

楽譜記述言語 MusicXML からの楽譜自動点訳

A Translation System for Braille-score from MusicXML

田村直良[†]
Naoyoshi Tamura後藤大輔[‡]
Daisuke Gotoh後藤敏行[§]
Toshiyuki Gotoh

1. まえがき

点字楽譜とは、2 次元的に表記される五線譜の情報を、6 点字を用いて 1 次元的に表現する楽譜の表記法であり [1]、視覚障害者の音楽教育や音楽情報の伝達手段として重要なものである。点字楽譜は、19 世紀にフランスのルイ・ブライユ (Louis Braille, 1809~1852 年) という盲学校の教師によって考案され、その後改良されて現在の形になっている。点字楽譜の作成には、点字楽譜の表記法に関する専門的知識と楽譜の論理的理解が必要なため、少数のボランティア団体によって支えられているのが現状であり、需要に応えきれていない。

このような背景のもとで、これまでに、楽譜の自動点訳を目指した研究が行われてきた。当初は、五線譜印刷用の電算写植データから点字楽譜の自動生成を目指した研究 [2] や、MIDI (Musical Instruments Digital Interface) データから点字楽譜を生成するシステムに関する研究 [3] が進められた。しかし、電算写植データについては、一般にほとんど普及していないという問題や、楽譜の論理構造を表現するのに適していないという問題があった。一方、MIDI 形式は、電子楽器の演奏のために制定された規格であり、音の強弱や高さが物理量として表現されるため、正確に楽譜記号に対応付けることが困難であった。その後、NIFF (Notation Interchange File Format) データから点字楽譜を生成する研究 [4] も進められてきた。しかし、NIFF は、楽譜を図形的な情報を用いて表現した形式であり、楽譜を論理的な構造として捉えることには適していなかった。

一方、最近のインターネット技術の進展に伴い、XML を拡張した楽譜記述言語が提案されている。現在、日本と米国で同一名称 (MusicXML) で異なる仕様の 2 種類の楽譜記述言語がある [5, 6]。これらの仕様に基づいた楽譜作成ソフトウェアやビューアも開発されており、健常者についてはインターネット上の楽譜情報にアクセスできる環境が整ってきた。このような背景から、筆者らはこれまで、MusicXML (日本版) から点字楽譜を生成するシステムを試作した [7]。MusicXML (日本版) は、楽譜記号の位置情報をもとに五線譜を図として記述する仕様であるため、表現の自由度が大きい反面、記述された楽曲の論理構造の解析が難しい。このような問題に対して、筆者らは、楽譜の論理構造に着目して記述する MusicXML (米国版) をベースに、楽譜の論理構造を点字楽譜の構文構造に変換することによって、点字楽譜を自動生成する手法を提案するとともに、平成 17 年度初めから試作システムを公開している。一方、最近、本研究とは別に、MusicXML (米国版) から点字楽譜を生成するシステムの開発プロジェクト DaCapo (DZB) や、ノーテーションソフトと組み合わせて MusicXML (米国版)

に対応したシステム GOODFEEL (Dancing Dots) などがある。

以下本稿では、筆者らが開発したシステムにおいて MusicXML (米国版) で表現された五線譜の論理構造を点字楽譜の論理構造に変換して点訳を行う方法と、試作した点訳システムの構成について述べる。

2. 点字による楽譜表記

2.1 点字楽譜の特徴

点字楽譜は、楽譜記号の 2 次元的な配置として表記される五線譜の情報を、通常の点字を用いて表現するものであり、一種の言語体系を構成している。一般に使用される点字は、3 行 2 列の 6 点の点の有無によって 64 通りの文字を表現する。五線譜の音楽記号は点字で表記可能な 64 通りより多くなるため、1 つの点字に複数の意味を持たせたり、複数の点字を組み合わせることで 1 つの楽譜記号を表現したりする。

2.2 音符の表記

図 1 に示すように、点字楽譜の音符は、上 4 点で 7 つの幹音を、下 2 点で音の長さを表現する。また、7 つの幹音がどの音域に属するかを示すために、音列記号を音符の前に配置する。音列記号は、すべての記号に付ける必要はなく、直前の音符と同じ音列でかつ音程が 5 度以下の場合や、直前の音符が異なる音列でも音程が 3 度以下の場合に省略することができる。一方、音の長さは 2 つの点で表現するために、図 1 に示すように全音符と 16 分音符、2 分音符と 32 分音符、4 分音符と 64 分音符を同一の記述で表現している。

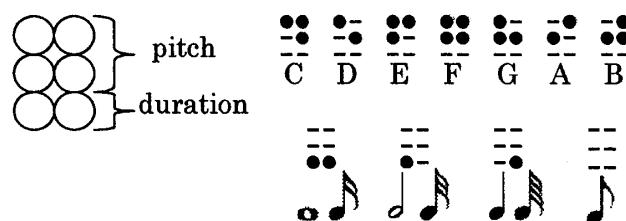


図 1 点字楽譜での音符の表記

2.3 楽譜の表記

点字楽譜は、1 行目に調号と拍子記号を記述し、2 行目から音符などの記号を記述する。図 2 の曲の例では、1 行目にフラットを表す点字を用いて変イ長調であることを示し、その後 4/4 の拍子記号が続いている。2 行目以降は音符などを記述するが、ピアノ譜の場合には、右手記号と左手記号を各行の先頭に配置し、右手と左手のバ

[†] 横浜国立大学大学院環境情報研究院, tam@ynu.ac.jp

[‡] ヤフー株式会社

[§] 横浜国立大学大学院環境情報研究院, gotoh@sci.ynu.ac.jp

ートを交互に記述する表記法が一般的である。表1は、点字楽譜で用いられる楽譜記号の例を示したものであり、多くの記号が複数の点字を組み合わせた「単語」として表記される。クレッシェンドや繰り返し記号等の時間的に幅を持つ記号は、開始記号と終了記号をペアで用い、該当部分を囲むことによって表現する。

また、点字楽譜では1つの小節に複数の旋律がある場合を部分けという。点字楽譜は2次元的な表記ができないため、図2の「multi-melodies」で示すような点字記号を複数の旋律の間に入れることによって、1次元的に部分けを表現する。また、小節の途中で旋律が分かれる場合は内分けといい、旋律の間に入れる記号と、旋律が分かれる切れ目を表す記号を配置することによって表現する。また、和音については、音符法と音程法という二種類の表記法がある。音符法は、基準の音符(左手では最低音、右手では最高音)に続いてその音符に近い側から順に、下がり音符(点字の下4点を使って音の高さのみを表記した音符)を並べることによって表現する。一方、音程法は、和音を基準の音符からの音程の差によって表現する。図2は、音程法を用いた例であり、基準の音符に続いて3度や2度、5度を表す点字を記述することによって和音を表現している。

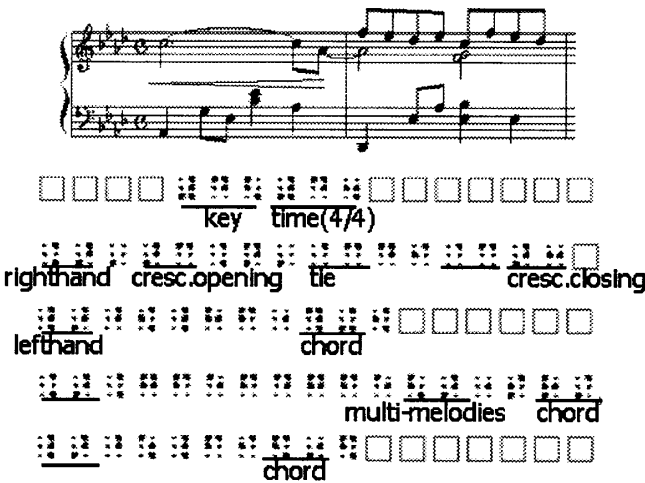


図2 点字楽譜の例

点字	楽譜記号
⠠⠠⠠⠠⠠⠠⠠⠠	4/4
⠠⠠⠠	トリル
⠠⠠⠠⠠	アクセント
⠠⠠⠠	フラット
⠠⠠⠠⠠⠠⠠⠠⠠	クレッシェンド
⠠⠠⠠⠠	mf
⠠⠠⠠⠠⠠⠠	反復記号
⠠⠠⠠	終止線

表1 点字楽譜で用いられる記号の例

3. 楽譜記述言語について

3.1 MusicXMLの概要

MusicXMLは、五線譜を記述するためのXMLベースの言語仕様である。MusicXMLは、日本のミュージカル・プラン社が2001年に制定した仕様と、アメリカのRecordare LLCが2002年に制定した仕様の異なる2種類が提案されている。MusicXML(米国版)は、グラフィカルな構造よりも各楽譜要素間の論理的な構造の記述を目指したものであり、点字楽譜への翻訳を考えた場合、楽譜構造の論理的な理解が不可欠となるために、今回の研究ではMusicXML(米国版)を入力として用いている。

3.2 MusicXMLの論理構造

MusicXMLの記述は、楽譜、パート、小節の順に階層化されている(図3)。それぞれのパートには、小節を表すmeasure要素が記述され、その下には、調号や拍子を表すattribute要素と、音符を表すnote要素などの楽譜記号が記述される。和音は、基準の音符以外の音符の子供要素にchord要素を付加することによって表現する。連符は、連符を構成するそれぞれの音符の子供要素に、何連符かを示す値を保持することによって表現する。また、MusicXMLには右手、左手のパートの区分はなく、右手や左手を含めて一つのパートに複数の旋律を記述する場合には、backupという時間を前に戻すことを表す要素を用いて表現することになる。

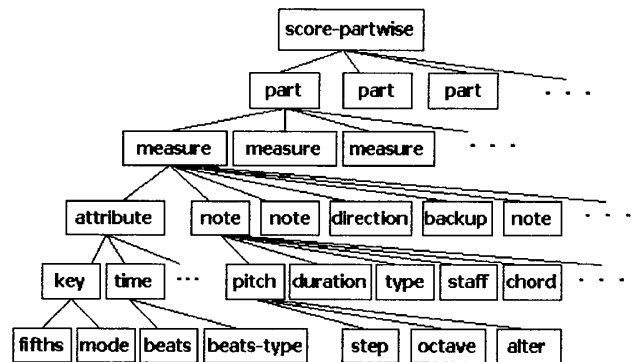


図3 MusicXMLの論理構造(例)

4. 楽譜自動点訳システム

4.1 システムの概要

図4に本システムの概要を示す。最初に、入力したMusicXMLの楽譜をXMLプロセッサで構文解析する。XMLプロセッサとは、XML文書を検証したり、別のアプリケーションで利用可能な形に変換したりするソフトウェアのことであり、これによってXSLT(XSL Transformations)によるスタイルの指定や、DOM(Document Object Model)によるXMLの操作などが可能となる。XSLTとは、XML文書の木構造を別の構造に変換するためのツールであり、DOMとは、XML文書のデータをアプリケーションに受け渡す汎用的なAPI(Application Program Interface)である。次に、XSLTとDOMを併用することによって、MusicXMLの論理構造を点字楽譜の論理構造に変換する。さらに、MusicXMLと点字楽譜の要素を対応付けた点字楽譜変換辞書を参照しながら変換を行い、点字フォーマットの一種で

ある base 形式のデータを出力する。

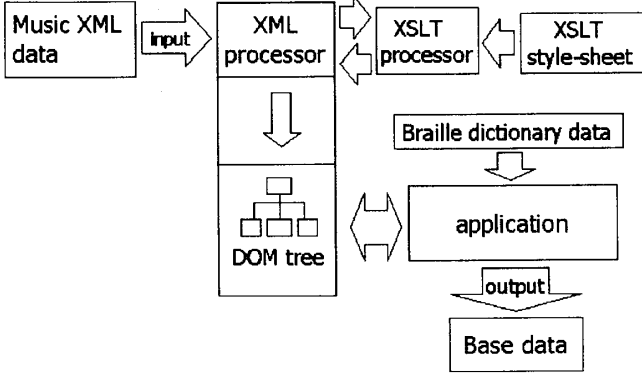


図4 楽譜点訳システムの概要

4.2 点字楽譜の論理構造

MusicXML から直接点字に翻訳することは困難であるため、一度、MusicXML の論理構造を点字楽譜の論理構造に変換する。そのために、我々は点字楽譜の論理構造を定義した。その例を図5に示す。楽譜、パート、小節の順に階層化する点は MusicXML と同様だが、それぞれの小節には、調号や拍子を表す attribute 要素の他に、右手を表す righthand 要素と左手を表す lefthand 要素を記述する。その下には旋律を表す melody 要素を記述するが、この要素には、開始時間と終了時間の属性を付加し、これによって部分け、内分けを判断する。melody 要素の下には音符要素などの楽譜記号要素を記述する。和音は、chord 要素の下に和音を構成する音符要素を、右手は高い音から、左手は低い音から順番に配置することで表現する。連符は、連符を構成する最初の音符のみ何連符かを示す値を保持することによって表現する。また、両手にかかる強弱記号、曲想記号は、右手が全休符の場合以外、全て右手に配置する。

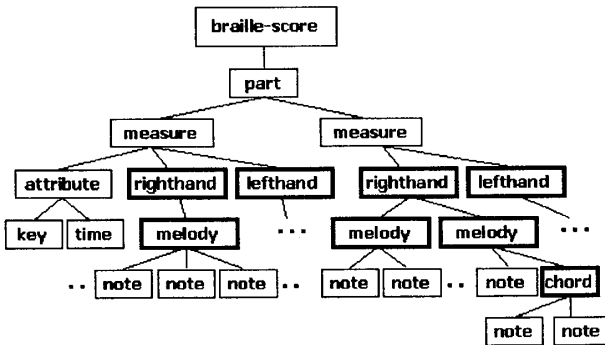


図5 点字楽譜の論理構造 (例)

4.3 構造変換処理の類型

点字楽譜の文法に対応した構造への変換処理は、大きく以下の5種に分類することができる (図6)。

- Ⓐ MusicXML の構造をそのまま用いる処理。(note の子供要素の一部や backup 要素など)
- Ⓑ 子供要素を親要素の前後に移動する処理。(accidental 要素や dot 要素など)
- Ⓒ 同じグループの要素をまとめる処理。(右手左手要素や chord 要素、部分けなど)
- Ⓓ 冗長な箇所を単純化する処理。(note の子供要素や time-modification 要素など)
- Ⓔ その他の処理。(direction 要素や key 要素など個別の処理)

- Ⓒ 同じグループの要素をまとめる処理。(右手左手要素や chord 要素、部分けなど)
- Ⓓ 冗長な箇所を単純化する処理。(note の子供要素や time-modification 要素など)
- Ⓔ その他の処理。(direction 要素や key 要素など個別の処理)

変換は、XSLT と DOM による。XSLT は、パターンマッチング以外の複雑な変換には対応できないため、パターンマッチングで変換可能な音符の子供要素の入れ替えや右手左手の構造変換などの操作には XSLT を用い、和音の音符要素を chord 要素にまとめる変換や、部分け、内分けの処理などの複雑な操作には DOM を用いている。

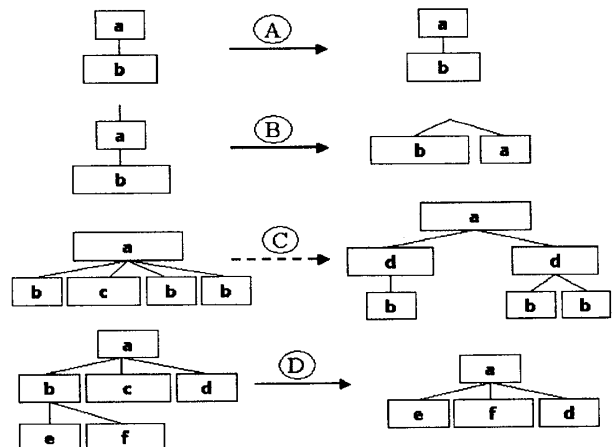
4.4 部分け、内分けの処理

MusicXML では、部分けを、backup という巻き戻し要素を用いることによって表現している。その例を図7に示す。この方法は、楽譜を記述するのに多様性(曖昧性)があり、また、どの音符がどの旋律に属するのかを表現することができない。一方、点字楽譜の場合の部分け、内分けは、旋律ごとに分離して記述する必要がある。このために、MusicXML の楽譜の解析にあたって、旋律の連続性の判別が必要になる。

これに対して本研究では、backup 要素で区分された小節内の要素を、拍を基準に再分割し、それぞれの連続する旋律要素の類似性に基づいて結合する方法を採っている。具体的な処理の流れを以下に示す。

- ① 小節内の要素を、拍を基準に再分割し、それぞれの旋律要素に開始時間と終了時間の属性を付加する。
- ② 旋律要素間で時間的に連結する可能性がある候補を探す。
- ③ ②で発見された旋律要素の候補の中で、類似性の高い候補を結合する。
- ④ すべての旋律要素がグループに分けられるまで②、③を繰り返す。

なお、旋律要素間の類似性の判定基準として、音の高さや長さの平均値と分散、音符の音程変化などを考慮している。



—————▶ XSLT による変換
 - - - - -▶ DOM による変換

図6 構造変換処理の類型

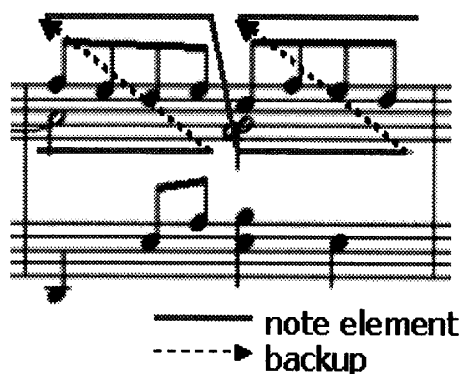


図7 MusicXMLにおける部分けの記述の例

5. web 対応のインターフェース

インターネット上に普及しつつある MusicXML をより有効に利用するために、楽譜自動点訳システムを web 経由で利用するためのインターフェースを開発した。図8にシステムの概要を示す。

まず、入力ページから、入力された MusicXML による楽譜を受け取り、受け取った XML 文書のデータ構造を検証したのちに、楽譜点訳プログラムを起動する。次に、エラーなどの解析のため、システム実行時におけるデータを log ファイルに記録するとともに、翻訳結果の点字楽譜をクライアントのブラウザに送信する。クライアント側は、翻訳結果を base 形式のファイルとしてダウンロードすることもできるようになっている。また、本システムの改良点を探るため、アンケートの欄も設けている。

6. 点字楽譜翻訳の処理例

6.1 楽譜点訳作業の流れ

印刷された楽譜から点字楽譜を作成するには、まず、スキャナーを用いて印刷された楽譜をコンピュータに取り込み、市販の楽譜認識ソフトを用いて認識後、MusicXML として保存する。そして、web ブラウザから楽譜点訳システムを起動し点字楽譜に変換する。本研究では、楽譜認識ソフトとして株式会社河合楽器製作所のスコアメーカーを用いている。また、今回試作した楽譜点訳サーバは、希望者に対し試験的に公開 (http://gotoh-lab.jks.ynu.ac.jp/braille_music_score/) している。

6.2 楽譜点訳結果

現段階で、本システムはピアノ譜と一段譜のみを対象としており、音符、休符など、楽譜の基本的な記号の点訳機能については実装済みである。

また、複数の楽譜**を点訳させ、その点訳時間を調べた。平均すると、楽譜 1 ページあたりの点訳時間は約 3.32 秒であった。

7. むすび

楽譜点訳は、少数のボランティアが手作業で行っているのが現状であり、需要に応えられていない。一方、ネットワーク環境で楽譜情報を扱う枠組みも整いつつある。この

ような背景から、本報告では、MusicXML(米国版)で表記されたピアノ譜の音楽情報から点字楽譜を生成する自動翻訳システムを提案した。試作システムにより音符、休符など、楽譜の基本的な記号の自動点訳を実現し、本システムの有効性を検証することができた。

楽譜記号の種類が多く、新しい記号も適時作られるため、ユーザーが簡単に拡張できる機能を組み込むことが今後の課題となる。また、ピアノ譜、一段譜以外の楽譜への対応や、視覚障害者が使いやすいインターフェースの改良も課題である。

謝辞

本研究は科学研究費(15500062)の一部支援のもとで実施した。本研究を進めるにあたり、システム要件に関する意見や点訳結果の評価などの面でお世話になった楽譜点訳ボランティア団体トニカの皆様に感謝する。

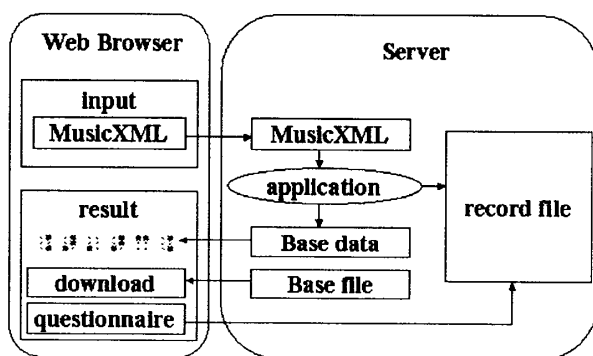


図8 web 経由システムの概要

文 献

- [1] 文部省, 点字楽譜の手引き, 日本ライトハウス, 大阪, 1984.
- [2] 松島俊明, 田中守, 橋本周司, 大照完, 江成保徳, “電算移植データを用いた楽譜の自動点訳システム”, 第10回感覚代行シンポジウム, pp.93-97, December 1984.
- [3] 佐藤孝三, 小松実, 小山善文, 松崎悟, 神田一神, “点字楽譜作成システムの構築”, 情処学音楽情報科学研報, 2000-MUS-36-7, vol.2000, no.076, pp.37-42, August 2000.
- [4] D. Langolff, N.B. Jessel, D. Levy, “NIFF transcription and generation of Braille musical scores”, the virtual score:representation, retrieval, restoration, Computing in Musicology 12, pp.34-44, 2001.
- [5] 江守幸一, 黒田大輔, 中塚秀子, “IPA 次世代アプリケーション開発事業報告-XML を用いた楽譜表示演奏システムの開発-”, ミュージカル・プラン, 2001.
- [6] M. Good, “MusicXML for notation and analysis”, the virtual score:representation, retrieval, restoration, Computing in Musicology 12, pp.113-124, 2001.
- [7] 遊佐郷平, 足立啓, 後藤敏行, 田村直良, 江守幸一, “楽譜記述言語 MusicXML を利用した点字楽譜生成システム”, 情処学デジタル・ドキュメント研報, 2004-DD-42, vol.2004, no.011, pp15-22, January 2004.

** Invention 4 BWV 775 (J.S.Bach) など 15 曲