

# 音楽ハイパーテキストシステム: HyperScore

有吉 勇介 † 下條 真司 †† 宮原 秀夫 †

† 大阪大学 基礎工学部 情報工学科

†† 大阪大学 大型計算機センター

本論文では音楽構造, 解釈や演奏プランを表現するためにハイパーテキストの概念を導入したハイパースコアモデルについて述べる. まず, ハイパーテキストの概念に基づいて音楽構造のモデル化を行ない, モデルは時間軸とパート構成軸, 役割の軸の3本の軸を持つ空間中に音列の意味的な固まりを表すフレーズを配置した. 各軸はその軸上に音楽構造を持った構成になっている. この空間を平面に投射したビューを提供し, 音楽構造に基づいた強調表示で, 利用者の音楽構造の理解を促進する. また, 解釈や演奏プランを参照リンクとコメントノードによって表現する.

## HyperScore: A Music Hypertext System

Yuusuke ARIYOSHI † Shinji SHIMOJO †† Hideo MIYAHARA †

† Dept. of Information and Computer Sciences,  
Faculty of Engineering Science, Osaka University

†† Computation Center, Osaka University

In this paper, we propose Hyperscore model, a structured hypertext model, to express a semantic structure, annotations and relations in a musical score along with other musical structure such as instrumental parts. In this model, a musical score is represented as a form of a hypertext which has nodes and two types of links, a structural link and a reference link. In the implementation, Hyperscore model represents a schema of musical score database. which stores a structure of a musical score, a role of a musical element, and annotations and relations to the musical elements. By querying this database, we provide various views of musical structure in various abstraction level.

## 1 まえがき

楽曲は音楽形式や和声進行などの音楽構造を上位構造として持っている。指揮者や演奏者は、まずこの音楽構造を楽譜から読みとる。それから、その音楽構造に基づき曲の解釈を行ない、各部分での抑揚の付け方や、どの演奏技法を用いるかという演奏のプランを立て、練習、演奏が行なわれる(図1)。つまり音楽構造は、曲を演奏するにあたり最初に調べられ、以後それに基づいて作業が行なわれる重要な情報である。

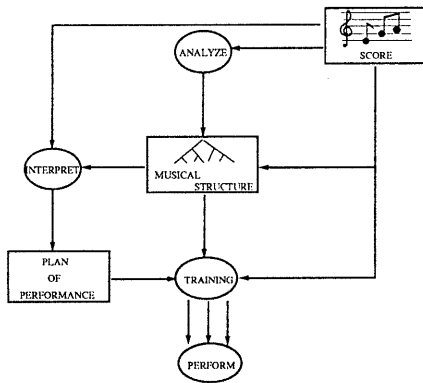


図1: 楽譜から演奏までの作業の流れ

しかし、この音楽構造は楽譜に明示的に書かれておらず、指揮者や演奏者は知識と経験により、構造を楽譜から読み取っている。そのため、初心者・初学者など知識や経験の乏しい者は、楽譜からこの音楽構造を読みとることができず、うまく演奏プランを立てることができない、練習等で相手に意図をうまく伝えられない、あるいは、相手の意図を捉え損なう、などという問題が起こる。

これらの問題は楽曲の音楽構造やそれに基づく解釈、演奏プランを表現する手段が無いために起こると思われる。現在、上記の作業は全て楽譜を用いて行なわれ、音楽構造等の情報は必要に応じて楽譜にコメントとして書き込まれる。そのため楽譜は大量の書き込みがされ、必要なコメントが別のコメントに埋もれてしまうなどということも起こっている。

本稿では音楽構造、解釈や演奏プランを表現するためにハイパーテキストの概念を導入したハイバースコアモデルを提案する。さらに、このハイバースコアモデルに基いた楽譜ハイパーテキストシステムを実装する。実装ではハイ

バースコアモデルは楽譜データベースのスキーマの役割をしている。楽譜データベースには楽譜情報と音楽構造やコメントが格納されており、検索をかけることで音楽構造を異なったビューで得ることができる。

## 2 音楽構造を表現するためのモデル

楽曲のモデル化にあたり、指揮者や演奏者などがどのように楽曲の音楽構造を捉えているのかをインタビューのもとで調査を行なった。

調査の結果、楽曲を捉えるのに一つ一つの音符をバラバラに捉えていてもあまり意味はなく、幾つかの音符がまとまったかたまりで意味を捉えている。本稿では楽曲を細かくかたまりに分解していき、意味を持つ最小単位に分解したものをフレーズと呼ぶことにする。

調査により聞きとった音楽構造の捉え方の例を下に挙げる。

- 指揮者は主題がどのように受け継がれ変奏されていくのかという楽曲の構成をつかみ、いつどの演奏者にどのような合図を送るのかを決める。また、オーケストラなどパートが多数ある楽曲では、まずは個々のパート毎ではなく、同じ音列を演奏しているパート群や、同じ役割のパート群を一つのグループとして捉えている。
- 演奏者はその時の主旋律に合わせて自分の演奏のテンポや音量などを調節する。逆に自分が主旋律の時は指揮者とともに演奏をリードする。そのためには、自分のパートの役割や、その時々でどのパートが主旋律なのかなどを知る必要がある。
- コンサートの照明や音響などの演出担当者は、いつ誰が主旋律か、ソロやアドリブを行なうかなどの情報によって、照明やミキサの操作を行なう。

上記の3つに共通しているのは役割という概念である。役割とは音楽の3要素であるメロディ・ハーモニ・リズムを指し、同時に演奏されている各々のフレーズのは、ある時点である役割を担っている。例えば、アルトのフレーズがメロディであり、その時のベースのフレーズはハーモニの役割を担っている、というように用いる。例から、演奏者は自分のパートのその時々役割を把握する必要があることが分かる。また、メロディという役割をいつこのパートが担当しているかということは3例ともに出てきており重要な情報である。

しかし、同じメロディという役割を持ったフレーズでも楽曲の進行に沿って、例えば主題、応答、変奏というように意味が異なる。この楽曲の骨組みの構成を音楽形式と呼ぶ。この音楽形式の構造の中でフレーズが持つ意味を進行上の意味と呼ぶ。

## 2.1 楽曲の空間モデル

前述の役割と進行上の二つの意味はそれぞれ空間と時間に依存し変化する。また、楽譜は楽曲のパート構成により構造化されている。そのため、フレーズの意味に注目した時の楽曲の構造は、楽曲の進行を表す時間軸と、音楽の3要素を表す役割の軸、パート構成を表す軸の3本の軸でできる空間上にフレーズを配置した構成にモデル化できる(図2)。

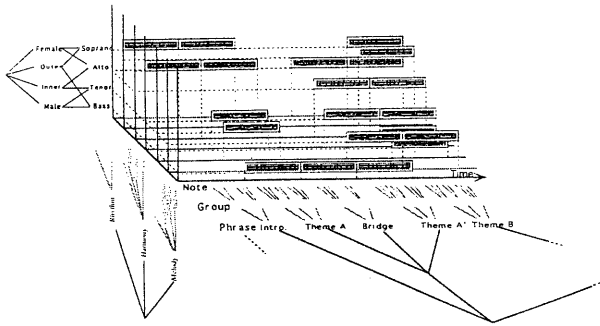


図 2: ハイパースコアモデル

モデルの中心となる音楽構造は、楽曲の進行を表す音楽形式の構造である。音楽形式の構造は楽章や楽段、フレーズなどの音楽形式の要素をノードとし、それらを構造リンクで繋いだ木構造をしている。また、フレーズに対応するノードはそのフレーズの音楽中での役割を意味する音楽の3要素メロディ、ハーモニ、リズムのいずれを担っているかなどを表現する。モデルはこのフレーズノードによって音楽形式の構造とそれ以外の二つの構造として、音楽の要素(メロディ、ハーモニ、リズム)の構造、パート構成の構造を関連付けた構成になっている。

また、モデル中の各軸上の要素間には図3, 4, 5の様構造が存在する。時間軸上には楽曲形式を表す木構造が存在し、パート構成の軸上には、例えばオーケストラなら弦楽器、木管楽器、金管楽器、打楽器に分かれ、さらに弦・木管・金管楽器は各々4声体に分かれる構造を持っている。役割

の軸も同様に音楽の3要素からさらに細かい役割に分かれる構造を持っている。

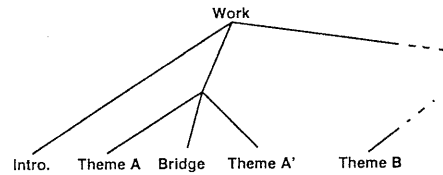


図 3: 音楽構造: 音楽形式

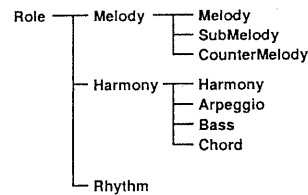


図 4: 音楽構造: 役割

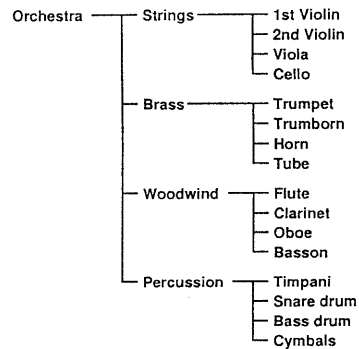


図 5: 音楽構造: パート構成

## 3 楽譜におけるコメント

ここでは実際に指揮者やパートリーダが楽譜に書き込んだコメントの構造を分析し、それを表現するハイパーテキスト構造を定義することで、2.のモデルに解釈や演奏プランを表現する能力を加える。

### 3.1 コメントの構造

まず、実際の楽譜から何に対してコメントされているのか、その対象の構造を調べた。その結果、コメントの対象と

なっているものは、音、音列、フレーズなどや表情記号や速度記号など様々である。しかし、単一パートの連続した標語や記号、またはそれらの隣接したものを一まとまり(以下領域と呼ぶ)と考え、コメントの構造は以下の7種類に分類できることが分かった(図6)。

- 1) 一つの領域
- 2) 時間的に連続した領域の間(例、プレスや間の取り方)
- 3) 時間的に連続した領域の関係(例、変化の明示、つながり感の指示)
- 4) 時間的に離れている領域の関係(例、呼応関係)
- 5) 同時に存在する領域間の関係(例、モノフォニー和声)
- 6) 時間的にも空間的にも離れた領域間の関係(例、音やメロディの受渡し、旋律線の対移法的な関係)
- 7) 領域とその内部にある部分領域の関係(例、音符の分割)

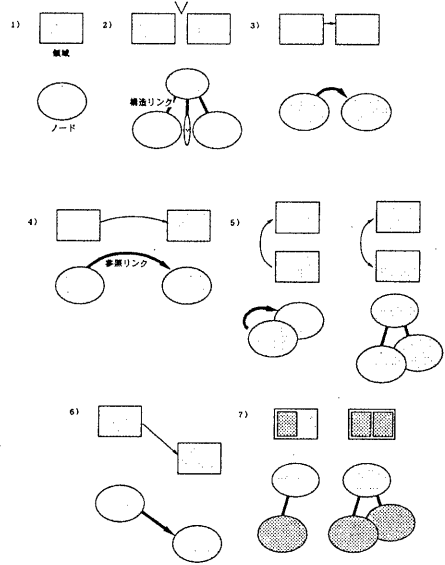


図6: コメントの構造

次に、この7種類の構造に対応するハイパーテキスト構造について述べる。

### 3.2 ハイパーテキスト構造との対応

本節では上記7種類の構造をハイパースコアモデル上のリンクとして表現する。すなわち楽譜に対するハイパーテキストとして、コメントを表す。

ハイパーテキストの特徴として1)非連続性と、2)多岐に渡る情報の流れを提供できることが挙げられる[1]。ここでは、この柔軟な構造の表現能力により、互いに関連しあう複雑な音楽構造を表現する。またデータベースと異なり後から柔軟に構造の変更ができるため、利用者が自分の考えに合わせて構造を変更することが可能になる。また、ハイパーテキストのモデルは利用者にダイレクトマニピュレーションによる情報操作を提供しやすい。この特徴により、利用者は今まで通りの「楽譜にコメントを書き込む」感覚でハイパーテキスト構造に情報を格納できる。

一般に、ハイパーテキストのモデルは情報を格納するノードと、関係を表現するリンクの2つの要素から構成されるが、ここでは構造リンクと参照リンクの二種類のリンクを持つハイパーテキストモデル[2]を採用し、構造リンクで楽曲構造を、参照リンクでコメント構造、つまり解釈・演奏プランに関する関係を表現する。また、ノードで音楽構造上のノードと、コメント構造での領域を表現する。この

ハイパーテキストモデルと7種類のコメント構造を以下のように対応付ける(図6)。

- 1) ノード
- 2) 上位ノードへの参照リンク
- 3) 参照リンク
- 4) 参照リンク
- 5) 参照リンク、上位ノードで領域を統合
- 6) 参照リンク
- 7) 構造リンクで下位ノードにする

## 4 ビュー: 音楽構造の可視化

複雑な立体的形状を伝えるのに複数の投影図を使って説明するように、ビュー機能は複雑な音楽構造の色々な異なる視点からのビューを提供する。例えばメロディのフレーズだけを強調表示した楽譜や、普通にパート毎に段組みをした楽譜ではなく、役割毎にメロディ、ハーモニ、リズムと段組みをした楽譜などを提供する。このビュー機能により利用者は音楽構造の理解の促進をはかる。

利用者はモデルからビューを取り出すために次の5つを指定する。

## 1) 投影面

2章で述べたように、ハイパスコアモデルでは音楽構造を3本の軸を持つ空間としてモデル化を行なっている。ここでは図7, 8のようにモデルの3次元空間をどの方向に投影したビューを取り出すか指定する。具体的には、利用者は画面の縦軸、横軸をモデルの時間軸、パート構成の軸、役割の軸のいずれに対応させるかを指定する。これにより取り出されたデータのビュー上での位置が決まる。言い替えると、指定された軸の値はビュー上の位置によって表現されることになる。

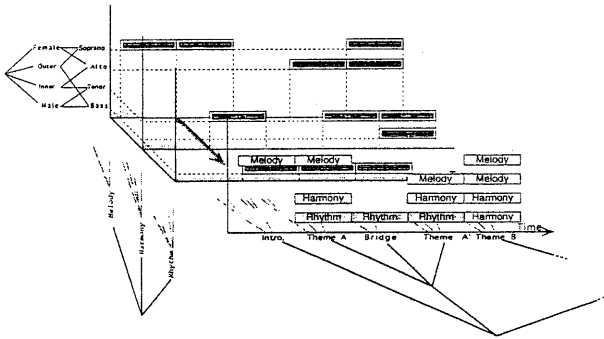


図 7: ビューの投影-1

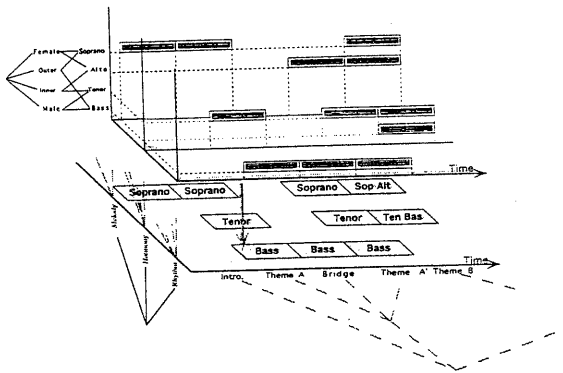


図 8: ビューの投影-2

## 2) 強調表示ルール

1) の指定でビューの軸に割り当てられなかった軸の情報つまり、画面上の位置で表現できない情報を、表示

音符	色
文字列	スタイル, 大きさ, 色, 表示位置
領域枠	スタイル, 幅, 色
領域背景	色

表 1: 表示属性

ルールによって表示属性で視覚化を行なう。例えば普通の楽譜と同じ表示を指定すれば、横軸でモデルの時間軸を、縦軸でモデルのパート構成の軸を表現することになる。この時、役割の軸の情報はビュー上の位置では表現できない。そこで、フレーズの属性役割の値によって音符の色を変えるルールを割り当てることによって、役割の情報を視覚化することができる。

## 3) 表示選択

例えば、メロディだけ表示する、または、しないないなど、特定の情報だけを表示する、もしくはしないこと指定できる。このとき、指定された条件を満たす情報だけがモデルより取り出される。

## 4) 粒度

曲全体の形式構成を知りたい時などは、細かな音符情報はかえって邪魔である。そのような時は、例えばフレーズを最小単位として楽曲形式の構成が俯瞰できればよい。つまり、利用者が局所的な細かい構造に注目しているのか、大域的な構造に注目しているのかによって、細かい所の情報の表示を行わず、幾つかをまとめ一つのものにし表示すれば良い。表示の詳しさの制御をするには、音楽構造の階層構造に従って、あるレベルより細かい情報は表示を行わない指示を出す。このとき、そのレベルより細かい情報は無視されるか、上位階層の粗い値に抽象化されることで表示が行なわれる。

## 5) コメントの表示

コメントは2つの条件で表示されるかが決まる。一つはコメントの種類属性を利用者が直接指定することにより、その種類のコメントの表示されるかが選択される。もう一つの条件は、一つ目の条件を満たしたコメントのうち、利用者が選択した部分空間内にあるコメントノードだけが取り出されて表示される。

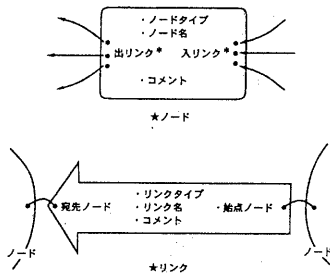
## 5 オブジェクトモデル

ここでは2., 3.で提案した楽曲構造とコメントのモデルをオブジェクト指向データモデルによって実現する。

まず、ハイパーテキストの構成要素であるノードとリンクをオブジェクトとして定義する。2.で提案したモデルは音楽形式、パート構成、役割の三つの構造からなり、各々の構造はそれぞれの構造に合わせて特殊化したノードと構造リンクで実現される。そして、3つの構造はフレーズノードのオブジェクトで関連付けられる。楽譜のコメントはコメントノードと参照リンクのオブジェクトによって、実現され空間モデルと関係付けられる。

### 5.1 ハイパーテキストモデル

ハイパーテキストモデルはノードとリンクのオブジェクトから構成される。



**ノード:** ノードは情報を表現するオブジェクトである。ノードは属性ノードタイプ、ノード名、出リンク、入リンク、コメントから構成される。

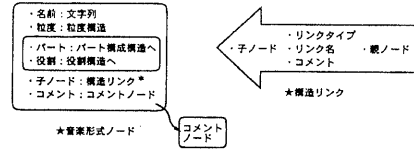
**リンク:** リンクは関係を表現するオブジェクトであり、方向性を持つ。リンクは属性リンクタイプ、リンク名、始点ノード、宛先ノード、コメントから構成される。

### 5.2 ハイパースコアモデル

#### 1. 音楽形式

音楽構造のうち、音楽形式は音楽形式ノードと音楽形式構造リンクで表現される。

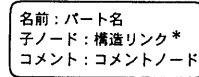
**音楽形式ノード:** 音楽形式での音列の単位であるフレーズやモチーフを表現する。部分音列を表すノードを構造リンクの先に子ノードとして持つ。フレーズはパートと役割の属性を持つ。



ち、ノード自身の進行形式上の意味は名前で持つ。

**音楽形式構造リンク:** 音楽形式の階層構造を表現する構造リンク

2. パート構成パート構成はパートノードとパート構成構造リンクで表される。

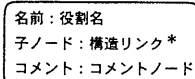


★パートノード

**パートノード:** 楽器を表現するノードで、楽器名が名前に入っている。構造の中間ノードは弦、木管、金管、打、鍵盤などの部門や外声、内声、高声、低声の四声体の分類を表す。

**パート構成構造リンク:** パート構成の階層構造を表現する構造リンク

3. 役割 (音楽の3要素) フレーズの持つ役割を表現する役割の構造は役割ノードと役割構造リンクで表される。



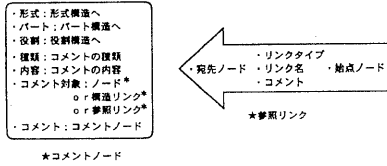
★役割ノード

**役割ノード:** 音楽の要素である役割名が名前属性に入っている

**役割構造リンク:** 役割構成の階層構造を表現する構造リンク

### 5.3 コメント

3章で述べたようにコメントの構造はコメントノードと参照リンクで表現される。



コメントノード: 内容にコメントの内容を, 対象にコメントの対象領域へのポイントを持つ. 種類にコメントの種類を持ち, コメントの表示の制御に利用する. また, コメントの音楽空間中の位置を指定するために形式, パート, 役割の属性がある. これによりフレーズと同様に検索ができるようにする.

参照リンク: 構造リンクが表現する音楽構造の枠組に入らない関係を表現する.

## 6 システムの実現

システムを開発するにあたり, 得られたビューを視覚だけでなく音としても確認できるように, 演奏など音楽処理の機能に優れた NeXT コンピュータを用いることにした. NeXT 上には GUI 構築ツール Interface-Builder, GUI ツールキット AppKit, 音楽処理用ライブラリ MusicKit などが揃っており, 比較的容易に開発できる. また MusicKit を用いることで得られたビューを演奏することができる.

本システムが想定するユーザははじめに述べたように音楽に携わる人々であり, 計算機の扱いに詳しいとはかぎらない. そのため本システムを設計するにあたり, 以下の事項を満たすことを目指して開発した

- 利用者がシステムを利用する際に, 計算機やデータベースに関する特別な知識を必要としないこと
- 利用者ができるだけ従来通り楽譜にコメントを書く感覚で使用できるようにする.

利用者はデータベースなどの使い方を知らなくても, パネルなどで視覚的に条件を与えればビューを取り出するように GUI の設計を行なった. ビューを取り出すための条件の指定などを行なうシステムのユーザインタフェースは 1) モデルの投影面を指定するパネル

2) モデルの各軸上の音楽構造を表示し, 表示選択と粒度を指定する軸毎の 3 枚のパネル 3) ビューの強調表示ルールを指定する表示属性インスペクタ 4) コメントの種類の一覧表を表示し, コメントの表示を制御するためのパネル 5) ビューを表示するウィンドウから構成されている.

## コメント

コメントはビューウィンドウ上で対象領域をマウスでクリックかドラッグで指定することにより指定し入力する. またそのときパネルでコメントの種類も指定する. 参照リンクを張るのも同様に始点と宛先の対象領域を指定することで行なわれる.

実際に楽譜のコメントは文字だけでなく, 抑揚や表情を図形で示したものも多い. そのため, システムのコメントの機能としてコメントに文字だけでなく図形も書き込めるようにした. 将来的には音声や MIDI などの演奏情報もコメントすることを考えている.

## 7 関連研究

音楽を構造的に扱うものとしては楽譜システムの研究が古くから行なわれている [4, 5]. 楽譜の表示印刷を行なう楽譜システムでは音符, 小節, 段という様に楽曲を階層的な構造として扱うシステムも多い. しかし, この構造は楽譜の表示印刷のための割り付け構造である. そのため, 楽譜の上位にある演奏に必要な音楽的意味構造を表現しようとしても, 構造が小節で分断さるなど不都合なことが起こるなどする.

作曲や分析目的の音楽言語システム [5, 6] では, 音列をリストやオブジェクトすることにより一つのものとして扱えるものも多い. さらにそれに属性やリストを付け加える形で意味の付与ができるものもある. しかし, これらを扱うには計算機言語を扱う能力が必要である. また, インタフェースが貧弱なものが多い.

1 章で述べた楽譜に書かれた楽曲を演奏する一連の作業を支援するためには, 楽譜システムの持つ五線譜という利用者の扱いなれたインタフェースの提供と, 音楽言語の持つ楽曲の解釈・演奏に必要な音楽構造を表

現し扱う能力の二つを合わせ持つことがシステムには望まれる。

しかし、現状ではその様なシステムは少なく、わずかに UPIC や Nutation [4, 7] が挙げられる。UPIC は作曲目的のシステムであり、対話的にタブレットに図形操作を行なうことで音楽制作ができる。しかし、UPIC は、楽譜とは異なる独自の図形表現を用いるようになっている。そのため、楽譜を使いなれた者がすぐに移行できるわけではない。

Nutation は楽譜表示・印刷と作曲編曲を目的としたシステムである。Nutation は楽譜の表示機能の他に、任意の連続した音列をオブジェクトにする構造的な機能を持っている。また、Nutation は Objective-C のインクリメンタルコンパイル環境の上に構築されている。その環境によって Nutation が提供しているクラスを拡張したり、新しいクラスを追加することが動的にできる。Nutation を拡張することで、楽譜のユーザインタフェースと音楽構造の表現の両方の能力を持つシステムを作ることはできるであろう。しかし、そのためには利用者は Objective-C を扱う能力を持つ必要がある。

このように、利用者が扱いたれた五線譜の GUI を持ち、その背後に音楽構造を持つことで音楽構造の可視化や音楽構造に基づいた操作ができるシステムは今まではなかった。

## 8 むすび

本研究では楽曲を楽譜から演奏するまでの一連の作業を支援する目的の音楽ハイパーテキストシステムハイパースコアについて述べた。ハイパースコアは音楽構造の理解を促進するための音楽構造の可視化機能と、解釈や演奏プランをコメントとして書き込める機能を提供する。

本研究では、まずハイパーテキストの概念を導入し音楽構造のモデル化を行なった。モデルは時間軸とパート構成軸、役割の軸の3つの軸を持つ空間中にフレーズを配置し、各軸にはその軸上に音楽構造を配置した。また、コメントは参照リンクとコメントノードによって表現した。

このモデルをオブジェクト指向に基づいてデータ設計を行ない、NeXT コンピュータ上にプロトタイプを開発した。プロトタイプは利用者ができるだけ従来通り楽譜にコメントを書き込む感覚で使用でき、かつ、音楽構造の可視化や音楽構造に基づいた操作ができるものを目指した。

今後は、本システムを実際に使用してもらい評価することを考えている。そのために、音楽構造の入力の支援する機能や、利用者独自の楽曲解釈に対応するために音楽構造の変更などのカスタマイズ機能を持たせることを考えている。また、和声進行など他の音楽構造も採り入れる必要もあると思われる。

謝辞 本研究に協力して頂いた NTT ソフトの上原 征樹氏に感謝します。

## 参考文献

- [1] J. Nielsen: "HyperText & HyperMedia", Academic Press, 1990.
- [2] P. Bruzza, Th.P. Weide: "Two Level Hypermedia. An Improved Architecture for Hypertext", *Pro. of DEXA90*, Springer-Verlag, pp.76-83, Aug 1990.
- [3] 上原, 有吉, 下條, 西尾, 宮原: 「音楽情報の多目的ビューについて」, 信学技法, DE91-55, pp.21-26, Mar. 1992.
- [4] 田口 ほか: 「音楽情報処理の技術的基盤」, 平成4年度文部省科学研究費総合研究 (B), 課題番号 043352030, Mar. 1993.
- [5] B. Balaban: "Music Structures: Interleaving the Temporal and Hierarchical Aspects in Music", *Understanding Music with AI*, MIT Press, pp.110-138, 1992.
- [6] E. B. Blevis, M. A. Jenkins, J. I. Glasgow: "On Designing a Typed Music Language", *Understanding Music with AI*, MIT Press, pp.140-154, 1992.
- [7] G. Diener: "TTrees: A Tool for the Compositional Environment", *The Well-Tempered Object*, MIT Press, pp.157-168, 1991.