

CrestMuseXML Toolkitを用いた音楽情報処理システム

北原鉄朗 徳網亮輔 戸谷直之 片寄晴弘

(片寄グループ：関西学院大学工学部片寄研究室)

概要 これまで音楽情報処理の共通データ形式やフレームワークがなかったため、個々の音楽情報処理システムの実装に共通性がなく、組み合わせて再利用することは困難だった。本稿では、この現状を打破するために現在開発を進めている共通データ形式「CrestMuseXML」と汎用実装フレームワーク「CrestMuseXML Toolkit」を紹介する。

キーワード：音楽の共通データフォーマット、音楽情報処理フレームワーク、音楽の予測・推論。

1. はじめに

いまここに2人の研究者がいるとしよう。研究者Aは音響信号処理に長けており、ピアノ演奏の音響信号を楽譜に起こすシステムを開発した。研究者Bは膨大な楽譜データに対する分析に基づいた自動編曲システムの研究をしている。現在はMIDIファイルやMusicXMLを用いているが、WAVファイルが利用できればと日頃から思っていた。そんなときにAの研究発表を聴き、これを利用したいと思って早速プログラムをもらってきた。さて、うまくいくだろうか。

このケースでは、Aのシステムの出力形式とBのシステムの入力形式が揃っていれば利用できる。もしも揃っていない場合は変換が必要になる。揃っていたとしても、楽譜への変換部も含めた単体アプリとして実装したり、Aのシステムの一部をいじろうと思ったら簡単にはいかない。

では、次の場合はどうか。別の研究者CもAの技術レベルの高さに注目していた。しかし楽譜化そのものにはあまり興味はなく、この技術を歌声の分析・合成に活かせるのではないかと考えていた。同様に、プログラムをもらってきて歌声への応用に取りかかった。さて、うまくいくだろうか。

この場合、システム自体に興味があるわけではないので、Aのプログラムをいじらなければならない。一般に他人が作ったプログラムのソースコードを読んで理解するのは非常に労力のいる作業で、このプログラムから自分の必要な部分、たとえば周波数解析や調波構造抽出などの部分を取り出して自分で使えるように作り替えるのは至難の技である。これまで多くの研究者が同じような場面に遭遇し、「これなら一から自分で作った方が早いや」と諦めてしまった経験があるだろう。

このように研究が進展し、高度な技術が開発され、様々な論文が発表されても、技術の再利用は容易ではない。技術の再利用はここでは即ちソフトウェアの再利用なわけだが、ソフトウェアの再利用には2つの立場が考えられる。1つはソフトウェアを単に使用する立場である。これをスムーズに行うにはデータの入出力形式が共通化されている必要がある。しかし現状では、WAVや標準MIDIファイルなどの少数を除いて標準的なデータ形式

が存在せず、研究者が自身の都合に合わせて独自の形式を用いていることが多い。もう1つはソフトウェアを修正・改変して利用する立場である。この場合、ソフトウェアを構成するモジュールの単位で可搬性が要求される。モジュール単位での可搬性をサポートするフレームワークがあれば、モジュールの組換えによって様々なアイデアをスピーディに試すことができる。

本プロジェクトでは、こういった考えの下、音楽の共通データ入出力形式「CrestMuseXML」および音楽情報処理の汎用実装フレームワーク「CrestMuseXML Toolkit」の開発を進めている。本稿では、これらの概要を簡単に紹介する。詳細は文献[1], [2]あるいは<http://www.crestmuse.jp/cmxml/>をご覧ください。また、さらなる展開として現在検討を進めている音楽の統一的推論アーキテクチャについても触れる。

2. CrestMuseXML (CMX)

2.1 CrestMuseXML

CrestMuseXMLは、音楽の様々なデータを記述するためのXMLフォーマット群の総称である。音楽は、波形レベルから楽譜レベル、MIDIレベル、認知構造レベルまで、様々な抽象度の記述が可能である。しかも、認知構造のような高次の表現は、しばしば解釈者によって違いが出てくるため、同一の楽譜データに対して複数の認知構造表現を付与するといった、「1対多」の記述が求められる。そこで、我々は、あらゆる種類の音楽データを表現できる巨大なXMLフォーマットを作るのではなく、抽象度ごとにXMLフォーマットを用意し、異なるXMLフォーマットに記述された要素間の対応をXLinkを使ってハイパーリンクとして表現する、という方針をとった(図1)。それにより、

- 個々のフォーマットをシンプルに保った上で、複数のフォーマットを組み合わせることで、幅広い種類の音楽記述が可能、
- XLinkは「1対多」のハイパーリンクが可能であるので、同じデータ(たとえば楽譜や演奏データ)に対して複数のメタデータ(楽譜や演奏に対する認知構造的解釈)を付与することが可能、
- ユーザが自分で設計したXMLフォーマットを追

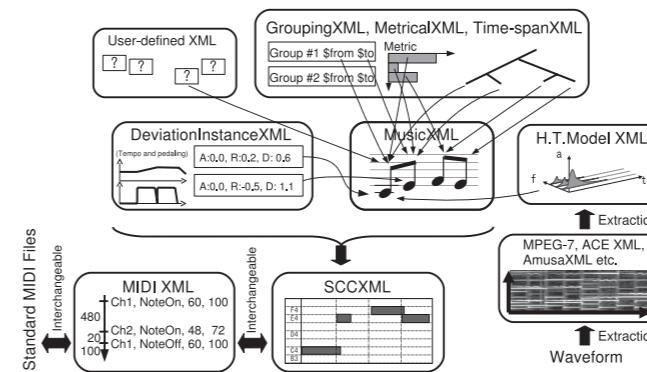


図1 CrestMuseXMLの全体像

加することで、記述能力の拡張が可能、
• フォーマットごとにデファクトスタンダードがある場合は、それをそのまま採用可能、
といったメリットが得ることができる。

2.2 CrestMuseXML Toolkit

CrestMuseXML Toolkit(以下、CMX Toolkit)は音楽情報処理システムを実装するための汎用フレームワークである。元々は、何らかのXMLファイルを読み込んで任意の処理をした後、結果をXML形式で出力するというオフライン型のシステムを対象としていたが、2008年6月現在で開発中のver. 0.40ではAPIを拡張し、オンライン処理もできるようにしている。

可搬性の高いオンライン処理モデルとして、Kahn Process Network [3]を採用した。Kahn Process Networkは信号処理モデルの1種で、プロセスと呼ばれる処理モジュールのネットワークとして信号処理システムを構成する(図2)。各プロセスは、キューから要素を1つ取り出して何らかの処理を行い、その結果を別のキューに追加する、という共通のAPIを持つ。複数のプロセスがキューを介してつながることで、あるプロセスでの処理結果が次々と後段のプロセスへ送られ、処理が進められる。そのため、並行処理に適したモデルといえる。また、各プロセスが共通のAPIを持つため、プロセスの再利用や組換えが容易である。

2.2 CrestMuseXML Toolkitを用いた音楽情報処理システムのデモ

我々は現在、予測型自動伴奏システムの開発を進めている。これは通常の伴奏システムとは異なり、ユーザが演奏する楽曲はあらかじめ決められていない。ユーザが演奏(即興演奏含む)すると、システムは演奏の次の展開を予測し、それに合う伴奏(コード)を生成する。いわゆる「流しミュージシャン」は初めて聴く歌に対しても楽曲がどう展開していくかを巧みに予測して伴奏をすることができる。予測型自動伴奏システムは、これを計算機にさせようという試みである。

このシステムをCMX Toolkitを用いて実装中である。MIDIキーボードからMIDIメッセージを受け取ると、その情報がKahn Process Networkに送られる。するとそれを受け取った予測・推論プロセス

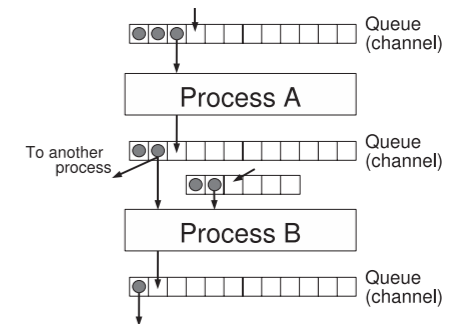


図2 Kahn Process Network

現する。この推論処理によって次の小節のコードが決まったらその情報がKahn Process Networkに戻される。伴奏生成部が、小節が変わるタイミングでその情報を受け取り、指定されたコードによる伴奏を生成する。

3. 音楽推論アーキテクチャへの展開

音楽の楽しさが、その予測可能性とそこからの逸脱のバランスにある[4]と考えれば、前節で触れた音楽データに対する予測機構は音楽情報処理全体において本質的な意味を持つ。そこで、我々は「予測」をキーワードに統一的な音楽推論アーキテクチャの設計を計画している。

人間が聴くのはMIDIのようなシンボル化された情報ではなく音響信号なので、この推論アーキテクチャでも音響信号を扱わなければならない。そのベースとなるのはAuditory Scene Analysis[5]である。音脈分凝の計算モデルを予測の観点から設計し、記号化された表現に還元する。それにさらに高次の予測が適用され、高次の構造が推定される。予測には、プリミティブな予測から短期記憶による予測、長期記憶による予測などが考えられ、それぞれに合った情報処理機構を設計する。

このようにして設計された音楽推論アーキテクチャにおいて、どのノードの情報が外部から与えられ、どのノードの情報を推論するかを切り替えることで、旋律予測、和声付け、旋律の類似度計算など様々なタスクに対応できる。今後、こういった方向に研究を進めていきたい。

参考文献

- [1] 北原鉄朗, 橋田光代, 片寄晴弘: 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, 2006-MUS-66, pp.149-154 (2007).
- [2] 北原鉄朗, 片寄晴弘: CrestMuseXML (CMX) Toolkit ver. 0.40について, 情処研報, 2008-HCI-128, 2008-MUS-75, pp.95-100 (2008).
- [3] G. Kahn: The Semantics of A Simple Language for Parallel Programming, Inform. Process., pp.417-475 (1974).
- [4] http://impbrain.shimojo.jst.go.jp/jpn/about_jpn.html
- [5] A. S. Bregman: Auditory Scene Analysis, MIT Press (1990).