月中村座初演。筆者は正本の書誌研究も行った(矢向 2005: 136-138, 159)。なお、《白妙》正本の胡麻点は、長唄正本にみられる胡麻点の中では比較的複雑なタイプである。図7のXMLデータが図8である。図8のデータ形式の工夫を述べる。データは、〈曲情報〉と〈譜情報〉とからなる。〈曲情報〉で、曲名や作曲者や調弦など楽曲全体に関する情報を、〈譜情報〉で、曲の時系列情報を記述する。詞章に付記される胡麻点と注記は〈譜情報〉に記述される。胡麻点を、〈記号 未解決="yes"〉とし、その内側に、〈位置〉、〈切れ目〉、〈種類〉、〈形状〉、〈屈曲〉、〈方向〉、〈長さ〉、〈太さ〉、〈太さ変化〉を記述する。たとえば、詞章の始めの文字である「う」には、上下に二つの胡麻点が付されている。この二つをともに〈記号 未解決="yes"〉と記す。そして、二つの胡麻点の位置関係を、〈位置〉でそれぞれ 001 と 002 とに区別して示す。このうち 001 が基準となる位置であることを、属性で記す。「と」には、上下に二つの胡麻点が付されているが、下側の胡麻点は左右二つの胡麻点から成る。この左右二つの胡麻点の記述のために、〈記号 未解決="yes"〉を入れ子にしたうえで、下左側の胡麻点の位置を 003 と認識し直し、〈左端〉と〈接続〉で左右の繋がりを記述する。なお、「し」の上側の胡麻点も、左右二つの胡麻点から成ると認識できる。こうして、胡麻点の示す意味が不明であっても、それをXMLデータとして記述できる。なお、平家の墨譜などでは、記号は名称を持つが、本稿ではそれらの呼称を用いない。

### 11. 胡麻点の解釈例

胡麻点の解釈は現段階では困難であるが、XMLを用いて胡麻点のさまざまな解釈の可能性を示しておいたうえで、他の種目との比較に供することは有益である。本稿では、《白妙》正本の胡麻の、形状、カーブの向き、方向、長さ、太さ、太さ変化が、音高、音高変化、声質の変化、音価、声の強さ、声の強さの変化、声の太さ、声の太さの変化に関する情報を示していることと仮定し、それぞれに二種類の解釈例を示してみる。無論、以下の解釈は、可能な解釈の一例である。

[胡麻の形状] 解釈1: 音高変化の有無を示す。直線は音高変化なし、曲線は変化あり。解釈2: 声質の変化の有無を示す。直線は声の変化なし、曲線は変化あり。

[胡麻のカーブの向き] 解釈1: 声尻の音高変化を示す。凹は声尻の音高を上げる、凸は声尻の音高を下げる。解釈2: 音高の高低を示す。凹は低めの音高を示す、凸は高めの音高を示す。

[胡麻の方向] 解釈1のみ: 直前の胡麻とのあいだに形成される角度が音高変化を示す。上向きの胡麻ほど高い音高を示す。胡麻と胡麻との角度を音高変化に置換するための工夫を次章のスタイルシートで試みた。

[胡麻の長さ] 解釈1: 音価を示す。長い胡麻は長い音価を、短い胡麻は短い音価を示す。解釈2: 声の強

```
<?xml version="1.0" encoding="shift_jis"?>
<曲情報>
  <曲名>白妙</曲名>
  <調弦>本調子</調弦>
</曲情報>
<譜情報>
  <注記>二の上</注記>
  <詞章>う
    <記号 未解決="yes">
      <位置 基準="001">
        <切れ目>|</切れ目>
        <種類>曲線</種類>
        <形状>凹</形状>
        <屈曲>緩</屈曲>
        <方向>右下右</方向>
        <長さ>中</長さ>
        <太さ>中</太さ>
      </記号>
    <記号 未解決="yes">
      <位置>"001"の下離</位置>
      <種類>直線</種類>
      <方向>右下右</方向>
      <長さ>短</長さ>
      <太さ>細</太さ>
    </記号>
  </詞章>し
    <記号 未解決="yes">
      <位置 基準="002">
        <種類>直線</種類>
        <方向>右上右</方向>
        <長さ>中</長さ>
        <太さ>中</太さ>
        <太さ変化>右太</太さ変化>
      </記号>
    <記号 未解決="yes">
      <位置>"002"の下離</位置>
      <記号 未解決="yes">
        <位置 基準="003">
          <種類>直線</種類>
          <方向>右下右</方向>
          <長さ>短</長さ>
          <太さ>中</太さ>
        </記号>
      <記号 未解決="yes">
        <位置>
          <左端>"003"の右</左端>
          <接続>繋</接続>
        </位置>
        <種類>直線</種類>
        <方向>下右</方向>
        <長さ>短</長さ>
        <太さ>中</太さ>
      </記号>
    </詞章>と
      <記号 未解決="yes">
        <位置 基準="004">
          <種類>直線</種類>
          <方向>右上右</方向>
          <長さ>中</長さ>
          <太さ>中</太さ>
          <太さ変化>右太</太さ変化>
        </記号>
      <記号 未解決="yes">
        <位置>"004"の下離</位置>
        <位置 基準="005">
          <種類>直線</種類>
          <方向>右下右</方向>
          <長さ>中</長さ>
          <太さ>中</太さ>
        </記号>
      <記号 未解決="yes">
        <位置>"005"の下離</位置>
        <記号 未解決="yes">
          <位置 基準="006">
            <種類>曲線</種類>
            <形状>凹</形状>
            <屈曲>緩</屈曲>
            <方向>右下右</方向>
            <長さ>短</長さ>
            <太さ>中</太さ>
          </記号>
        <記号 未解決="yes">
          <位置>
            <左端>"006"の右下</左端>
            <接続>繋</接続>
          </位置>
          <種類>曲線</種類>
          <形状>凸</形状>
          <屈曲>緩</屈曲>
          <方向>右下右</方向>
          <長さ>中</長さ>
          <太さ>中</太さ>
        </記号>
      </詞章>
    </譜情報>
  </譜情報>
</譜情報>
```

図8: 《白妙》正本のXMLデータ

さを示す。長い胡麻は強い声を、短い胡麻は弱い声を示す。

[胡麻の太さ] 解釈1：声の強さを示す。太い胡麻は強い声を、細い胡麻は弱い声を示す。解釈2：声の太さを示す。長い胡麻は太い声を、短い胡麻は細い声を示す。

[胡麻の太さ変化] 解釈1：声の強さの変化を示す。太い胡麻は声を強くする、細い胡麻は声を弱くすることを示す。解釈2：声の太さの変化を示す。太い胡麻は声を太くする、細い胡麻は声を細くすることを示す。

## 12. XSLTによるXMLデータ形式の構造変換

次に、胡麻点にこれらの解釈を施したXMLデータを生成する手順を示す。XMLのための周辺規格が今日では数多く作られているが、XML文書の構造変換のためにはXSLT(Extensible Stylesheet Language Transformations)が使われる。XSLTは、XMLにやや遅れて1999年にW3C勧告となった規格であり、XMLデータから、HTMLデータなど非XMLデータやXMLの別データに変換する用途で多方面で広く用いられる(Bradley 2000)。XSLTでは、そのスタイルシート自体がXMLのタグを使って書かれ、XSLTの名前空間に属する要素をXSLTの命令として使用するため、読みやすく書きやすい。

本稿では、XSLTのスタイルシートによる処理プログラムを用いて、**図8**のXMLデータを、それに複数の解釈を施したXMLデータに構造変換して出力する実例を示す。解釈を施したXMLデータをそのつど増やしていくためには、解釈を記述するためのXSLTのスタイルシートはそれぞれ別々に書かれる必要がある。この目的のために、XSLTのスタイルシートを、データの変換処理のための共通スタイルシート(**図9**)と、それぞれの解釈に対応するスタイルシート(**図10**、**図11**)とに分割した。**図9**、**図10**、**図11**のXSLTスタイルシートは、同時並行的に処理される。**図9**のスタイルシートでは、直前の胡麻点の<方向>との角度を音高変化に変換する処理が行われる。本研究では、二つの解釈を二つのXSLTスタイルシートで示したが、可能な解釈が多い場合でも、スタイルシートを増やすことにより対応できる。また、**図9**、**図10**、**図11**のスタイルシートは、一部を書き換えるだけで、他の記譜法のデータにも容易に応用できるように作られている。**図12**と**図13**は、それぞれ**図10**、**図11**による処理の出力結果である。**図8**のデータの内側の階層に、未解読記号それぞれの解釈が書き加えられて出力される。

なお、**図9**、**図10**、**図11**のXSLTの実行は、oxygen XML エディタ and XSLT デバッガなどを用いて容易に可能である(<http://www.oxygenxml.com/> 2007年4月10日取得)。

## 13. 終わりに

未解読楽譜を認識しその複数の解釈を記述するためにXMLとXSLTを用いる方法を、長唄正本に記される

胡麻点を例にとり述べた。胡麻点を未解読記号として認識し、その種類、形状、屈曲、方向、長さ、太さ、太さ変化を記述する工夫、複数の胡麻点の位置関係を記述する工夫、胡麻点にその解釈を書き加えるためにXSLTスタイルシートを用いる工夫、スタイルシート分割して解釈の増加に対応する工夫を示した。胡麻点の方向については、直前の胡麻点との角度を音高変化に変換する処理を示した。これらの工夫は、今後、未解読楽譜のデータに解釈を施すときの、基本的な方法となりうるものと考えられる。

従来の未解読楽譜の研究では、解読結果の提示に至らない途中段階の研究が、研究者間に認知されずにおかれることが多く、そのことはこの分野の研究を阻む一因になっていたと思われる。しかし、研究とは、誰もが容易にその進捗を進行できるものでなければならない。そのためには、解読を急がず、まずデータとそれに対応する複数の解釈例を提示し、研究の途中段階を共有してみる必要がある。

本研究の手順によるなら、だれもが未解読楽譜のデータをXMLデータ形式として蓄積できる。それらは、HTMLへの変換が容易なので、Webでの公開に適し、研究者間で広く共有することが可能である。また、蓄積された楽譜データを活用するためには、データを検索・閲覧する仕組みを作ることが不可欠であるが、XMLやXSLTの周辺技術が普及した今日では、プログラミングの経験が少なくても、本稿の楽譜データの実用的な検索・閲覧システムを作ることには可能である。記譜法研究の今後において、研究に耐える楽譜データの蓄積が進んでいく事態を想定したとき、本研究の方法とデータ形式を多くの研究者が活用するなら、研究の活性化と発展が期待できると考える。今後は、XMLデータの入力を迅速に行うためのアプリケーションの開発などを課題に進めていきたい。

## 参考文献

- [1] Casten, Gerd. 2001 "NIFFML: An XML Implementation of the Notation Interchange File Format". *Computing in Musicology*, 12, 103-112
- [2] Good, Michael. 2001 "MusicXML for Notation and Analysis". *Computing in Musicology*, 12, 113-124
- [3] Haus, Goffredo ; Ludovico, Luca A. 2004 "Music Segmentation: An XML-oriented Approach". in Wiil, Uffe Kock, ed. *Computer Music Modeling and Retrieval*, Denmark, Esbjerg, 330-346
- [4] 菊地宏一郎；矢向正人 2002 GIDA\_U (XMLを用いた長唄譜のデータ形式)――スキーマのモジュール化『情報処理学会研究報告』48, 15-22
- [5] Lee, Jin Ha ; Downie, J. Stephen ; Renear, Allen 2002 "Representing Traditional Korean Music Notation in XML". Article Abstract 20 of 3rd International Conference on Music Information Retrieval, Paris: IRCAM

- [6] Roland, Perry. 2002 “The Music Encoding Initiative (MEI)”. Proceedings of First International Conference on Musical Application Using XML, 55-59
- [7] 和田正明之、山本桂一郎、木村捨三編 1936 『長唄原本集成（巻十四）』東京：長唄原本集成刊行会
- [8] 矢向正人 2005 長唄正本の注記の研究 - - 現行譜と照合してその意味をさぐる『デアアルテ（九州芸術学

会誌）』21, 133-160

- [9] 矢向正人；岸田哉生 2001 「標準データ記述言語を用いた伝統音楽のデータ形式 - XML による長唄譜のデータ形式」『音楽学』47(1), 55-77

- [10] 矢向正人 2007 「標準データ記述言語を用いた未解説楽譜の認識と記述 - 長唄正本の胡麻点を例にとつて」『音楽学』52(2), 122-138

```

<?xml version="1.0" encoding="Shift_JIS"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  version="1.0">
  <xsl:output method="xml"
    indent="no"
    encoding="utf-8"
    omit-xml-declaration="no"/>
  <!-- 処理対象外をそのまま出力するためのテンプレート -->
  <xsl:template match="*[@comment()|processing-instruction()]">
    <xsl:copy>
      <xsl:apply-templates
        select="*[@comment()|processing-instruction()|text()]">
      </xsl:apply-templates>
    </xsl:template>
  <xsl:template match="* [ @未解説 = 'yes' ]">
    <xsl:copy>
      <xsl:apply-templates
        select="*[@comment()|processing-instruction()|text()]"
        mode="mapping"/>
    </xsl:copy>
  </xsl:template>
  <!-- パラメータ 'value' の値を内容とする <解釈> 要素を出力するテンプレート
  -->
  <xsl:template name="add-interpretation">
    <xsl:param name="value"/>
    <xsl:if test="$value!=''">
      <xsl:element name="解釈"><xsl:value-of select="$value"/></xsl:element>
    </xsl:if>
  </xsl:template>
  <xsl:template match="*[@comment()|processing-instruction()]" mode="mapping">
    <xsl:copy>
      <xsl:apply-templates
        mode="mapping"/>
      <xsl:choose>
        <xsl:when test="local-name()='方向'">
          <xsl:call-template name="add-interpretation.orientation"/>
        </xsl:when>
        <xsl:otherwise>
          <!-- 変数 'value' に要素とその内容に応じて、値を代入。 -->
          <xsl:variable name="value">
            <xsl:choose>
              <xsl:when test="local-name()='種類'">
                <xsl:choose>
                  <xsl:when test="='直線'">
                    <xsl:value-of select="$種類,直線"/>
                  </xsl:when>
                  <xsl:when test="='曲線'">
                    <xsl:value-of select="$種類,曲線"/>
                  </xsl:when>
                </xsl:choose>
              <xsl:when test="local-name()='形状'">
                <xsl:choose>
                  <xsl:when test="='凹'">
                    <xsl:value-of select="$形状,凹"/>
                  </xsl:when>
                  <xsl:when test="='凸'">
                    <xsl:value-of select="$形状,凸"/>
                  </xsl:when>
                </xsl:choose>
              <xsl:when test="local-name()='屈曲'">
                <xsl:choose>
                  <xsl:when test="='緩'">
                    <xsl:value-of select="$屈曲,緩"/>
                  </xsl:when>
                  <xsl:when test="='急'">
                    <xsl:value-of select="$屈曲,急"/>
                  </xsl:when>
                </xsl:choose>
              <xsl:when test="local-name()='長さ'">
                <xsl:choose>
                  <xsl:when test="='夏'">
                    <xsl:value-of select="$長さ,長"/>
                  </xsl:when>
                  <xsl:when test="='中'">
                    <xsl:value-of select="$長さ,中"/>
                  </xsl:when>
                  <xsl:when test="='短'">
                    <xsl:value-of select="$長さ,短"/>
                  </xsl:when>
                </xsl:choose>
              <xsl:when test="local-name()='太さ'">
                <xsl:choose>
                  <xsl:when test="='太'">
                    <xsl:value-of select="$太さ,太"/>
                  </xsl:when>
                  <xsl:when test="='中'">
                    <xsl:value-of select="$太さ,中"/>
                  </xsl:when>
                  <xsl:when test="='細'">
                    <xsl:value-of select="$太さ,細"/>
                  </xsl:when>
                </xsl:choose>
              <xsl:when test="local-name()='太さ変化'">
                <xsl:choose>
                  <xsl:when test="='右太'">
                    <xsl:value-of select="$太さ変化,右太"/>
                  </xsl:when>
                  <xsl:when test="='左太'">
                    <xsl:value-of select="$太さ変化,左太"/>
                  </xsl:when>
                </xsl:choose>
            </xsl:variable>
            <xsl:call-template name="add-interpretation">
              <xsl:with-param name="value" select="$value"/>
            </xsl:call-template>
          </xsl:otherwise>
        </xsl:choose>
      </xsl:copy>
    </xsl:template>
  <!-- 方向から音高を選ぶ -->
  <xsl:template name="add-interpretation.orientation">
    <xsl:param name="current"/>
    <xsl:variable name="text">
      <xsl:choose>
        <xsl:when test="$current=''"><xsl:value-of select=""/></xsl:when>
        <xsl:otherwise><xsl:value-of select="preceding:方向[1]"/></xsl:otherwise>
      </xsl:choose>
    </xsl:variable>
    <xsl:choose>
      <xsl:when test="$content=''">
        <!-- 直前の<方向>がない場合 -->
        <xsl:value-of select="$方向,単独"/>
      </xsl:when>
      <xsl:otherwise>
        <xsl:variable name="number">
          <xsl:choose>
            <xsl:when test="$content='上'">0</xsl:when>
            <xsl:when test="$content='上右'">1</xsl:when>
            <xsl:when test="$content='上右'">2</xsl:when>
            <xsl:when test="$content='右'">3</xsl:when>
            <xsl:when test="$content='右'">4</xsl:when>
            <xsl:when test="$content='右'">5</xsl:when>
            <xsl:when test="$content='右'">6</xsl:when>
            <xsl:when test="$content='下'">7</xsl:when>
            <xsl:when test="$content='下'">8</xsl:when>
          </xsl:choose>
        </xsl:variable>
        <xsl:when test="$current=''">
          <!-- call recursively -->
          <xsl:call-template name="add-interpretation.orientation">
            <xsl:with-param name="current" select="$number"/>
          </xsl:call-template>
        </xsl:when>
        <xsl:otherwise>
          <xsl:variable name="differ" select="$number - $current"/>
          <xsl:choose>
            <xsl:when test="$differ=-8 or $differ=8">
              <xsl:value-of select="$方向,180度"/>
            </xsl:when>
            <xsl:when test="$differ=0">
              <xsl:value-of select="$方向,0度"/>
            </xsl:when>
            <xsl:when test="$differ<0">
              <xsl:value-of select="$方向,伏角"/>
            </xsl:when>
            <xsl:when test="0<=$differ">
              <xsl:value-of select="$方向,仰角"/>
            </xsl:when>
          </xsl:choose>
        </xsl:otherwise>
      </xsl:choose>
    </xsl:variable>
    <xsl:call-template name="add-interpretation">
      <xsl:with-param name="value" select="$text"/>
    </xsl:call-template>
  </xsl:template>
</xsl:stylesheet>

```

図9：XSLTによる変換処理の共通スタイルシート

