

# Web 上での音環境設計実現に関する研究

中西 功\*<sup>1</sup>, 林 秋光\*<sup>2</sup>

## 1. まえがき

普段我々の身の回りには音が溢れている。その中には車の騒音などの不快な音も多い。その中で「サウンドスケープ (Soundscape)」という言葉がカナダの作曲家マリー・シェーファーによって提唱された<sup>[1]</sup>。サウンドスケープは「個人あるいは特定の社会がどのように知覚し、理解しているかに強調点の置かれた音の環境」と定義され、聴覚を切り口に周囲の世界をどのように感じるのか、そして今一度周囲の音を聴きなおい美的感覚を養うことを目的とした。

日本でも多様なサウンドスケープ論が展開しつつある。その領域は、概念成立の直接の引き金となった「現代の芸術思想」と「エコロジー」といった二つの領域にとどまらず、都市、社会、環境をめぐる様々な思想や活動さらには人々の生活にまで及ぶものとなっている<sup>[2]</sup>。その中でも周囲の音環境の現況を調べ、望みの音環境を構築することは、地域科学や地域設計という観点からも取り組むべき重要な課題であると考えられる。しかしながら、実環境での設計ならびにその評価は音源や建造物の移動などを伴うために容易には実現できない。

そこで、本研究では、世界中の誰もがどこからでも音環境の設計・評価が可能になるように Web 上での音環境設計の実現について検討を行う。以下、2章においてサウンドスケープについて説明を行う。3章ではサウンドスケープ概念に基づき Web 上での音環境設計を実現する際に必要となる機能について検討を行い、その具体的な実現方法ならびに実現結果を示す。最後に4章でまとめとする。

## 2. サウンドスケープ

### 2.1 概念

「サウンドスケープ [Soundscape]」とは、「サウンド (音)」と「～の眺め/景」を意味する接尾語「スケープ [-scape]」との複合語、すなわち「音の風景」を意味し、聴覚を切り口に周囲の世界をどのように読み込み、その環境をどのように捉えるのか、そして今一度周囲の音を聴きなおい美的感覚を養おうとするものである。サウンドスケープを現代社会における新たなコンセプトとして初めて提唱したのは、カナダの作曲家マリー・シェーファーである。この概念成立の背景には、音の世界や音楽作品を地球や宇宙のスケールに拡大して捉えようとするシェーファー個人の独特な世界観に加え、1960年代の「エコロジー運動」に代表される環境全体に対する社会的関心の高まり、「視覚中心の西洋近代文明に対する反省と聴覚文化復権の試み」といった思潮と、現代音楽の領域に共通して見られた「芸術の環境化」、「西洋近代音楽の枠組みからの解放への欲求」といった風土があった。当時のカナダでは、「公害」といえば水質や土壌、大気などの汚染が中心で、騒音はそれに比べ

\*<sup>1</sup>地域設計学講座 助教授

\*<sup>2</sup>教育学部 総合科学課程 理数情報コース (現 株式会社 ティピカル)

るとあまり問題にされておらず、また彼自身が様々な騒音に悩まされていたことから、環境への人々の意識を音の面からも高めようと努力していた。そうした中で、シェーファーは「防止」や「規制」といったアプローチのみによって音環境の問題に対処していく「騒音研究」の方法論に限界を感じ、騒音問題を含む音環境の全体を扱う研究の方法論を探る必要性を感じるようになった。その一方、音素材を「楽音」、他の様々な環境音を「非楽音」として両者を厳格に区別した「西洋近代音楽の伝統」が現代音楽における様々な試みの中で打ち破られていく流れの中で、シェーファーにとって自らが体験する「騒音の問題」を「音楽とは関係のないこと」として自らの音楽的思考の対象から除外せず、「音楽の問題」と同次元にあるものとして考えた。シェーファーは、「騒音公害は人間が音を注意深く聴かなくなったときに生じるのであり、騒音とは我々がいないがしろにするようになった音である」としている<sup>[2, 3]</sup>。

## 2.2 サウンドスケープの広がり

サウンドスケープ・デザインとはサウンドスケープ概念に基づいたデザイン活動を意味し、サウンドスケープの美的な質を改善するための原理を発見しようとするものである。そこには「騒音規制」や「作曲活動」も含まれ、シェーファーは人々の耳を周りの音環境に開かせるための音楽作品を創作している。「星の王女」という作品では、ハートレイクという森の中の湖でオペラが演じられた。音楽活動の場をコンサートホールから現実の湖に移行した目的は、反響効果や残響時間など、従来の音楽には存在しなかった様々な音響的特性を獲得することだけでなく、オペラを鑑賞するときの聴き方で自然界に向けて耳が開かれることとなり、作品が美的インターフェイスとして人々の感性を現実の環境へ開かせることだった。また聴覚の美的感覚を取り戻すための課題集「サウンドエデュケーション」を出版した。その内容は、聞こえた音をすべて書き出す「音聴き歩き」などにより「異なる聴き方への気づき」、「多様な聞き方の発見」といった「他者の理解」をも大切にするものである。シェーファーはサウンドスケープへの分析的なアプローチにも強い関心を示し、野外調査を通じてサウンドスケープ解析のための新しい原理や概念を生み出した。

日本でも既にシェーファーを越えた多様なサウンドスケープ論に基づき、多岐にわたる活動が展開しつつある。山形県では「サウンドスケープ創造事業」の一環として、県民一人一人に身近な音との関わりを意識する中で環境を考える契機とすることを目的として、「山形の音の募集」という企画が実施された。大分県竹田市にある「滝廉太郎記念館」における庭園整備計画では、普段視覚に比べて無意識化しがちな景観や環境への聴覚的感性や思考を喚起しようという考え方に基づいて実施された。「廉太郎が当時聴いていたであろうこの家や庭の音風景を、訪れる人が少しでも追体験できるような音環境を設計する」ことを基本コンセプトとし、「竹の葉が触れ合う響きを聴かせるために孟宗竹を植える」、「雀の鳴き声が聞こえるように雀の食べる実のなる木を植え、水飲み場を作る」、「当時多くの人が履いていた下駄の響きを復元するために来館者用の下駄を用意する」など、従来の視覚を意識した「庭づくり」という観点とは異なる可能性を見出した<sup>[2]</sup>。

環境問題に対するサウンドスケープの考え方の最も本質的な意義のひとつは、環境をめぐる問題解決やそのためのデザインを、これまでのようにそれぞれの個人や社会の周辺に限定して考えていくのではなく、常により広い範囲の環境への配慮、地球規模での発想をもって考えていこうとする点である。例えば、遮音性の高い住宅やその他の建築物によって、周囲の「音」をはじめ「熱」や「風」、「空気」といった様々な環境要素から切り離された建築内部空間に居心地のよい環境を整えていくということが、今日の建築における一般的な手法であるが、それによって空調装置などの音や

熱といった廃棄物を建物の外に絶えず排出し外部環境を悪化させている。このことは、コンサートホール内の音にのみ美的な感性を働かせ、その外側の音には十分な注意を払わなかったことが騒音公害を生んだとするシェーファーの考え方にそのまま当てはめることができる。

Web上でもサウンドスケープをテーマにしたサイトや音風景を紹介したサイトがある。環境庁のサイトには1996年に同庁が実施された「残したい日本の音風景百選」事業により選定された音を紹介されている<sup>[4]</sup>。「サウンドスケープ徒然記」というサイトでは、九州各地のお祭り、伝統芸能などの行事や自然音、生き物の鳴き声などのアースサウンドを「音の風景」として文章で紹介している<sup>[5]</sup>。株式会社HID内のホームページでは、北海道の代表的な音風景を聴くことができる<sup>[6]</sup>。また小中学校の授業のテーマにサウンドスケープを取り上げ、活動内容を紹介しているサイトもある<sup>[7]</sup>。ここで、サウンドスケープは一見聴覚に偏っているように見える言葉だが、これは忘れられた聴覚を意識化させるためであり、決して「視覚に対する聴覚の優位」を説くものではなく、また、音だけを切り離して考えるべきものでもない。しかしながら、先に示した従来のWebサイトは収録した音を一方的に提供しているだけであった。

### 3. Web上での音環境の設計

#### 3.1 実現すべき機能

本研究の目的は実環境の設計ではなく、Web上における仮想的な音環境設計の実現である。その表現方法に加え、電子的、仮想的だからこそ実現可能で有効な機能について検討を重ねた結果、以下に示す3つの項目が必須であると判断した。

- (1) サウンドスケープを扱った従来のサイトでは録音した音風景をそのまま公開している。しかしサウンドスケープは音を切り離して捉えるものではなく、視覚も含めたトータルな意味での風景として表現する必要がある。
- (2) 音の近くを通り過ぎる時、音は大きくなりやがて小さくなるというように、実際の環境では自分と音との距離や位置関係によって聞こえ方も違ってくる。そこで使用者の意志で風景内を動くことができ、それに応じたスピーカーからの出力変化を表現する必要がある。また、季節が変化すれば虫の鳴き声などの音風景を構成する音も変わるため、季節などによる環境の自動的な変化も大切な要素である。
- (3) 実際の音環境の設計では、演出用のスピーカーや音の出るモニュメントの設置や移動、土木工事など大掛かりな作業が必要となる。そこで、シミュレーション機能があれば擬似的に体感し評価することができる。

#### 3.2 実現方法

以下では前節で示した3つの項目を実現するための具体的な方法について述べる。

##### 3.2.1 動きに応じた音環境の変化

前節の項目(1)、(2)によれば、視覚も含めた3次元の風景を表現し、かつ、その中を自由に移動でき、さらに動きに応じて自動的に音環境が変化する必要がある。そこで、音環境の構築にはVRML (Virtual Reality Modeling Language) を用いる。VRMLはWeb上で仮想的に3次元空間を表現でき、その環境内での移動に対する周囲環境の変化が可能である。ただし、Web上でVRMLファイルを表

示するためにはVRMLプラグインが必要であり, 本研究ではVRMLプラグインにCosmoPlayerを使用した。CosmoPlayerは様々なサイトで採用されており, 次節で説明するEAIを利用したJavaアプレットの作成にはCosmoPlayerに付属するクラスライブラリが必要である。またプラグインによっては対応していないJavaScriptを, 次々節で説明するようにScriptノードで使用することができるのが特長である。

### 3.2.2 音環境設計

音環境を設計するためには, あらかじめ用意されている音に加えて, 設計者が望む音や視覚的オブジェクトをWeb上の3次元空間に追加, 削除でき, また, それを移動させることが可能でなければならない。そのため, 新たに追加したり, 削除するオブジェクトや音を収納するファイルの在処をVRMLに認識させなければならないが, 残念ながらVRMLはその機能を備えていない。そこで本研究では外部プログラムがVRMLにアクセスすることを可能にするインターフェイスであるEAI (External Authoring Interface) を用いて実現する。外部プログラムとしてはJavaアプレットを採用した。Javaとは米Sun Microsystems社が開発したプログラミング言語で, JavaアプレットとはJavaによって作られたWebブラウザの中で動くプログラムである。これらの関係を図示したのが図1である。

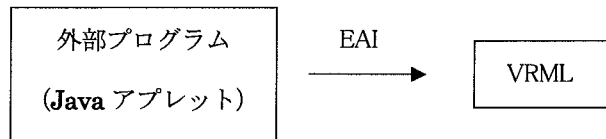


図1 外部プログラム (Java アプレット) と VRML, EAI の関係

VRMLはノードという命令文で構成され, ノードとノードを組み合わせることによって一定の動きなどを表現することができる。図2はVRML側のノード間の関係を示している。今回VRML側には10個のInlineノードとSoundノードを用意した。Inlineノードとは, 他のURLのVRMLモデルを読み込むことができるノードである。Soundノード内には音源と再生情報を表すAudioClipノードがあり, 音源のURLや再生する時間などが設定できる。Javaアプレットで新たなオブジェクトが取り込まれるとInlineノードにオブジェクトのファイルが存在するURLが送られ, 音を取り込まれるとAudioClipノードに音のファイルのURLが送られる仕掛けとなっている。Javaアプレットで1~10の番号を指定することで送られてくるURLを受け取るInlineノード, AudioClipノードが決まる。また, 取り込んだオブジェクトや音を任意の場所へ配置できるようにPlaneSensorノードを使用する。PlaneSensorノードはドラッグした移動距離を出力し, それをオブジェクトの位置やスケールを変化させるTransformノードに送ることで移動を可能にする。Transformノードは複数のノードをひとまとめにして扱うことができる。また, まとめられたノードは移動や拡大, 縮小が可能になる。

次に, Javaアプレット側の処理を説明する。Javaアプレットの役割は挿入されるオブジェクトや音のファイルが存在するURLをVRMLの

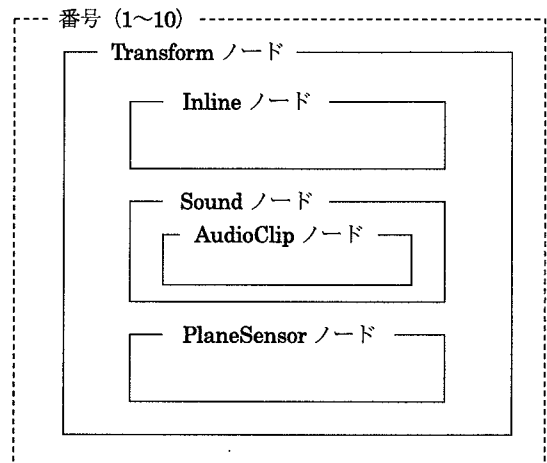


図2 VRMLにおけるノード間の関係

Inline ノード, AudioClip ノードに送ることである。URL を受け取るために変数 url1 ~ url10, sound1 ~ sound10 を定義する。url はオブジェクトの URL を指定する変数で, sound は音の URL を指定する変数である。また, VRML 側のどの Inline ノード, AudioClip ノードに送るのかを指定するために, 変数 num (1 ~ 10) を定義する。図3はJavaアプレットとそれに対応するVRMLのノードとの関係を示すものである。

あらかじめ用意されているオブジェクトや音を選択したり, 番号を指定する場合には選択形式のポップアップメニューを作成するChoiceクラスを使用する。

Javaにはポップアップメニュー, ボタン,

テキストフィールドなどを生成するクラスがあらかじめ用意されている。オブジェクトが選択されるとそのオブジェクトのURLが, 音が選択されるとその音のURLが現在選択されている番号 num に基づき url ならびに sound にそれぞれ代入される。

新たに必要なオブジェクトや音を追加する場合は, 1 ~ 10の番号, 取り込みたいオブジェクトや音のアップロード先URL, さらに音の鳴らしたい季節を指定する。ここで, 音は設計者によって鳴らしたい季節が異なるために指定させる必要がある。追加するオブジェクトや音のアップロード先URLの指定にはテキストフィールドを作成するTextFieldクラスと, ボタンを作成するButtonクラスを使用する。オブジェクトと音のURL用にテキストフィールドとボタンを2つずつ用意し, テキストフィールドにはアップロード先URLを指定させる。テキストフィールドに書かれた文字列はボタンが押されると現在選択されている番号 num に基づき url ならびに sound に代入される。ただし, 音は目に見えないために挿入された音ファイル名を表示し, 確認する必要がある。そこで, 表示にはテキストを複数行表示できるTextAreaクラスを使用する。

オブジェクトや音の削除にはButtonクラスを使用し, オブジェクト用と音用に2つボタンを用意する。ボタンが押されると現在選択されている番号 num に基づき, url ならびに sound に代入されている文字列が消去される。

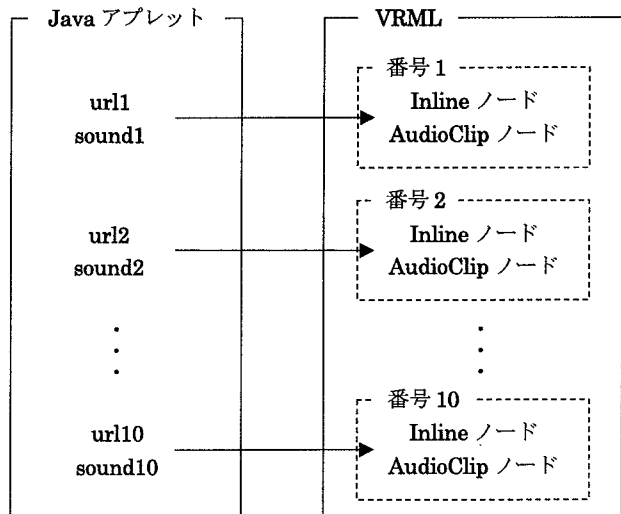


図3 Javaアプレットの変数とVRMLのノードとの対応

### 3.2.3 時間による環境の変化

時間による環境の変化はVRMLにおいて行う。現在日時を取り込み, 音の鳴る時刻や色の変化が始まる時刻をセットすることによって, 季節の違いによる色の変化を表現する。具体的には, 図4のようにScriptノード, TimeSensorノード, ColorInterpolatorノードを使用する。Scriptノードとはノード内にScript言語を記述することによってVRMLであらかじめ定義されているノードでは不可能な表現を可能にする。使用できるスクリプト言語にはJavaScript, Java, CosmoPlayer使用時にはvrmlScriptがあるが, 本研究では比較的記述が容易なJavaScriptを使用する。TimeSensorノードは一定間隔で何かを繰り返す場合やある時刻になると動作を開始する場合に使用する。

ColorInterpolator ノードは色の補間を行うノードである。Script ノードで現在日時を取り込み、次の季節への変化が始まる時間になると ColorInterpolator ノードに現在の季節の色と次の季節の色情報を送り、同時に TimeSensor ノードを起動させる。TimeSensor ノードには現在の季節から次の季節までの変化時間が一周期として設定されており、起動後の経過時間が一周期に対する割合 (0~1) として ColorInterpolator ノードに送られる。ColorInterpolator ノードは TimeSensor ノードから受け取った割合に応じて、現在の季節の色から次の季節の色へ変化させる。例えば木の葉の部分構成するノードに色を送出することで季節に応じた木の葉の色の変化が実現される。

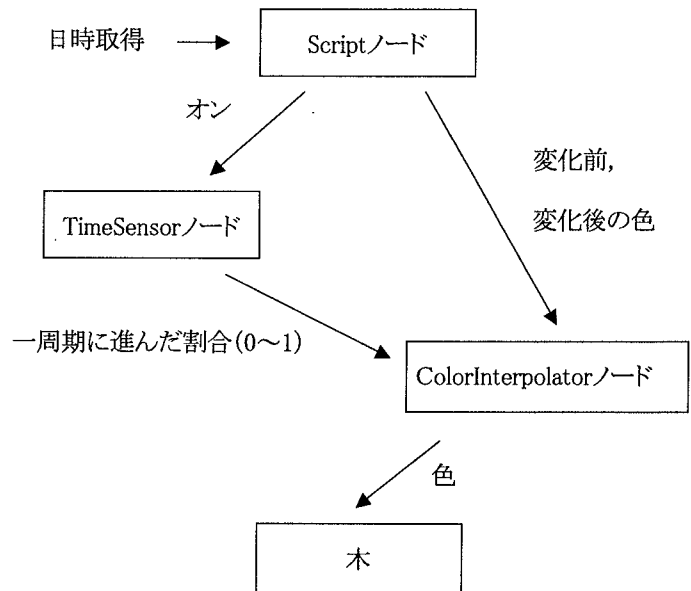


図4 時間による環境の変化におけるノード間の関係

図5は時刻に関する情報の流れを示している。オブジェクトに付加される音が鳴り始めるには音の鳴り始める時刻を、音が止まるには音の鳴り止む時刻をAudioClipノードが受け取らなければならないが、これらはJavaアプレットで処理する。設計者の音を鳴らしたい季節の情報は前節で説明したようにJavaアプレットに記述されている。Scriptノードは時刻をJavaアプレットに送り、Javaアプレットは、設計者が取り込む音の鳴らしたい季節の情報に基づき、どの番号のAudioClipノードに送るのかを判断する。

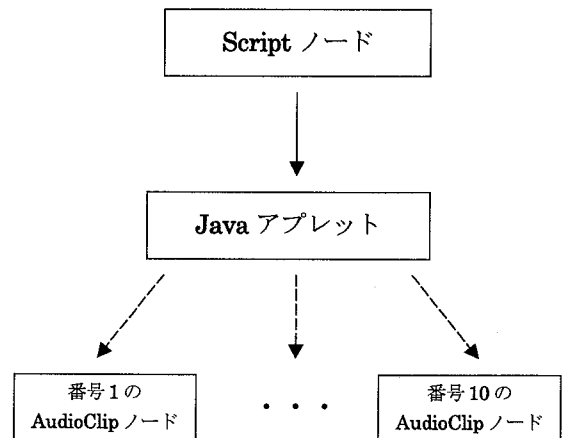


図5 時刻に関する情報の流れ

図6は色と音の状態推移を示している。色が変化する時刻になると移行期間1に入る。そして完全に次の色へと変わると次の移行期間2まで色が変化しない安定期間1となる。安定期間1において鳴るように設定された音は、安定期間1に入って鳴り始め、移行期間2でもそれは鳴り続け、次の安定期間2に入ると鳴り止むように設計されている。

色	...	移行期間 1	安定期間 1	移行期間 2	安定期間 2	...
安定期間 1 の音	...	オフ	オン		オフ	...

図 6 色と音の状態推移

### 3.2.4 その他の表現

オブジェクトを配置するためのフィールドの作成には ElevationGrid ノードを使用する。これは X-Y 平面上の格子面の各点に高さを与えてできる面を定義するノードで、高低差のあるフィールドを作成することができる。オブジェクトを配置する時、3.2.2 で説明したように PlaneSensor ノードを使用するが、PlaneSensor ノードは平面上で指定されるので、フィールド面に高低差がある場合、その高低差を考慮して表示する必要がある。そこでフィールドの各地点の高さの情報を Script ノードにセットしておき、その都度判断させることにする。模式図を図 7 に示す。配置されるオブジェクトの Transform ノードから現在位置を対応する Script ノードに送る。Script ノードは送られてきた現在位置から、その地点のフィールドの高さを調べる。そして送られてきた現在位置にフィールドの高さを加えて、配置されるオブジェクトの Transform ノードに送り返す。このようにすることで、オブジェクトをフィールドの高さに沿って移動させることができる。

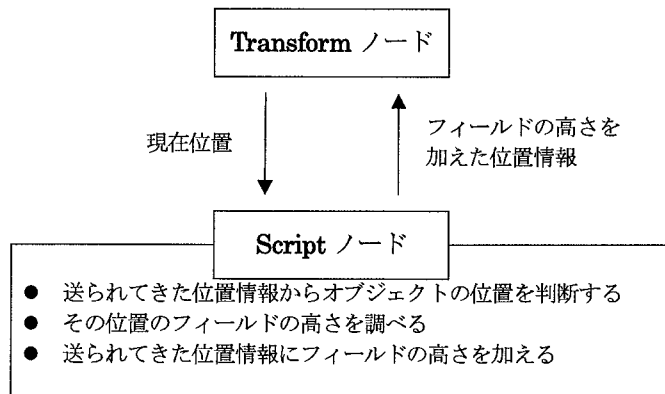


図 7 フィールドの高さにあわせてオブジェクトを配置するためのノード間の情報のやりとり

### 3.2.5 HTML ファイルへの埋め込み

構築した音環境を Web 上で公開するためには、HTML 内に VRML ファイルと Java アプレットを埋め込む必要がある。HTML 内に VRML ファイルを埋め込むためには、embed タグを記述し、VRML プラグインに対し読み込む VRML ファイルの URL を src で指定する。例を以下に示す。width は横幅で、height は縦の長さである。

```
<embed src="xxxx.url" width=800 height=320>
```

また、HTML 内に Java アプレットを埋め込むためには、Java アプレットをコンパイルしてクラスファイルとした後、以下のように applet タグを記述し、クラスファイルの URL を code で指定する。width、height は先ほどと同じである。

```
<applet code="yyyy.class" width=800 height=280></applet>
```

## 3.3 実現結果

これまで述べてきた方法を用いて Web 上における音環境設計のためのプロトタイプをを実現した。図 8 は実現画面例である。上半分が VRML プラグインである CosmoPlayer の表示部分であり、下半分が Java アプレットの表示部分である。画面には設計者の視点として「setting view」と「start view」

視点変更ボタン

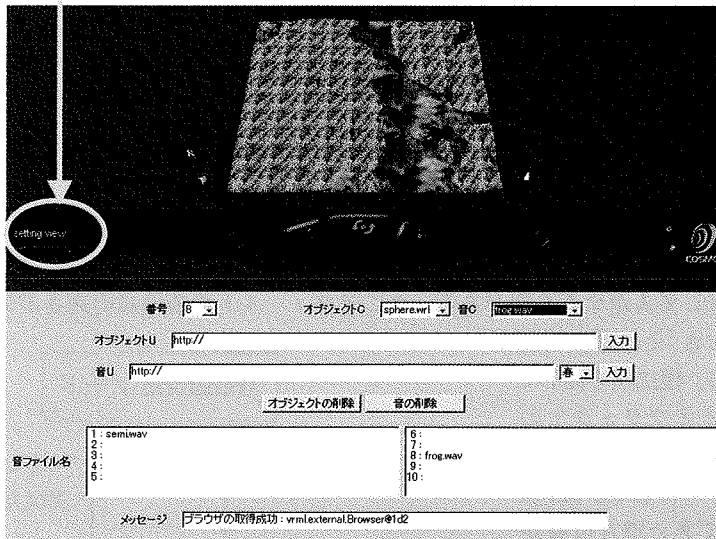


図8 実現画面例 (setting view)

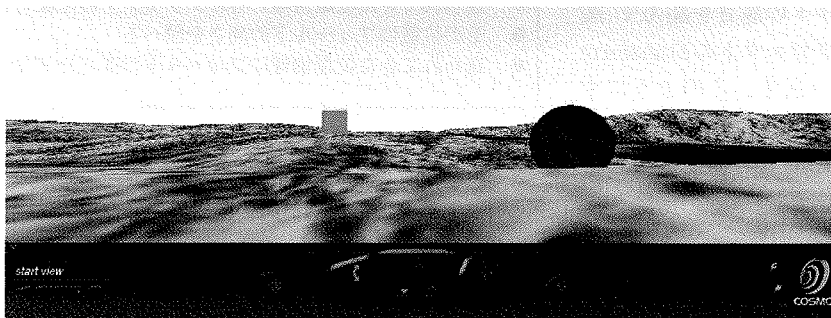


図9 start viewの画面例

が用意されており, 図8は「setting view」画面である。それに対し図9は「start view」の画面である。これらの視点は CosmoPlayer の左下に設けられている視点変更ボタンで変更される。「setting view」画面ではオブジェクトや音の挿入, 移動が可能であり, 斜め上方からフィールド全体を見渡す構成となっている。挿入されたオブジェクトはフィールド左側に表示され, 区別のために選択した1~10の番号がオブジェクトの前に表示される。挿入されたオブジェクトはマウスなどのポインティングデバイスによりドラッグすることでフィールド内に自由に配置することができる。「start view」では, 視点はフィールド部分に立つような形に変化し, 指示カーソルをドラッグさせて移動させることで, フィールド上を自由に移動することができる。



番号：取り込んだものを区別するための番号（1～10の選択）

図 10 Java アプレットによるオブジェクトと音の取り込み画面例

図 10 は Java アプレットによるオブジェクトと音の取り込み画面例である。まず、番号を選択し、あらかじめ用意されたオブジェクトをフィールド部分に挿入したい場合は、オブジェクト C 欄から選択する。設計者が用意したオブジェクトを挿入する場合はそのオブジェクトファイルを Web 上にアップロードし、アップロード先の URL をオブジェクト U 欄で指定し、入力ボタンを押す。オブジェクトに音を付加したければ、あらかじめ用意された音を選択する場合は音 C 欄から選択し、新たな音を挿入したい場合はオブジェクトと同様にその音を Web 上にアップロードし、アップロード先の URL を音 U 欄で指定し、入力ボタンを押す。ただし、音 U は設計者により鳴らしたい季節が異なるため、季節を選択する機能を備えている。オブジェクトを削除したい場合は削除するオブジェクトの番号を選択し、削除ボタンを押す。音を削除したい場合も同様である。オブジェクトに付加された音は音ファイル名欄に表示される。エラー等があればメッセージ欄に表示される。

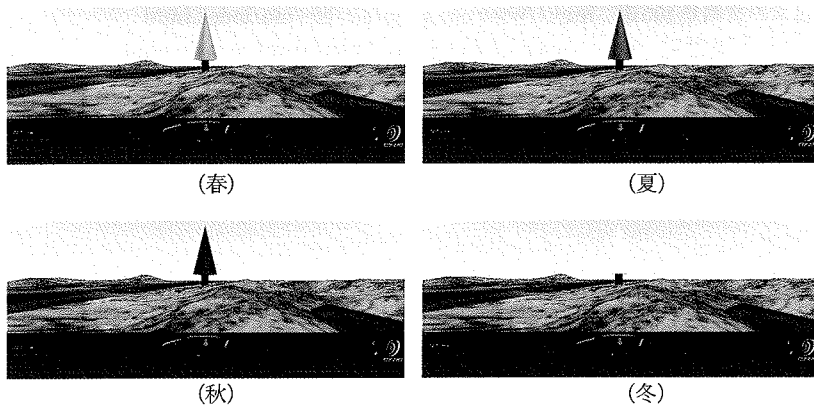


図 11 Java アプレットによるオブジェクトと音の取り込み画面例

図 11 に季節の移り変わりによる変化の表示例を示す。カラーでないので分かりづらいが、画面中央にある木の葉の部分、春には桜色、夏には葉が繁る緑、秋には紅葉の赤、冬には雪の白に色が変化するように設定されている。また、この木は、夏になるとセミが、秋になると鈴虫が鳴くように設定されている。そのため、葉が緑になるとセミが鳴き、赤になると鈴虫が鳴く。VRML は環境内での移動に対して周囲環境の変化が可能なので、木に近づくるとセミや鈴虫の鳴き声は大きくなり、

遠ざかると小さくなりやがて聞こえなくなる。詳しくは以下のホームページをご覧ください。OSは Windows Me, Web ブラウザは InternetExplore において動作を確認している。プラグインとして CosmoPlayer も必要である。

<http://www1.fed.tottori-u.ac.jp/isao/soundscape/top.html>

#### 4. むすび

Web上の音環境設計として実現すべき機能について検討を行い、それを VRML, Java アプレット, JavaScript を用いて実現した。今回実現したものはプロトタイプであるため、実際の音環境設計システムとするには、例えば、建物などの実際に環境を構成しているオブジェクトを作成したり、季節の移り変わりだけでなく、昼と夜といったより細かな時間による制御も必要になる。ただし、これらの事柄は今回提案したシステムを拡充させていくことで対応は可能である。それに対し、フィールドを構成するときを使用している ElevationGrid ノードは高低や色の変更は可能であるが広さは変更することができないなど、単純な拡張では対応できない課題もあり、今後検討して行く必要がある。また、実際に音環境の設計として用いる中で、設計システムとしての評価を行っていくことも必要である。

#### 参考文献

- [1] R. マリー・シェーファー：“世界の調律—サウンドスケープとはなにか”，平凡社（1986）
- [2] 鳥越けい子：“サウンドスケープ [その思想と実践]”，鹿島出版会（1997）
- [3] 岩宮眞一郎：“音の生態学—音と人間のかかわり—”，コロナ社（2000）
- [4] 残したい日本の音風景100選 <http://www.env.go.jp/air/life/oto/index.html>
- [5] サウンドスケープ徒然記 <http://www.d1.dion.ne.jp/~msp/turezure-main.html>
- [6] 株式会社HID <http://www.dosanko.co.jp/sound/>
- [7] EILNet博覧会 サウンドスケープ <http://www.fsifee.u-gakugei.ac.jp/eilnet/eilexpo/sound.html>