

音楽情報処理のための共通データフォーマット

CrestMuseXML 全体構想と基本設計方針 *

北原 鉄朗, 橋田 光代, 片寄 晴弘

(科学技術振興機構 CrestMuse プロジェクト / 関西学院大学)

1 はじめに

本研究の目的は、現在行われている様々な音楽情報処理に関する研究のための入出力データフォーマットを共通化し、またそのための標準 API を提供することで、本分野の技術の底上げと活性化を促すことである。共通化された入出力フォーマットおよび API の提供は、様々なメリットをもたらすと考えられる。第 1 に、様々な中間状態に関するデータ表現を用意することで、処理をコンポーネント化し、自由に処理を組み合わせることができるようになる。たとえば、周波数解析結果を表す共通化されたデータフォーマットが存在すれば、ある自動採譜システムにおける周波数解析部だけを他の研究者が開発したものに置き換えてみる、といったことも簡単にできるようになる。また、たとえば演奏情報を表すデータフォーマットがあれば、音響信号からシンボル化された演奏情報を抽出する自動採譜システムと、シンボル化された演奏情報を解析する演奏解析システムとを組み合わせるといったこともできるようになる。第 2 に、標準 API や標準ツールを提供することで、音楽データを扱うための敷居を下げ、研究者間での開発の重複を省くことができる。たとえば、スタンダード MIDI ファイルや WAV ファイルの読み書きは、個々の研究者がそれぞれ実装していることも少なくないであろう。こういった重複を省き、個々の研究のオリジナルな部分の開発に労力を注ぐことができるようになる。

CrestMuse プロジェクトでは、上記の目的を達成するため、XML ベースの総合的な音楽データフォーマットとその標準 API を開発している。音楽データを記述するための XML フォーマットには、MPEG-7[1], MusicXML[2], WEDELMUSIC Format[3], Music Encoding Initiative (MEI)[4], MPEG Symbolic Music Representation (SMR)[5] など様々なものが存在する。しかし、これらの多くは、音楽情報処理研究用途としては記述できる音楽データの種類が不足していなかったり、拡張性が十分でなく、上記の目的を達成するに十分なものではなかった。我々は、多様な音楽データをすべて単一の XML フォーマットで記述するのではなく、音楽データの種類(粒度)ごとに異なる XML フォーマットを策定し、異なる XML フォーマット内のデータ間の関係を XLink/XPointer による参照の形で記述する。この方針により、既存のフォーマットを取り込んだり、ユーザが新たなフォーマットを追加したりといったことが柔軟にできるようになる。この一連の XML フォーマットを総称

して CrestMuseXML と呼ぶ。本稿では、CrestMuseXML およびその API の基本的な考え方や現状の仕様について述べる。なお、CrestMuseXML の最新情報は <http://www.crestmuse.jp/cmxml/> に掲載する予定である。また、文献 [6] も参照されたい。

2 CrestMuseXML の基本方針

我々は、CrestMuseXML が備えるべき性質として、次の 4 つを要請する。

要請 1 マルチレイヤー

粒度の異なる情報を統一的枠組みのなかで自然な形で記述し、処理できる必要がある。扱うべき内容としては、信号・波形レベルの情報から旋律や楽曲の構造、さらには感性情報など様々なものが考えられる。

要請 2 既存規格との互換性

音楽データを表す XML 規格はすでにいろいろと存在し、利用され始めている。こうした利用実績のある規格を取り込むべきである。その際には、既存の規格を独自に拡張して用いるべきではない。既存規格への独自の拡張を行ってしまうと、その規格のバージョンアップ時に対応が難しくなると予想される。

要請 3 拡張性

あらゆる種類のデータの記述方式をあらかじめ網羅的に定義するのは事実上不可能であるため、拡張性は必須である。個々の研究者が独自に拡張したものを矛盾なく統合できることが望ましい。

要請 4 標準ツールの提供

XSLT や XLink といった周辺規格の充実により XML 自体が巨大な規格になりつつあるため、XML に精通していない人でも容易に利用できるようにする必要がある。そのため、ファイルの読み書きの標準 API や、この API を使った場合のプログラムの雛型、多くの研究者が共通に必要なとしそうな機能を実装したツールなどを提供する必要がある。

また、必須ではないが、次の点も考慮すべきと考える。

要請 5 必要最小限度の複雑さ

標準規格を普及させていくには、学習の容易さも重要なポイントとなる。一般に、様々な応用に配慮して記述能力を高くするほど、規格が複雑になり、規格の習得に労力を要するようになる。そのため、幅広い応用可能性や高い記述能力を維持しながら複雑さを抑えることが重要である。

*CrestMuseXML: Universal Data Format for Music Informatics —Overview and Fundamental Design Policy— by Tetsuro Kitahara, Mitsuyo Hashida, and Haruhiro Katayose (The CrestMuse Project, CREST, JST / Kwansai Gakuin University)

以上の要請に基づき、次の4つの基本方針を立てた。
方針1 レイヤーごとにXMLフォーマットを設計

あらゆる種類の音楽データを単一のXMLフォーマットで記述するのではなく、レイヤーごとに別々のXMLフォーマットを設計し、異なるXMLフォーマットの要素をXLink/XPointerでリンクする形をとる。この方法には、次にあげるメリットがある。第1に、同じデータにリンクしている別レイヤーの複数のデータを分析するといった応用が可能になる。たとえば同一楽曲の楽譜データにリンクしている様々な演奏家による演奏制御データを解析することで、演奏家の演奏の特徴を抽出するといったことである。第2に、レイヤーごとにデファクトスタンダードとなっているフォーマットがある場合には、それを変更なしに採用することができる。第3に、CrestMuseXMLに含まれる複数のXMLフォーマットのうち、自分の研究に関連するものだけを学習すればよいので、研究者の学習の労力を抑えることができる。

方針2 MusicXMLを一切の規格変更なしに採用

我々が扱う音楽データのうち、楽譜に相当する情報は、MusicXMLを用いて記述する。MusicXMLは、ノーションソフトFinaleの他、様々なソフトウェアでサポートされている。また、文献[7,8]などすでにMusicXMLをベースとした音楽アノテーション研究が行われており、これらとの連携も考慮できる。楽曲の構造分析結果や演奏制御パラメータといった楽譜に立脚した情報の記述は、このMusicXMLドキュメントのノードを参照する形で記述する。

方針3 XMLフォーマットの追加で仕様拡張を実現

CrestMuseXMLを単一のXMLフォーマットではなく複数のXMLフォーマットの集合ととらえているので、CrestMuseXMLの拡張は、CrestMuseXMLの枠組み内での新たなXMLフォーマットの追加という形で対応できる。

方針4 標準API・ツールの開発にJavaを採用

標準APIや標準ツールの開発にはJavaを採用する。Javaはプラットフォーム非依存なプログラム言語として広く使われており、特にXML関連のライブラリは非常に充実している。また、C++などに比べて言語仕様がシンプルなので、言語の習得も容易である。さらに、MATLABやMax/MSPでもJavaのプログラムを利用できるので、これらの言語からの利用も期待できる。

以下で、この基本方針に則って開発中のCrestMuseXMLおよびそのJava APIについて詳細を述べる。

3 CrestMuseXMLの設計

3.1 CrestMuseで扱う音楽データ

ここで扱う音楽データを次の4種類に分けて考える。

- notewise データ
楽譜上の個々の音符に対するデータである。楽譜

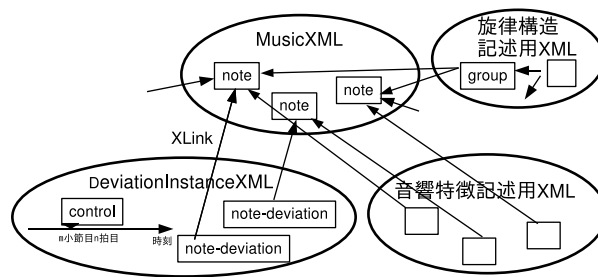


Fig. 1 CrestMuseXMLの全体像

の音符が持つ情報自体（音の高さ、音価など）や演奏制御データ（その音符を弾くときに楽譜通りの時刻からどのぐらいずらして弾くか、どのぐらいの強さで弾くか、など）の他、音符ごとの音響的特徴も含む。音響的特徴としては、HTC[9]で扱われている特徴量セットなどが考えられる。

- イベントデータ
時間軸上のある時点での出来事に関するデータである。代表例として、ピアノのペダルのオン・オフがあげられる。ピアノのペダリングのように特定の楽器パートに対するものとテンポの変化のように全楽器パートに対するものがある。
- 時系列データ
ある時間間隔（たとえば10ms）ごとに、基本的には楽曲の最初から最後まで n 次元ベクトルデータを並べたものである。これは、メル周波数ケプストラム係数（MFCC）のような音響特徴量の他、音響信号自体も該当する。これらはデータ量が極めて大きいため、どのように記述するかはまだ未定である。そのため、時系列データは今後の課題とし、本稿では扱わないこととする。MPEG-7のバイナリフォーマットなどを参考に、効率的な記述方法を検討していきたい。
- 構造データ
複数のデータ間の関係を記述したものである。関係記述の対象となるデータは、上記3種類のデータである場合もあれば、それ自体が構造データである場合もある。MOMI[7]で扱われているGTTMベースの旋律のグルーピング構造などが該当する。

3.2 CrestMuseXMLの全体像

現在のCrestMuseXMLの全体像をFig. 1に示す。notewiseデータに関しては、MusicXMLドキュメントのnoteタグをXLink/XPointerで参照する形で記述する。イベントデータについては、何小節めの何拍めかを属性などに記述することで時間軸上の位置を指定する。構造データは、MOMI[7]に準拠する予定である。構造の記述の対象となるデータは、notewiseデータと同様にXLink/XPointerで指定することになる見込みである。

CrestMuseXMLに含まれる個々のフォーマットは、ほとんどが検討中あるいは検討開始前であり、仕様は

確定していない。2008年8月に開催する演奏表情付けコンテスト Rencon で利用計画がすすんでいることから、2007年7月現在、演奏制御パラメータを記述するフォーマット DeviationInstanceXML の仕様検討を先行して行っている。その他の記述の仕様検討は、これがある程度終わってから着手する予定である。

3.3 MusicXML

MusicXML は Recordare が開発した楽譜記述用 XML フォーマットである。MusicXML には partwise と timewise の 2 種類の記述法がある。前者は、トップレベルタグの中にパート (part) 要素が 1 つ以上あり、各パート要素に小節 (measure) 要素があり、その中に音符 (note) などの要素があるという構造である。一方後者は、トップレベルタグの中に小節要素があり、その中にパート要素があるという構造になっている。partwise の方式の方がよく用いられ、我々の API でも当面は partwise のみの対応を予定している。note 要素は、notewise データが対象となる音符を指定するのに参照先として使われ、各 notewise データの記述における中心的な位置付けとなる。

3.4 DeviationInstanceXML

Deviation とは本来「逸脱」という意味であるが、音楽情報処理 (特に演奏表情付け) 分野では、「音楽的な表現のための演奏 (楽器制御) における楽譜からの意図的な逸脱」という意味で用いられることがある [10]。取り上げる楽器制御情報としては、設計の第 1 段階として、ピアノ演奏で用いられる制御情報を取り上げる。具体的には、各音符に対する発音時刻、消音時刻、音の強さ、テンポ、ペダルである。ピアノは、発音後は消音以外の音の制御が一切できず、発音や消音におけるパラメータも鍵盤を押したり離したりするときの強さだけであるので、比較的単純な仕様で見通しよく記述することができる。そのため、第 1 段階における検討材料として適している。

現行の DeviationInstanceXML は、non-partwise, partwise, notewise の 3 つのセクションに分けられる。このうち、notewise は文字どおり個々の音符に対する制御情報で、発音・消音時刻 (の楽譜通りの時刻からのずれ) と音の強さである。non-partwise と partwise はともに個々の音符に対するものではない制御情報で、前者は全パートに作用するもの、後者はパートごとに記述し、作用するものである。具体的には、前者はテンポコントロール、後者はペダリングなどが該当する。詳細は文献 [6] を参照されたい。

3.5 MIDI XML

MIDI XML は、Recordare が配布している MusicXML の DTD 群に付属している規格で、SMF をそのまま XML 化したものと考えればよい。CrestMuseXML では、SMF と 1 対 1 で対応する XML フォーマットとして MIDI XML を採用する。ただし、MIDI XML は 3.1 ~ 3.2 節で述べた考え方・方針とは完全に

は一致しない。

3.6 その他の XML フォーマット

旋律構造記述用は、前でも述べたように MOMI 準拠にする予定である。MOMI では、たとえばグルーピングの対象となる note 要素を XPointer で参照する形をとっており、同様の形になる見込みである。ただし、具体的な設計については未定である。

音響特徴記述用に関しては、HTC モデルパラメータ [9] のような個々の音符の情報を表すものについては、deviation データと同様に note 要素を参照する形で記述する。MFCC などのような個々の音符に対応しないもの (3.1 節でいう時系列データ) については、記述方法を検討中である。

その他、適宜記述フォーマットを増やしていく予定であるが、各ユーザが独自にフォーマットを増やすことも可能である。CrestMuseXML における各フォーマットは基本的に XLink/XPointer でリンクされているだけで、相互依存性は高くないので、個々のユーザが個別にフォーマットを増やしていても、既存のフォーマットに手を入れない限り競合はしない。そのため、各研究者や研究室で独自に新フォーマットを追加することは自由に行えるようにする予定である。そうして様々な研究者がそれぞれの立場から新フォーマットを作り、仕様がある程度固まった段階で、他の研究者にも有用なものであれば、CrestMuseXML の本体に導入し、仕様を充実していきたい。

4 Java API の設計

API 提供の目的は、様々な応用で共通な処理の流れや機能をあらかじめ実装しておくことで、個々の音楽システムの設計者の負担を軽減することにある。音楽システムは、インタラクティブシステムとバッチ型システムに大別できる。後者は多くの場合、指定されたファイルを読み、そこに含まれる音楽データに対して何らかの処理を行い、処理結果をファイルに書き出すという流れになる。そこで、現在のバージョンではバッチ型システムを念頭において、この一連の流れの雛型を提供する。これを次の基本方針で実現する。

- 個々のバッチ型システム (コマンドと呼ぶ) の共通基底クラスを用意し、コマンド作成で共通に必要な機能を提供する。
- 個々の XML ドキュメントやドキュメント内の個々の要素をラップするクラスを用意し、ドキュメントや要素へのアクセスをこれを通じて行う。

4.1 コマンドの基底クラス

コマンドの基底クラス CMXCommand が用意されており、コマンドはこれを継承して作成する (Fig. 2)。継承する際に、抽象メソッドとして宣言されている run をメソッドオーバーライドし、ファイル読み込み後の処理の内容を記述する。

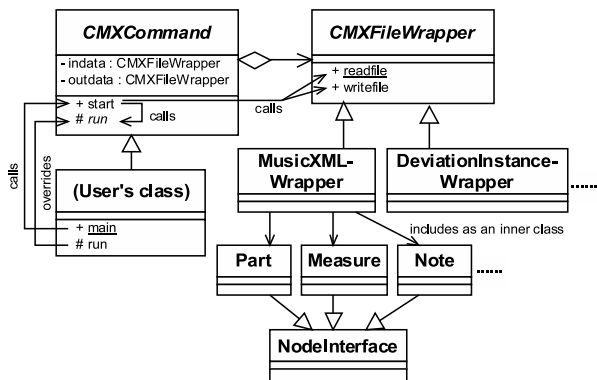


Fig. 2 本 API におけるおおまかなクラス図 .

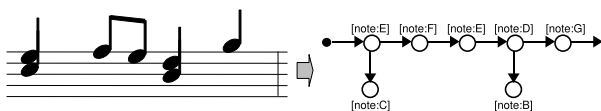


Fig. 3 二分木を用いたノートビュー . この二分木では、右の子へのパスが音符間の時間的な関係を表し、左へのパスは同時刻に発音する音符 (順不同) を表す .

4.2 ファイルラップとノードインターフェイス

XML ドキュメントをラップするクラスをファイルラップと呼び、基底クラス CMXFileWrapper のサブクラス (例: Fig. 2 の MusicXMLWrapper) として提供される . XML ドキュメント内の個々の要素をラップするクラスは、NodeInterface クラスのサブクラス (例: Fig. 2 の Part, Measure, Note) として提供される .

4.3 二分木を用いたノートビュー

ある音符に対して次に来る音符や同時に鳴る他の音符の情報が欲しい場合には、楽譜上の音符のならばを Fig. 3 のような二分木に見立て、二分木上をたどることで、上のようなことを実現する . このように音符の集合を二分木に収めたものをノートビューと呼ぶ . ノートビューは用途に合わせて複数のものが用意される . たとえば、全パートの音符をすべて1つの二分木に収めたものやパートごとに別々の二分木におさめたものなどである . 前者は、ある音符と同時に鳴る音符を他のパートも含めて取得する場合、後者は、ある音符に対してパート内での次の音符を取得する場合などに有用である . スラームノートビューを用いて管理する . あるスラーに対してスラーがかかった音符の集合を取得したり、ある音符に対してその音符を含むスラーを取得したりということがノートビューを介してできる .

5 ICMPC-Rencon '08

Rencon (Performance Rendering Contest) は、情緒豊かな音楽演奏を生成するシステムのための演奏コンテストを中心とした総合研究プロジェクトである . これまで国内外で6回のワークショップを行って

きた . 次回は2008年8月に札幌で行われる国際会議 ICMPC 10 の企画イベントとして開催される [11] .

この ICMPC-Rencon '08 では、楽譜データや演奏データの記述フォーマットとして CrestMuseXML が採用される予定になっており、本稿で述べた API のうち演奏生成にかかわる部分が Rencon Kit の一部として配布される予定である . CrestMuseXML の立場からは、最初の実地運用の機会を得たということになる . ここでの実地運用で得た経験を基に、フォーマットそのものや API 仕様はもちろん、開発体制の作り方、サンプルやドキュメントの整備方法など様々な部分を強化していく予定である . この開催に合わせて公開を予定している演奏表情データベース [12] にも CrestMuseXML が採用される見込みである . なお、Rencon 自体の情報は <http://www.renconmusic.org/> を参照されたい .

6 おわりに

本稿では、音楽情報処理研究で用いる様々な種類のデータを共通の枠組みで扱うための試みとして、CrestMuseXML とその Java API について述べた . 本プロジェクトでは、開発にご協力いただける方を募集している . いまのところ、XML フォーマットや API の仕様を議論するためのメーリングリストを運営中であるので、興味のある方は <http://www.crestmuse.jp/cmxml/> をご覧いただきご連絡いただきたい .

謝辞 ご討論いただいた長田典子氏 (関西学院大学)、野池賢二氏に感謝する .

参考文献

- [1] Manjunath, B. S., Salembier, P. and Sikora, T.: *Introduction of MPEG-7*, John Wiley & Sons Ltd. (2002).
- [2] Good, M.: *MusicXML: An Internet-Friendly Format for Sheet Music*, *XML 2001 Conf. Proc.* (2001).
- [3] Bellini, P. and Nesi, P.: WEDELMUSIC Format: An XML Music Notation Format for Emerging Applications, *Proc. WEDELMUSIC*, pp.79-86 (2001).
- [4] Roland, P.: *The Music Encoding Initiative (MEI)*, *Proc. MAX 2002*, 2002.
- [5] Billini, B. and Nesi, P.: Symbolic Music Representation in MPEG, *IEEE Multimedia*, pp.42-49 (Oct.-Dec. 2005).
- [6] 北原他: 音楽情報科学研究のための共通データフォーマットの確立を目指して, 情処研報, 2007-MUS-71 (2007).
- [7] 平田他: MOMI: 音楽メタ情報記述のためのフレームワーク, 情処研報, 2005-MUS-61, pp.13-17 (2005).
- [8] 梶, 長尾: 楽曲に対する多様な解釈を扱う音楽アノテーションシステム, 情処学論, 48, 1, pp.258-273 (2007).
- [9] 亀岡他: 調波時間構造化クラスタリング (HTC) による音楽音響特徴量の同時推定, 情処研報, 2005-MUS-61, pp.71-78 (2005).
- [10] 豊田他: 音楽解釈研究のための演奏 deviation データベースの作成, 情処研報, 2003-MUS-51, pp.65-70 (2003).
- [11] 橋田他: Rencon の現状報告と ICMPC-Rencon '08 の実施計画について, 情処研報, 2007-MUS-71 (2007).
- [12] 橋田他: 音楽演奏表情データベース構築に向けて, 人工知能学会全国大会, 1H1-4 (2007).