

演繹オブジェクト指向データベースを用いた演奏生成アプリケーションオントロジー

Application Ontology for Performance Rendering based on a Deductive Object-Oriented Database Approach

平田 圭二*¹ 平賀 瑠美*²
Keiji Hirata Rumi Hiraga

- * 1 NTT コミュニケーション科学基礎研究所
NTT Communication Science Basic Research Laboratory, 3-1 Morinosato-Wakamiya, Atsugi 243-0198, Japan
- * 2 筑波技術短期大学
Tsukuba College of Technology, 4-3-15 Amakubo, Tsukuba 305-0005, Japan

1999年4月5日 受理

Keywords: musical domain ontology, performance rendering, performance application ontology, deductive object-oriented database

Summary

This paper discusses the roles of a musical ontology for performance rendering application. Basically, performance rendering consists of two stages: score analysis and the application of a performance model to the results of the score analysis. However, few attempts have been successful at programming these two stages to realize natural and expressive performance. This is because the characteristics of music are tacit, incomplete, partial, and subjective, and accordingly, almost all conventional systems employed ad hoc methods with heuristics. While a few exceptional systems work well, they do so in very limited musical genres.

We introduce a musical ontology to build a performance rendering system. Since the musical ontology allows us to treat music in a formal and explicit way, it will enable musical concepts to be real entities in a computer program. With the musical ontology, we can elaborate cause-and-effect relationships among the score analysis, performance model, and system output.

In our paper, we will first formalize the problem of performance rendering and survey the current situation of musical ontology research. Then, we will describe the application ontology for performance rendering, which is based on a deductive object-oriented database technique.

1. はじめに

本論文では、人間が行うような自然な音楽演奏を生成するシステムに関して、音楽オントロジーの果たすべき役割について考察し、従来の演奏生成システムが抱えていた問題点を洗い出し、音楽オントロジー導入による解決法について述べる。

音楽は、人間の知的活動の所産として自然発生し、長い年月を経て発展し、現在ではある範囲で一定の約束事や習慣を確立するに至っている。しかし、音楽

に関する約束事は暗黙かつ曖昧な形で人間のコミュニティに根付いており、音楽に関する概念は、通常、個人、時代、地域、ジャンルなどによって大きく異なる。人間どうしのコミュニケーションでは、このように暗黙で曖昧な約束事でも特に問題は生じていなかった。しかし、楽曲や演奏という音楽的なデータや知識を計算機で取り扱う場合には、その形式的な定義が必要になる。計算機上で音楽的なデータや知識を処理する技術を研究対象とする分野は、音楽情報処理と総称されている [Roads 96, bit 98]。音楽情報処理の中心的課題は、この曖昧で暗黙的な約束事の上に成り立っている

る音楽的なデータや知識に合理的な形式的定義を与え、それに関する十分制御された計算を実現することである [Dannenberg 93].

音楽的なデータや知識が持つ以下のような特徴が、形式的な定義や制御された計算の実現にとって問題となる: (1) 暗黙的, (2) 曖昧, 部分的, (3) 主観的. これらはいわゆる計算機科学的な情報の持つ特徴とは対照的である.

(1) の暗黙的とは、内省などを通してそのような概念が存在していることは認識できるが、記号や構造として記述するのが困難なことを指す. 例として、楽曲のある個所に $G_7 \rightarrow C_{maj7}$ という和音進行が現われている場合を考える. この和音進行だけなら機能と和声的に理解することも可能だが、前後の和音進行との関係、曲全体においてその和音進行の果たす役割などを還元的に厳密に述べるのは困難である.

(1) に関連して、ある音楽的なデータや知識を記述しようとした場合、どうしても記述できない部分が残ってしまうと、その記述は (2) のように曖昧、部分的にならざるを得ない. 実際、この場合、比喩を用いたり (従って語彙はほぼオープンである)、人間の認知認識能力や文脈に依存して相対的、定性的に記述する. 前者の例には「暖かい演奏」という言い方などがある. 後者の例として、ウイナワルツの三拍子を自然に演奏するために、「二拍目のタイミングを少し早める」と言う場合などがある.

また、[Balaban 96, Dannenberg 93] で指摘されているように、部分的なデータや知識から出発し、徐々に拡張して大規模なシステムに容易に移行できるような枠組が望ましい.

音楽は知的活動の一部であることを考えると、上の (1), (2) の特徴が人工知能 (AI) で扱われている知性の特徴に共通しているのは当然であろう.

このように AI が扱う問題との共通性がある一方で、音楽は (3) のような特徴も持っているため、これまでの音楽情報処理研究では、芸術性、感性、個性等に注目した研究事例が多かった. しかし、評価が主観に左右されると、音楽データや知識が合理的に定義されているのか、それに関する計算は十分制御されているのかを判断するための絶対的な尺度を見い出すのが大変難しい. チェス、将棋、囲碁のように、実行や勝敗決定の規則が定められており、優劣の決定が明確な処理とは対照的である.

現在我々は音楽演奏を生成するシステムに取り組んでいるが、上述した音楽データや知識の 3 つの特徴のため、自然で表情のある演奏を生成することはそれ

ほど容易ではない. 我々は、音楽のドメインオントロジー及び演奏のアプリケーションオントロジー構築により、この問題解決を試みる.

本論文では、音楽システム構築に影響を与える 3 つの音楽データや知識の特徴について述べ、以下、2 章では音楽演奏を生成するシステムに関して、その問題の定式化、従来の研究事例と問題点を述べる. 3 章では Balaban の Object-Oriented Music Piece (OOMP) 及び演繹オブジェクト指向データベース (DOOD) による音楽オントロジーについて述べ、音楽的な概念の記述法について議論する. 4 章では、DOOD による演奏のアプリケーションオントロジー記述を提案する. 5 章では OOMP と DOOD の比較などを議論する.

2. 演奏生成システム

2.1 問題の定式化

音楽演奏を生成するシステムとは、五線譜などの楽譜を入力とし、図 1 のようなプロセスで、人間が行うような自然で表情のある演奏を出力するシステムである.

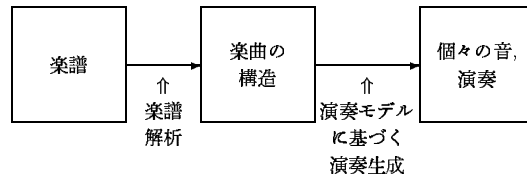


図1 人間のような演奏を生成する処理プロセス

図1 で演奏生成のための入力としては五線譜が用いられることが多く、一般に多数の発想記号や標語*1が含まれる. ピアノ曲に限ると、五線譜上の各音符の音高はそのまま解釈して良いだろう. しかし、音の長さ (音価) や大きさ (音量) は、演奏者や曲の解釈などによって大きく異なる.

楽譜解析 (図1) では、楽譜に記述された楽曲が、旋律、リズム、和声に関してどのような構造を持っているかを解析し、楽曲中の各音はその構造の中でどのような機能を持っているかを決定する. この解析された構造により、音楽的に意味をもつ事象や概念は明示的に表現され、楽譜に現われる事象やその間の因果関係を形式的、定性的、定量的に述べることができる.

*1 楽曲の表情や表現方法の指示. 発想記号には、テンポの指示 (リタルダンド、アレグロなど)、強弱の指示 (フォルテ、クレッシェンドなど)、アーティキュレーション (音のつながりかたの指示のためのスラーなど) などがある. 発想標語は、例えば "vivace" (生き生きと) など、発想記号より主観的な表現を用いる.

一般に、楽譜構造解析には多義性があり、さらに 1 つの楽曲は複数の構造の重畳として多面的に理解される。

演奏モデルに基づく演奏生成 (図 1) では、解析された楽曲構造に対応する演奏のモデルを用いて、実際の各音の発音タイミングと音量を定量的に決定する。演奏モデルは、ある楽曲構造中における音符の出現場所の情報を、具体的な演奏パラメータ (ある音符の音高、音量、開始/終了のタイミング) の制御にマッピングする*2。1 つの音符の演奏パラメータは関係するすべての演奏モデルの影響を受けるので、それらの影響を最終的な演奏パラメータとして合理的に組み合わせなければならない。

このように、楽譜解析の処理では、音楽学、音楽心理学、音楽認知科学に関する多種多様な音楽知識を解析及び獲得し、演奏生成に必要な知識として体系化しておく必要がある。演奏生成の処理では、個々の音の音量や発音タイミングという具体的な数値を計算する必要がある。しかし、上述した音楽データや知識の 3 つの特徴が、これらの処理を困難にしている。

2.2 研究事例

これまで、多くの演奏生成システムの研究が行われている [平賀 98]。それらは楽譜解析の手法に関して大きく 2 群に分けることができる。

- A** 音符、発想記号等、楽譜上の表層的なパターンに基づき各音の演奏パラメータを制御するシステム
B 現在最も精密であると考えられている音楽理論に基づく解析を利用したシステム

これらは、楽曲の表層的な構造を実際の音の発音パラメータと関連づければ十分であるという考え方 (A) と、自然な演奏を生成するためには楽曲の構造をより深く理解し解析する必要がある [Balaban 96, Chafe 97, Honing 92] という考え方 (B) に対応する。A, B に属する研究事例 [Friberg 91, 彌富 97, 青野 97, Widmer 93, Arcos 97] を表 1 にまとめる。

ここで、現在最も精密であると考えられている音楽理論とは、Generative Theory of Tonal Music (GTTM)[Lerdahl 83] 及び Narmour の理論 [Narmour 90] を指す。GTTM は楽曲に現われる旋律やリズムの階層構造を解析するのに有用であり、Narmour の理論は、旋律の持つ進行感 (暗意 - 実現, implication-realization と呼ばれる) を解析するのに有用である。

*2 楽器固有の奏法 (例えばピアノならペダルの使用など) も考慮しないと演奏としては不十分である。しかし、現在殆どどのシステムはそのような情報まで扱えない。

表 1 演奏生成システムの研究事例

A	Friberg	彌富ら	青野ら
楽譜解析	音符、発想記号等の譜面上の表層的な情報をそのままパターンとして使う		
演奏モデル	アプリオリな数式	アプリオリな数式	重回帰分析によるパラメータフィッティングにより得られた一次式
B	Widmer	Arcos et al	
楽譜解析	GTTM 及び 暗意 - 実現 (I-R) モデル		
演奏モデル	定性推論に基づく線形方程式のパラメータ学習	I-R の情報を付加した演奏事例による CBR	

紙面の制約上、各システムの詳細については省略するが、いずれのシステムも、演奏対象の曲目やジャンルを限定し、評価/改善のプロセスを何度も繰り返して、あるレベル以上の演奏結果を得ている。しかし、未知曲にも対応できるほど一般性のあるシステムは開発されていない。

これら研究事例の楽譜解析や演奏モデルにおいて、暗黙的な或は曖昧、部分的な音楽データや知識が出現した場合の対応法は主に 2 通りある。1 つは恣意的で一般性の無いデータや知識で補う方法であり、もう 1 つはそのようなデータや知識は考慮しないようにする方法である。いずれの対応法でも、得られた知見やルールの殆んどは与えられた楽曲のジャンルや作曲家等に強く依存しており適用範囲は狭い。従って、特定の状況だけでのみ適切に動作し、その他多くの状況では不自然な演奏を生成する。これも 1 章の音楽データや知識の 3 つの特徴に起因するものである。

3. 音楽オントロジーの現状

これまで提案されてきた音楽知識表現手法の多くは、制約、属性文法、ニューラルネット、オブジェクト指向モデルなどの汎用的な手法に基づいている。しかし、1 章で挙げた音楽的なデータや知識の 3 つの特徴全てに対応できているものは殆んど無いと言ってよいだろう。音楽オントロジーとは、これら 3 つの特徴に対応できるようなものでなければならない。

これまで音楽オントロジーを志向した研究は少なく、Balaban の Object Oriented Music Pieces (OOMP) [Balaban 96] だけであると思われる。

本章では、音楽のドメインオントロジーの先行研究である OOMP と、本論文で提案する演繹オブジェクト指向データベース (DOOD) に基づく音楽オントロジーについて述べる。

Heijst らは、オントロジーを 2 つの観点 (概念化の構造に関する量と型及び概念化の対象) から分類でき

ると述べている [Heijst 97]. OOMP は前者の観点からは知識モデルオントロジーと見なせ、後者の観点からは音楽のドメインオントロジー (の一部) と見なすことができる. DOOD はオントロジー記述の枠組でありオントロジーそのものではないが, OOMP とほぼ同様に分類できるだろう.

3・1 Balaban の OOMP

OOMP は音楽オントロジー記述の枠組である. まず, Structured Music Pieces (SMP) という基本データ構造によって, 楽譜上に書かれた楽曲の持つ階層関係と時間の関係だけを表現する. SMP に対して, 属性 (例えば dynamics, atmosphere, style など) やメソッド (例えば delays など) を付加し, 楽譜上に書かれていない音楽的性質を明示的に表現する. その音楽的性質は SMP 間の階層関係や時間の関係に沿って伝搬する. これら以外の SMP 間の関係としては, component や elaboration などがアプリアリに与えられている.

非常に簡単な SMP の例として, “かえるのうたがーきこえてくるよー” という童謡の一部 (*ms* と呼ぶ) を 2 つのパートで輪唱する場合を考える. “かえるのうたがーきこえてくるよーきこえてくるよー” *³ は *ms* を用いて *ms*@0 | *ms*@duration(*ms*/2) のように記述できる. これは, まず *ms* を時刻 0 で演奏し, *ms* 全体に要する時間の半分が経過したら, 次の *ms* を演奏する, ということを意味している. SMP の構成子には @ (開始時刻), | (並置) の他に - (連結) がある. この SMP に属性やメソッドを付加すると, 例えば *ms*(*style:jazz, delays(mp):{0, duration(mp)/2}*) のようになる. これは, *ms* に関して, スタイルはジャズで, 遅延 0 と遅延 *duration(mp)/2* の 2 つの場合があることを表わしている. SMP 間を伝搬する音楽的性質は, その種類によって伝搬方向が異なる.

OOMP は, 音楽データや知識の 3 つの特徴に関して, 以下のように対処できると主張している.

暗黙的: 例えば *ms*₁ と *ms*₂ は何らかの関係にはあるが, その関係を明示的に記述できないような場合, *ms*₁ = *tacit*(*ms*₂) のような新たな関数 *tacit* を導入する. これにより, *ms*₁ と *ms*₂ 間の暗黙的な関係が表現できるとしている.

曖昧, 部分的: 一般に, 1 つの楽譜は何通りもの SMP で表現することができ, その異なる各 SMP が別々の楽譜構造解析結果を表現していると考えられる. これが, 前述した楽譜構造解析の多義性に対

*3 1 つめの “きこえてくるよー” と他パートの “かえるのうたがー” が重なる.

応しており, 曖昧性に対処できるとしている.

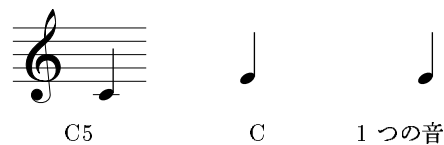
さらに, SMP では下位 SMP やパラメータの現れる場所に変数を置くことができる. 例えば, 下位階層のある要素の値が不明な場合, 変数によって部分的, あるいは不完全な楽曲構造にも対処できるとしている.

3・2 DOOD による音楽知識の記述

本節では DOOD に基づく音楽知識の記述について述べる. DOOD [Hirata 96] の利点の 1 つは, DOOD のオブジェクト項と包摂関係 (半順序関係) によって, 楽譜, 教科書の知識, 楽譜解析の結果などの異なる抽象度に属する音楽的実体と関係が統一的に表現, 操作 (計算) できることである.

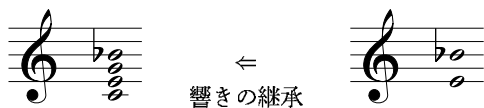
音楽情報処理には様々な抽象度を持った音楽的概念が現われる. 例えば, 部分的な楽譜, 音価が省略された楽譜, 重要な音だけを抜き出して作ったメロディ, テンションが省略されたコード進行, 和音中の重要な構成音などである.

今, C5, C, 単なる 1 音という 3 種類の音を考える. C5 というのは, ピアノの 88 鍵盤の中のある特定の鍵やギターの手があるポジションの時に出て来る音を指す. C5 に音色や演奏する楽器の情報は含まれていない. C というのは, オクターブを無視した C の音の集まりとか, 「ト長調のスケールは G A B C … から成る」と言った時の C であったり, 「A マイナという和音は A C E から成る」と言った時の C などである. ここで音の長さや音の絶対的な高さ (周波数) などの情報は含まれていない. さらに, 単なる 1 音というのは, C や F などのいわゆる音を総称する概念である.



この時, C5, C, 単なる 1 音というオブジェクトは, 種類は同じだがその意図する所は異なっており, C5, C, 単なる 1 音という順に抽象的になって行くと考えるのは自然である. この時, C5 と C, C と単なる 1 音の間関係としては, 上位 / 下位, is-a, クラス / サブクラスなどが割り当てられる.

次の例では C, E, G, Bb から成る和音を考える. この和音はトライトーン (E, Bb の減五度) を含むので, ドミナント 7th の響きを持つ.



ここで、和音 C, E, G, Bb がトライトーン E, Bb の性質を継承していることを表現するためには、和音 C, E, G, Bb をトライトーン E, Bb のサブクラスであると考えれば良い。

これらの音楽的概念を DOOD のオブジェクト項と包摂関係で表現する。例えば 5 番めのオクターブでピッチが C という音は、`note(pitch=C, octave=5)` というオブジェクト項で表現される*4。G7 という和音名は `chord_symbol(root=G, name=7)` のように表現できる。順序のないモノの集まりを表現する時には集合を用い、例えば C, E, G から成る和音は {C, E, G} と表現する。次に、音楽オブジェクト間の抽象化 (具体化) の関係は包摂関係 (subsumption relation, \sqsubseteq) で表現する。包摂関係は、直感的には「具体的なオブジェクト \sqsubseteq 抽象的なオブジェクト」、「特殊 \sqsubseteq 一般」を意味する。上の例で `note(pitch=C, octave=5)` と `note(pitch=C)` を比較すると、`note(pitch=C, octave=5)` の方が `octave=5` という属性の分だけ記述が具体的なので、`note(pitch=C, octave=5) \sqsubseteq note(pitch=C)` となる。同様に `note(pitch=C) \sqsubseteq note` となる。和音 C, E, G, Bb と和音 E, Bb の場合は、{C, E, G, Bb} が {E, Bb} のサブクラスなので、{C, E, G, Bb} \sqsubseteq {E, Bb} となる。そして、これらオブジェクト間の包摂関係は演繹規則により定義される。

オブジェクトの識別に影響を及ぼす属性は固有属性と呼ばれ、包摂関係を含むオブジェクト間のあらゆる関係は固有属性に基づいて定義される。上の例に出てきた属性は全て固有属性である。

4. 演奏生成のためのアプリケーションオントロジー

3・1 節と 3・2 節で触れた音楽オントロジーは、現在、音楽知識やタスクの一部分を記述しているに過ぎない。しかし、特定のアプリケーション (ここでは演奏の生成、表情付け) に限定すれば、アプリケーションオントロジーを構築することができる。

演奏アプリケーションオントロジーとは、楽曲の構造や演奏ルールなどの事実に関する制約や関係の概

念化である。我々は、既存の音楽理論 (GTTM) を DOOD の枠組で表現し、演奏アプリケーションオントロジーを構築する。これより、演奏生成システム実現において、音楽データや知識の持つ 3 つの特徴に対処できると考える。

4・1 演奏生成のタスクモデル

演奏生成のタスクは様々な抽象レベルで記述される。例えば、上昇する音列の自然な演奏を生成したい時、市販の自動演奏器 (シーケンサ) の場合なら、「音列の個々の音の長さや強さを、与式に従って変化させよ」と指示することができる。一方、人間どうしの会話ならば、単に「この音列を自然に聴こえるように弾け」と言うだけで良いだろう。

前者の市販シーケンサを操作するような抽象レベルであれば、比較的曖昧性が少ないので、例えば、Friberg の演奏表情付けタスク [Friberg 91] の GMI 1C 規則 (Faster Uphill, 先行する音が低く後続の音が高い場合、音価を $2 \times k$ ミリ秒短くせよ。ここで k は演奏規則に関するグローバル変数である) を用いることができる。これより、タスクモデルを導き、このタスクモデルをマップすべきアプリケーションオントロジーを構築する。このアプリケーションオントロジーに含まれるべき概念は音符だけであり、関係は「先行する」「後続する」の 2 種類である。

一方、後者の人間に指示するようなタスクでは、表現が暗黙的で曖昧性が大きい。しかしアプリケーションオントロジーを用いることで、暗黙的な情報を含んだ指示を理解し、曖昧さをある程度解消することができる。例として、2 章に示した演奏生成システムのうち、B の GTTM [Lerdahl 83] に基づいてタスクモデルを構築する場合を考える。この時、人間どうしの「この音列を自然に聴こえるように弾け」という指示は、「ある楽曲構造内でのテンポは、その構造中の最も顕著な音 (prominent note, pn) に向けて次第に速くなり、pn から構造の終りに向けて次第に遅くなる。音量も、pn に向けて増大し (クレッシェンド)、構造の終りに向けて減衰する (デクレッシェンド)」と換言される。これは、GTTM に基づいて楽曲の構造や曲の流れを考慮したことで、「自然に聴こえる」という言い方に含まれる暗黙的な情報を明示的に表現でき、曖昧性が減少したと見なすことができる。

ここでは、GTTM のタイムスパン還元*5に基づくタスクモデルを採用する。

*4 この時、C5 という基本オブジェクトのみで C5 の音を表現することも可能であり、どちらの表現を選択するかはユーザの自由 (主観依存) である。

*5 楽曲を階層的な時間間隔 (タイムスパン) に分割し、各タイムスパンを構造的に重要な音 (prominent note) に簡約化すること。

4・2 DOOD による演奏アプリケーションオン トロジー記述

演奏アプリケーションオンロジー構築のため、タイムスパン還元に見られる音楽的概念や制約を、DOOD で表現したオブジェクトやオブジェクト間の関係にマップする。

タイムスパン還元を表現するために、次のようなオブジェクト項を導入する。

$\text{timespan}(\text{pn} = \dots, \text{pp} = \dots)$
/($\text{play} = \{\text{express}_1(\dots), \text{express}_2(\dots), \dots\}$)

ここで pp は polyphony (多声メロディ, 五線譜に記述された楽曲の一部) の略である. pp 属性値の一般形は $c_1 \rightarrow c_2 \rightarrow c_3 \rightarrow \dots$ のようになっており, 時間軸上の楽曲の進行を表す. c_i は timespan オブジェクトである. 例として, 図2 (b) にバッハのマタイ受難曲 BWV244 の譜面を, 図2 (a) に GTTM のタイムスパン還元によるその解釈結果 [Lerdahl 83] を DOOD のオブジェクトで表現した様子を示す. 図中, 台形が timespan オブジェクトを表わし, 実線矢印は pp 属性値に含まれる参照を表わし, 点線は譜面上の場所とオブジェクト階層の対応を表わす. timespan オブジェクト項間に成り立つ包摂関係が, 楽曲の「部分-部分」や「部分-全体」という関係に対応する.

‘/’ 構成子は, その左辺にオブジェクト識別子として機能する項を記述し, その右辺に非固有属性というオブジェクトの識別に影響を及ぼさない属性を記述する. 非固有属性は, あるオブジェクトにとっての付加的, 補足的な属性と見なすこともできる. express オブジェクトはその属性値として演奏のパラメータを含んでいる. 一般に 1 つの楽曲は複数の構造の重畳として多面的に理解されるので (2・1 節), 1 つの timespan に対して複数の表情付けが重畳する可能性がある. 従って, play 属性の値は express_i オブジェクトの集合となっている. 図2 (a) の timespan オブジェクト毎に play 属性が設定されている. さらに, 楽曲全体の表情付けは, これら階層を成す各 timespan の表情付け情報を組み合わせたものになる. この組み合わせ方も各 timespan の属性として表現される.

また, 我々の枠組ではオブジェクト項の型が概念階層に対応する. つまり, 属性ラベルが概念階層のアーキ名に, 属性値のオブジェクトがノードに対応する (図3). 図中, 実線は固有属性, 破線は非固有属性を表し, set-of はその属性値が集合であることを表わす.

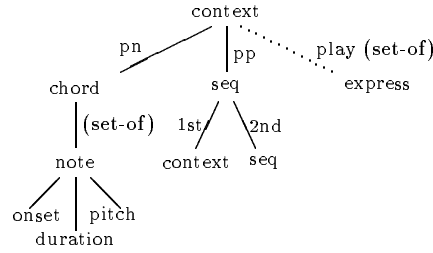


図3 タイムスパン還元のための概念階層

4・3 非固有属性による主観的な知識の表現

あるユーザ (聴取者) にとって, どのようなテンポ緩急や音量増減が好ましいかはユーザの主観に大きく依存する. よって, 演奏パラメータは, 楽曲の構造や演奏ルールなどに関する事実というよりは, タスク固有の拡張部分と見なすのが適当であろう.

DOOD による演奏アプリケーションオンロジーでは, この演奏パラメータのようにユーザが主観的と判断した情報は非固有属性を用いて記述するので, 演奏アプリケーションオンロジーの事実に関する部分に影響を与えないよう分離して記述することができる. これより, ユーザの趣向に依存する振る舞いにも柔軟に対応できるようになる.

5. 考 察

5・1 OOMP と DOOD

1章 で挙げた音楽的なデータや知識の 3 つの特徴の内, (1) 暗黙的と (2) 曖昧, 部分的に関しては, 3・1 節 と 3・2 節, 4・1 節 で述べたように, OOMP も DOOD もある程度対応していると考えられる. DOOD と SMP はともに素性構造に似た仕組みを持っているので, (1) や (2) に関する記述力が大きく異なることはないだろう.

OOMP の枠組では, 楽曲の階層構造と時間関係だけを SMP で表現し, それ以外の情報は SMP に対する属性, メソッド, SMP 間の関係として付加的に記述する. しかし, この切り分けをア priori に与えたことが OOMP の柔軟性を損なっていると考えられる. 何故なら, 演奏生成のタスク (4・2 節) を実現するには, 3・2 節 で述べたように, 様々な種類やレベルの音楽データや知識を取り扱う必要があるからである. 例えば, 楽譜では表現できない知識がある. 演奏を聴取することで得られるデータや知識がある. さらに, 主観的な情報もある. 演奏アプリケーションオンロジーを構築するためには, これら音楽データや知識を全てを表現し関連付けなければならない.

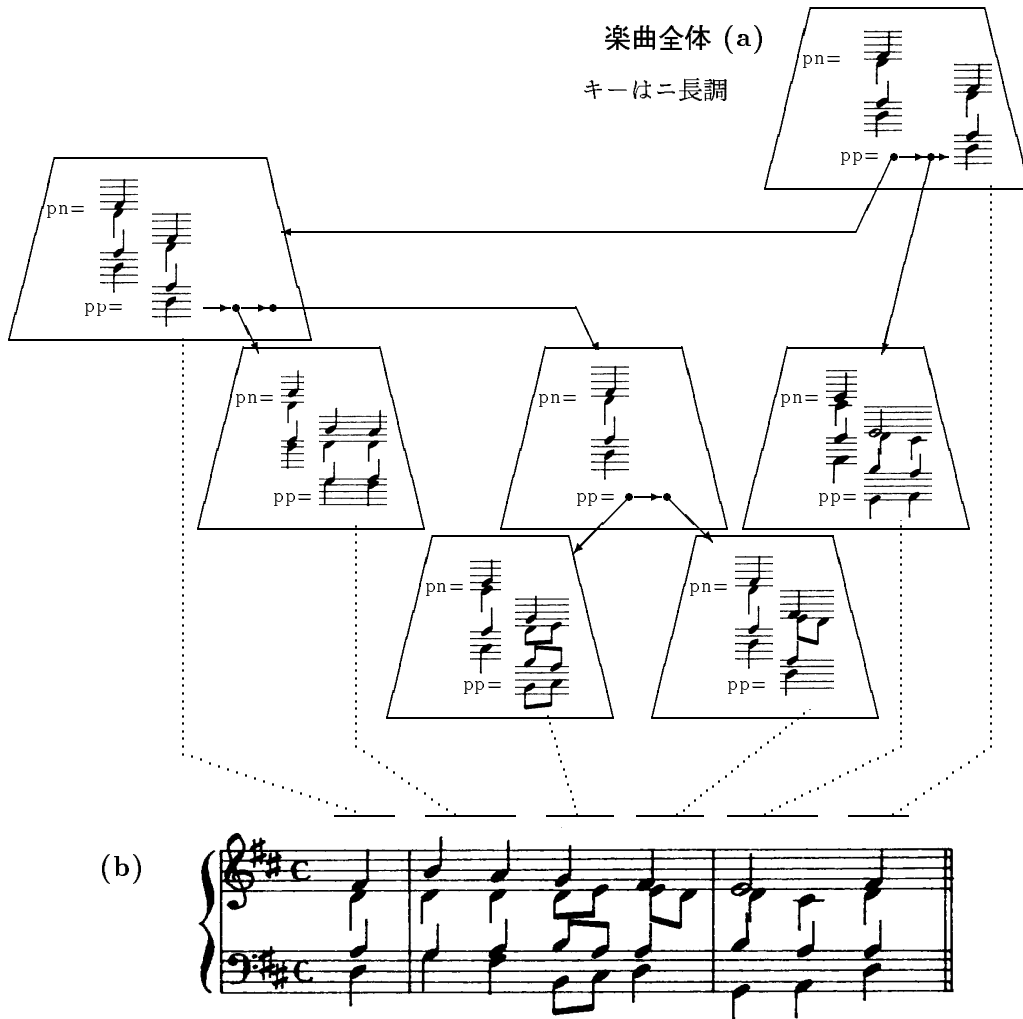


図2 バッハのマタイ受難曲 BWV244 のテーマ部分 (b) とそのタイムスパン還元結果の DOOD 表現 (a)

OOMP が主観的な情報をどのように表現するのかについて、Balaban は陽に触れていないが、少なくとも、SMP では全ての属性がオブジェクト識別子に影響を与えてしまうので、我々のように主観的な情報を事実に関する制約や関係として取り扱わない立場でオントロジーを構築するのは困難であろう。

さらに、演奏パラメータを実際に計算するための推論システムとして、OOMP にはまだ意味論が定義されていないが、DOOD では意味論やその実装が研究されているという点も重要だと考える。

5.2 オントロジーライブラリ

音楽システムの比較評価において音楽のオントロジーライブラリは有効である。これまで提案されて来た種々の音楽システムや計算モデルの比較は、前述した音楽データや知識の 3 つの特徴があるために容易ではない。

Desain らは、演奏生成システムの研究成果を比較

するために、各システムのアルゴリズムや処理手順を記述する共通プログラミング言語を提案した [Desain 98]。これに対し我々は、音楽的な概念や関係のレベルでアルゴリズムや処理手順を比較すべきであり、プログラムの比較ではレベルが低すぎると考える。もし各演奏生成システムを音楽の共通ドメインオントロジーのライブラリに基づいて構築すれば、システムの用いているタスク、システムの扱っている音楽的な概念や関係が直接的に明らかになり、生成された演奏の比較や評価 / 改良を行う際の改良点の同定などが容易になると考える。

音楽情報処理の他アプリケーションシステム [平賀 99] (例えば、自動作曲、伴奏、ビートトラック、音楽データベースなど) でも、オントロジーライブラリに基づくシステム構築は有効であろう。

6. ま と め

音楽データや知識には、暗黙的、曖昧 / 部分的、主観的という 3 つの特徴がある。そのような音楽データや知識を取り扱う音楽システムにおいて、いかにデータや知識に合理的で形式的な定義を与え、それらに対する十分制御された計算を実現することは大変困難な問題である。本稿では、その一例として、演奏生成システムを取り上げ、その問題点を具体的に議論し、演奏アプリケーションオントロジー導入による解決法について述べた。

オントロジーの定義には様々な流儀があるが、人工知能の立場から「概念化の明示的な規約」(by Tom Gruber) と定義すれば、音楽データや知識の暗黙的、曖昧 / 部分的という特徴に関してはオントロジー導入が有効に機能するであろう。

では、音楽データや知識の持つ主観的という特徴に関してオントロジー導入はどれほど有効に機能するのだろうか。我々は、データや知識を明示的に記述し体系化しただけでは本質的な解決にはつながらないと考える。これに対し本稿では、主観的な情報を演奏アプリケーションオントロジーのタスク固有の拡張部分(DOOD の非固有属性)として記述する手法を提案した。しかし、主観的な情報の中にも何らかの構造や関係が見い出せるのであれば、それはオントロジーのメタモデル的機能で対処できるのではないかと考えている。

音楽のドメインオントロジー構築はまだその端緒にすぎたばかりである。演奏アプリケーションオントロジーを作成するための音楽オントロジーライブラリも未整備であり、各アプリケーション特有のタスクについても整理して行かねばならない。

我々が演奏生成のために構築する音楽オントロジーは、音楽情報処理の他分野にも貢献すると期待し、より体系的に発展させることを考えている。いずれ音楽的なデータや知識とは一体どのようなものであるかという理解が深まり、音楽という知性の本質に迫ることができると確信している。

◇ 参 考 文 献 ◇

- [青野 97] 青野 裕司, 片寄 晴弘, 井口 征士: 重回帰分析を用いた演奏表現法の抽出, *情報処理学会論文誌*, vol. 38, no. 7, pp. 1473-1481 (1997).
- [Arcos 97] Arcos, J. L., de Mántaras, R. L., and Serra, X.: SaxEx: a case-based reasoning system for generating expressive musical performances, *Proc. of ICMC*, pp. 329-336 (1997).

- [Balaban 96] M. Balaban, The Music Structures Approach to Knowledge Representation for Music Processing, *Computer Music Journal*, vol. 20, no. 2, pp. 96-111 (1996).
- [bit 98] 長嶋, 平賀, 平田, 橋本編: bit 別冊 “コンピュータと音楽の世界—基礎からフロンティアまで”, 共立出版 (1998).
- [Chafe 97] Chafe, C.: Statistical Pattern Recognition for Prediction of Solo Piano Performance, *Proc. of ICMC*, pp. 145-148 (1997).
- [Dannenberg 93] Dannenberg, R. B.: Music Representation Issues, Techniques, and Systems, *Computer Music Journal*, vol. 17, no. 3, pp. 20-30 (1993).
- [Desain 98] Desain, P., Honing, H., van Thienen, H., and Windsor, L.: Computational Modeling of Music Cognition: Problem or Solution? *Music Perception*, vol. 16, no. 1, pp. 151-166 (1998).
- [Friberg 91] Friberg, A.: Generative Rules for Music Performance: A Formal Description of a Rule System, *Computer Music Journal*, vol. 15, no. 2 (1991).
- [Heijst 97] van Heijst, G., Schreiber, A. Th., and Wielinga, B. J.: Using explicit ontologies in KBS development, *International Journal of Human-computer Studies*, no. 46, pp. 183-292, (1997).
- [平賀 98] 平賀 瑠美: 演奏の表情付け, bit 別冊 “コンピュータと音楽の世界—基礎からフロンティアまで”, pp. 270-282, 共立出版 (1998).
- [平賀 99] 平賀 瑠美, 劉 剣利, 五十嵐 滋: 音楽知識共有のための楽曲分析支援システム, *人工知能学会誌*, vol. 14, no. 3, pp. 504-511 (1999).
- [Hirata 96] Hirata, K.: Representation of Jazz Piano Knowledge Using a Deductive Object-Oriented Approach, *Proc. of ICMC*, pp. 244-247 (1996).
- [Honing 92] Honing, H.: Espresso, a strong and small editor for expression, *Proc. of ICMC*, pp. 215-218 (1992).
- [彌富 97] 彌富 あかね, 平賀 瑠美, 五十嵐 滋: 演奏ルールを用いる音楽表情の構造的展開, 1997 年度人工知能学会全国大会 (第 11 回), pp. 280-282 (1997).
- [Lerdahl 83] Lerdahl, F. and Jackendoff, R.: A Generative Theory of Tonal Music, The MIT Press (1983).
- [Narmour 90] Narmour, E.: *The Analysis and Cognition of Basic Melodic Structures - The implication-realization model*, The University of Chicago Press (1990).
- [Roads 96] Roads, C.: *The Computer Music Tutorial*, The MIT Press (1996).
- [Widmer 93] Widmer, G.: Understanding and Learning Musical Expression, *Proc. of ICMC*, pp. 268-275 (1993).

著 者 紹 介

平田 圭二

1987 年 東京大学情報工学専門課程 博士課程終了。工学博士。同年 NTT 基礎研究所入社。1990 年 2 月より (財) 新世代コンピュータ技術開発機構 ICOT に出向、並列推論マシンの研究開発に従事。1993 年 4 月より NTT 基礎研。音楽情報処理及び並行論理型プログラミングに興味を持つ。IPJSJ, JSSST, ICMA 会員。KLIC 協会理事。
(hirata@bri.ntt.co.jp)

平賀 瑠美(正会員)

は、前掲 (Vol.14, No.3, p.511) 参照。