音楽情報科学 46－10
（２００２．７．７）

GDS (global delayed session) Music

--- ネットワーク遅延を伴う音楽セッション・モデル

長嶋洋一(SUAC)

物理学的な光速の限界により不可避であるネットワーク遅延を前提として、音楽演奏情報の「同時性」を否定した新しい音楽の概念に基づくネットワーク音楽セッションモデル[GDSM]を提唱した。具体的なソフトウェアとして試作したシステム "Improvisession-II"とともにその可能性を検討した。

GDS Music --- Musical session model with global delay of Internet

Yoichi Nagashima (SUAC)

nagasm@computer.org

This paper is intended as an investigation of new interactive environment in music. I will describe a performing/composing model for improvisational sessions for networks, called "GDS (Global Delayed Session) music". I will also introduce a new software called "Improvisession-II" as a prototype of the model, and this paper reports not only the basic idea but also the latest version and applications in education and entertainment.

1. はじめに

音楽における「演奏」、特に独奏でなく複数のメンバーの合奏の形態においては、基本的に同一の時間・空間の共有が前提となってきた。その理由は「音」の伝播速度の遅さにある。20世紀になって電気的なサポートが加わり、音速に比べて格段に高速な電気信号(PAと無線)により、さらに演奏情報を伝達するMIDIにより、空間的時間的同一性はかなり拡張され、例えばスタジアムで10万人の聴衆を前にしたバンドの演奏は、バンドメンバーが互いに数十メートル離れた状態でも成立し、また数百メートル離れた多数の聴衆も分散型多点PAによりほぼ良好な音響を「同時に」鑑賞できるようになった。

インターネットに代表される情報ネットワークにおいては、世界中で個人が個別のコンテンツ(音楽もこの一つ)をオンデマンドに「鑑賞」する方向に進展している。ところが「演奏」型の音楽は、ここにきて本質的な問題に直面した。情報伝達の最高速度である「光速」は、地球サイズの情報リンクを前にして、音楽のためにはどうしようもなく遅い、という限界である。通信衛星の高度や情報ネットワークのルータ遅延等を技術的に圧縮しても、光ファイバの光速伝送でも遅延は不可避的に発生し、これがインターネットによって地球規模となった場合、情報の遅延時間は「秒」オーダ(ジッタのレベルと同様)となって、アンサンブル音楽演奏に必要な同時性の確保は本質的に不可能である。

筆者はこれまで、インタラクティブなメディアアート等に関する研究を進める中で([1]-[4])、TCP/IPパケットにMIDI情報をデルタタイム等をパックして伝送する後藤氏のRMCP([5]-[7])を用いた即興セッションシステムも研究してきたが([8]-[9])、この本質的遅延は情報インフラがどう発展しても本質的に残る問題点である。シーケンスデータを伝送「再生」するのであれば、そのデータをあらかじめ先に転送しておけば遅れは問題ないが、リアルタイムに演奏情報を交換するアンサンブル演奏においては「同時」が得られない、つまり「自分が出す音と同時に鳴る相手の音」が間に合わないために、20世紀までの音楽の概念ではインターネット越しにアンサンブルはできない、という壁が明白となった。

2. "GDS Music" の基本コンセプト

2-1.「セッション音楽」の発想の転換

上記のような背景を受けて、本研究で提案するのは、いわゆる「1小節遅れ音楽」、あるいは「GDS music」(Global Delayed Session music) と名付けた、21世紀型の新しい音楽のコンセプトである。これはインターネット時代に必須必然の「地球規模の遅れ」を、本質的・積極的なコンセプトとして置いた新しい音楽モデルの提案、というものである。

これまでの技術的アプローチは全て、なんとか知覚されないように情報の遅れを誤魔化す方向でなされてきたが、直視してみればこの遅れは吸収できないので、音楽の方を新しい概念に進化させよう、ということである。音楽を一般大衆に親しみのあるポップ的な音楽、と限定した場合には、これは一定のビートに基づく音楽となるために、具体的なイメージとしては「1小節遅れ音楽」というような形式となる。しかしここには単に「常に相手から1小節だけ遅れた演奏情報が届く」「常に相手に自分の演奏情報が1小節だけ遅れて届く」というだけでない、音楽の概念のある種の革命も盛り込んでいる。

また、「GDS music」をインターネット対応として展開した場合には、新しいエンターティメント、あるいは新しいビジネスモデルが考えられる。音楽に対する文化的側面、つまり「聴くだけでなく参加する」というインターネットが本質的に持っているインタラクティブ性を盛り込んだ新しい音楽(不特定多数による参加型の音楽)という意義も重要であると考える。

2-2. Remote-GIGとOpen Remote-GIG

前述のような背景を受けて、後藤氏のRMCP研究はその後、RMCPを用いた遅延セッションシステムとして、Remoto-GIG / Open Remoto-GIG([10]-[11])を提唱した。これは、遅延によって音楽セッションの同時性は無理、という前提で、12小節のブルース形式の1ループのような繰り返し構造を単位として、「相手の1単位前の演奏と自分の演奏とでの合奏」というセッションを相互に行う、という新しい音楽概念である。自分の演奏情報は相手に遅延して届いて、相手はその「影」と同時セッションして、その演奏をまた「影」として返してくる。RMCPはUnix上のNTP(network time protocol)を利用したRMCPtss(time synchoronized server)によってそれぞれの「時計」を同期させているので、最大遅延時間よりも十分に長い「単位」を用いることにより、1単位遅れの音楽セッションがそれぞれの演奏者において必ず成立する。音楽セッションに遅れを導入する、という意味では誰でも同じところに到達するために、外見上は本研究とも極めて似たシステムである。

2-3. GDSMの思想

不可避的な遅延があり古典的な同時性のもとでの音楽セッションが成立しないインターネット時代には、RemoteGIGのように音楽そのものの概念を変える必要がある、という基本は筆者も後藤氏との議論で確認・確信した。そしてRMCPという比較的「重い」環境でなく、パソコン1台を単位としてもっと手軽に即興を重視したネットワークセッションを実現する、という新しいシステムを開発した[12]。「常に相手の影と演奏する」という基本概念は同等であるが、ここではRMCPで提供した「遅れを考慮した同時性」までも捨てて、それぞれのPlayer(ここでは演奏者というよりもセッションという一種のゲームの参加者としての意味も)の時間については、インターネット遅延時間の大きさとか音楽演奏のループ長とかをいっさい考慮・調整しない、というものである。図1はその概念図であるが、ここではRemoteGIGと違って、それぞれのPlayerの時間を揃える、という視点が不要になっている、という相違点に注意されたい。

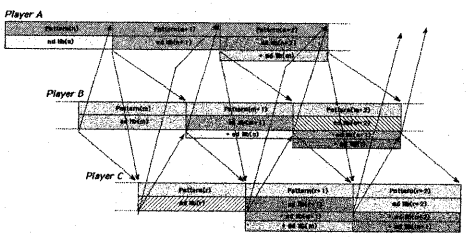

図1. GDSMの概念図

2-4. Open Remote-GIGとの相違点

RMCPをベースとしたOpen RemoteGIGとの違いとしては、(1)各プレーヤの(遅れを考慮した)時間的同期については何も考慮しない、(2)他プレーヤから届いた情報をいったん溜めて自分のループに同期させて再生合奏する、(3)自分のフレーズは自分のループの最後にまとめて送出する、(4)伴奏部分についてのMIDI情報は転送しないで各プレーヤがサーバから入手したパラメータを元にアルゴリズミックに生成する、(5)プレイヤーは芋づる式に他プレイヤーをリンクしてもよい(別の機会に発表予定)、等である。

2-5. プラットフォーム

筆者はこのアイデアを実際に"Max/MSP"上に実装したシステム"Improvisession-II"を開発し、Webでも公開した。ここではCNMAT[13]の研究成果として公開されているOpenSoundControl[14]というパッケージによって、簡単にネットワーク上での情報交換を実現できた。以降では、このシステム"Improvisession-II"の内容を具体例として紹介していくが、あくまで本研究の本質的な部分はGDSMという新しい音楽モデルにあり、将来は別のプラットフォーム上にいろいろと実装されていく可能性を持っている。

3. "Improvisession-II"

3-1. 全体構成

GDSMの試作ソフトウェア"Improvisession-II"は、Max/MSP(Macintosh)上に構築した。実験で使用したのは、3人のPlayerを想定した計3台のPowerBookおよびiBookであり、それぞれ固定IPを持ち、AirMacでイーサネットに接続した。音源は全て内蔵のQT(GM)音源のみを使用した。

3台のマシンの操作者(以降playerと呼ぶ)は、IPアドレスの指定部分以外はまったく同一の試作ソフトウェアを走らせ、マウスにて画面内のキーボードを演奏する。player同士の演奏の関係としては主従関係はなく「対等」である。そしてそれぞれのplayerのマシンは同じテンポで走るが、スタートのタイミングは任意であり、まったくタイミングの同期関係は存在しない。図2は、そのソフトウェアのメイン画面である。

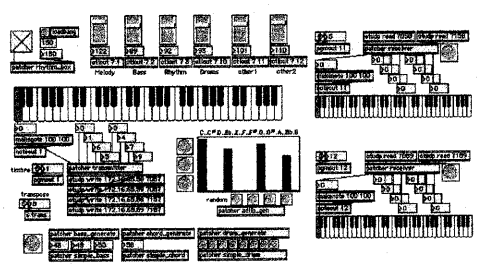

図2. "Improvisession-II"のメイン画面

2小節ループの基本的な伴奏(ドラム、ベース、ビブラフォン)がランダムにいくつかのパターンを選択して走る。ドラム、ベース、ビブラフォンの音量は個別に指定可能である。

3-2. playerのメロディー・アドリブ演奏

playerの画面内キーボードのマウスによる演奏は、大きく以下の3つの演奏モードにより行える。これらの演奏情報のトータルが自分のフレーズとなる。

(1)画面内の鍵盤をマウスでクリックして演奏する
(2)鍵盤の右下の棒グラフによりスケール登場確率を設定して、その左側の3個のボタン(音域が異なる)によってそのスケールを用いたアドリブフレーズを自動生成させる
(3)棒グラフの下にある4個のボタン(音域が異なる)によってランダムなクロマチック・アドリブフレーズを自動生成させる

3-3. 1小節遅れと同期関係

player Aの画面内キーボードのマウスによる演奏は、player Aの2小節ループを単位としてサンプリングされ(今回の分解能は16分音符単位)、player BおよびplayerCにイーサネット経由でbroadcastされる。自分のこの演奏音色、および音量は画面内のスライダーとインクリメンタで自由に指定できる。同様にBからはAとCに、CからはAとBに送られる。

player Aのマシンでは、player Bから受け取ったマウス演奏が自分のマシンのループのタイミングに同期して、つまり情報を受け取ったあとで最初に登場する自分のループの先頭に同期して再生される。この音量と音色は自由に設定できる。player Cからの演奏情報も同様。他のマシンのPlayerでもそれぞれ同様に動作する。つまり全てのplayerは「一緒に、異なった音楽を楽しむ」のである。これは、自分がセッション音楽した模様をサンプリングしてオリジナルサウンドクリップ等として利用したいという場合、playerごとに異なった音楽となるためにそれぞれがそれぞれの著作者となって、本人は自由に利用できるという新しい音楽著作物の活用シーンの可能性を提示している。SMFのように固定した音楽演奏情報の再利用が著作権によって不自由であったのと対照的である。

player Aにとっては、自分のマシンで走っているBGMに同期してアドリブでマウス演奏したフレーズが聞こえ、そこに加えてplayer Bとplayer Cがアドリブ演奏したフレーズも同期して聞こえてくるが、これはちょっとだけ過去の演奏である。これに対応して返したフレーズは相手に、これまたちょっとだけ遅れて届く。常に相手の「影」を追い掛けながらの掛け合いである。

3-4. 内部動作の詳細解説

BGM再生とフレーズ演奏/サンプリングはいわゆるシーケンサ的なタイミングパルスに同期

して行うが、固定的なシーケンスデータは持たない。フレーズの要素をテーブル等で持ち(これはネット音楽としてはデータベースサイトからいつでもダウンロードして更新できるようにする)、この選択情報だけをリアルタイム交換することで速度を確保する。図3はシステム全体の動作を規定するマスタークロックに相当するタイミングパルスを生成しているブロックである。基準タイミング信号として、b01からb32までの32個の等間隔なタイミングパルス例を生成する。これが16ビートで2小節分としてループする。

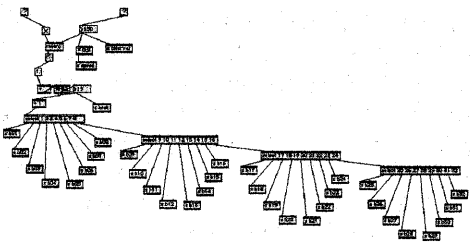

図3. マスタークロック生成サブパッチ

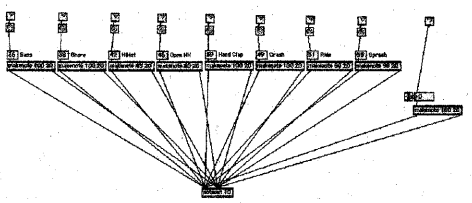

図4. ドラム音源サブパッチ

ドラム部分としては図4のように8種類のシンプルな打楽器のみを使用した。図3のタイミングパルスのそれぞれに対して、図4のどの打楽器を対応させるか、という部分がドラムパターン生成ブロックとなり、今回は図5のようにそれぞれのスイッチを配置して設定した。

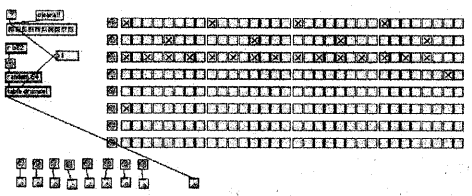

図5. ドラムパターン生成サブパッチ

ベース音とコード音の音源としては図6のように非常にシンプルにしている。コードのパターンは図7のようにランダムにいくつかのパターンから生成して、同時の和音ないしアルペジオ的な和音とした。

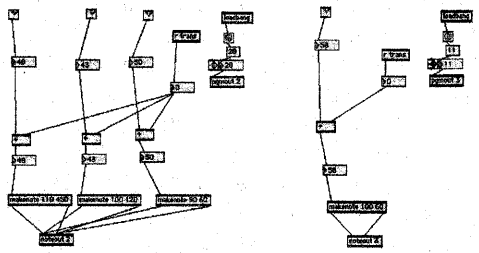

図6. ベース(左)とコード(右)の音源サブパッチ

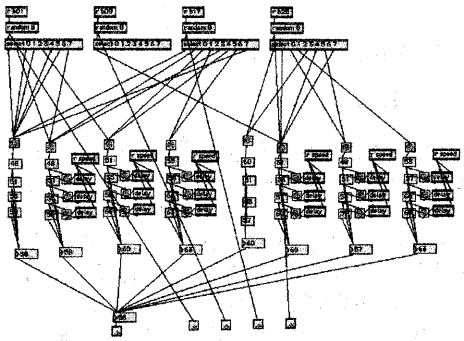

図7. コードパターン生成サブパッチ

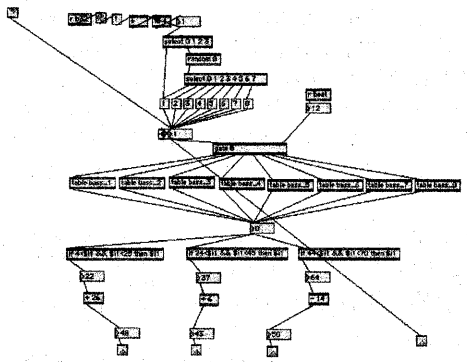

図8. ベースパターン生成サブパッチ

ベースのパターンは図8のように、durationとして3種類の長さのみを定義して、図9のように8種類のテーブルで設定フレーズから選択した。テーブルは3種類のdurationをデータ値域として分離することで一つのテーブルに多重化している。具体的なベースのフレーズをいちいちUDPパケットとしてリアルタイム交換せずに、あらかじめこのようなテーブルをインターネットからダウンロードしておいて選択番号のみをリアルタイム交換することで、多様なバッキングパターンを最少のネットワーク負荷により実現で

きるサンプルである。

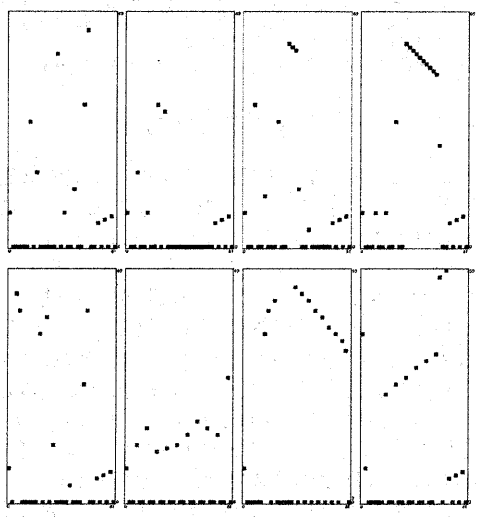

図9. ベースパターン生成テーブル8種

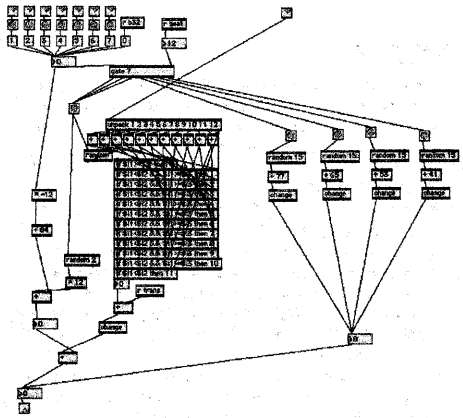

図10. アドリブメロディー生成サブパッチ

メロディーとして、Maxの画面内の鍵盤をマウスで演奏する以外のフレーズ(半)自動生成ブロックを図10に示した。ここでは7個のボタンのうち3個は、図2のメイン画面内の棒グラフによって12音のそれぞれのスケールの出現確率の重み付けを設定し(キーはon C、全体としてはコードはCm7ぽいtonalityとなっている)、ここにそれぞれの16分音符に相当するタイミングでノートのランダム生成を行う。前回と同じピッチの場合には後の音を省略することで疑似リズムを発生している。残り4個のボタンについては、単純に12音からのランダムノートによってフレーズを生成している。

図11が、UDP/IPパケットの送信部サブパッチ、図12がUDP/IPパケットの受信部サブパッチである。試作ソフトウェアでは、16分音符単位で2小節に32データがあるために、前半と後半とに分けて20バイトのUDP/IPパケットをbroadcastしている。データはintegerをパックしてリストとして交換している。フレーズのサンプリングは16分音符の幅で32音符分だけ前後にシフトした時間窓で、得られた最終情報をデータとしている。今回はこれより細かい演奏情報は失われる。UDP/IPパケットとして受け取ったデータは、まず最初のバッファにデータ受信イベントをトリガとしてストアされ(送信者のタイミング)、次に自分のループのタイミングに同期して次段バッファに再度ストアされる。この2段サンプリングにより、フレーズは送り手と受け手のタイミングの非同期を吸収して、確実に「遅れて」再現される。この部分が「1小節遅れ音楽」のキーとなる部分である。

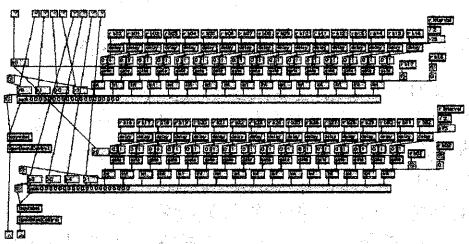

図11. パケット送信サブパッチ

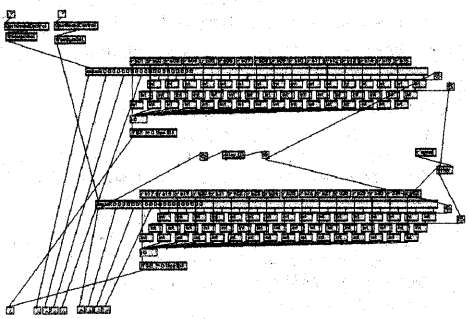

図12. パケット受信サブパッチ

3-5. プロトコルの一例

今回の試作ソフトウェアは1パケット20データの整数、という枠組みでプロトコルを構成した。冒頭4データは未定義、その後に16分音符単位での音高情報があり、「データ=0」は「発音ナシ」と定義しただけである。冒頭4データ部分に、以下のような情報を送信部だけ定義して送っているが、これをこの試作ソフトウェアに

おいて受信側で利用すること(BGMパートの生成サブパッチ内でランダムにより設定している選択番号のところにつなぐだけで完了)はきわめて容易である。

データブロック(A) --- 前半
(data 01)データ種別　今回はダミーとして「ゼロ」
(data 02)ベースパターン番号
(data 03)コードパターン番号1
(data 04)コードパターン番号2
(data 05)-(data 20)前半サンプル・フレーズデータ16個
データブロック(B) --- 後半
(data 01)データ種別　今回はダミーとして「ゼロ」
(data 02)コードパターン番号3
(data 03)コードパターン番号4
(data 04)ドラムパターン番号
(data 05)-(data 20)後半サンプル・フレーズデータ16個

最大公約数としては、今回の試作システムの2倍とか4倍程度のサイズで十分であると思われる。中身についてはあまり固定しないで、複数のパターンから冒頭の「データ種別番号」により選択、というような体系が柔軟であろう。

3-6. プロトコルと音楽モデルとの関係

試作ソフトウェアにおけるプロトコルが上記のように比較的自由なものであるのは、GDSMを具体的に構築する上で、その音楽演奏モデル、セッション・エンターティメントモデルによってプロトコルの具体的な形式が大きく変わるからである。従って、先頭データの番号(数値)によってこのモデルを選択することで、可変長のパケットを全て定義として盛り込めるようにしておく(Javaのバイトコードでも SMFでも基本的な発想は全て同様)ことが重要である。以下に示すのはその一例である。

プロトコルタイプ例[1]（今回の試作モデル）
- ループは2小節パターンとして回る
- それぞれにBGMが走り、playerのアドリブフレーズを交換する
- それぞれのBGMパターンはplayer内部で自由に変化するとともに、互いにパターンのバリエーションを交換できる
- playerのアドリブフレーズは16分音符の単位でサンプリングされ他Playerに伝送される。自分のフレーズと他playerのフレーズとがそれぞれの音楽の中でミックスされるが、他playerの演奏は実は自分の時間よりも1ループ以上、前のものである

プロトコルタイプ例[2]
- ループは4小節/8小節パターンとして回り、コード進行を伴う
- playerのアドリブフレーズだけでなく、コード進行の情報を交換して互いに影響しあうことを楽しむ。コード進行の情報はある時にはリハーモナイズ、ある時には転調としてそれぞれのplayerにおける音楽を変化させる
- playerのアドリブフレーズは16分音符の単位でサンプリングされ他Playerに伝送され交換される。同じフレーズであっても、その時のコードが異なることでスケール感が変化し、それぞれのplayerのもとで実現される音楽は「ほぼ同時にセッションして生成した」音楽であるにもかかわらず、それぞれ別個のオリジナルとなる
- このセッションに参加せず「聞く」だけの人もまた新しい音楽を聴取できる

プロトコルタイプ例[3]
- playerは完全に対等でなく、一人の「指揮者」を位置付ける。「指揮者」は自分のフレーズを同時に演奏してもいいし、以下のパラメータ制御だけをするリスナーであってもよい
  - 「指揮者」はそれぞれのplayerのBGMパターンの選択・生成パラメータを指定して送ることで、全体の音楽に関与できる
  - 「指揮者」は、今回の試作モデルでは固定であったテンポ情報を変化させて、全playerのシステムのテンポを制御できる
  - 「指揮者」は、試作モデルでは固定であったトランスポーズ情報を変化させて全playerのシステムの転調・移調を制御できる

4. ネットビジネスの可能性

GDSMが一般的なプラットフォーム上で稼動し普及した場合には、一種のネットビジネスの可能性がある。例えば、遅延のあるセッションを行う、そのバッキングパターンとしてPOPなものを著明なミュージシャンに作曲・編曲してもらってサーバに格納し、囲い込んだユーザに有償で提供するという、着メロと同様のビジネスがある。さらに、セッションによって生成された音楽は個々のplayerごとに、さらにネットの遅延や生成パラメータをカスタマイズすれば視聴者ごとに異なる音楽であり、それらが全てオリジナルである。この「音楽スナップ」をMIDファイルやMP3化してオリジナルの音楽やサウンドとして有償販売する事も考えられる。

参考文献

[1] Y.Nagashima : Multimedia Interactive Art : System Design and Artistic Concept of Real-Time Performance with Computer Graphics and Computer Music, Proceedings of Sixth International Conference on Human-Computer nteraction (ELSEVIER) 、1995年
[2] 長嶋洋一 : アルゴリズム作曲 ; コンピュータと音楽の世界, 共立出版, pp.402-418, (1998)
[3] 長嶋洋一 : コンピュータサウンドの世界, CQ出版, pp.1-180, (1998)
[4] Yoichi Nagashima : Real-Time Interactive Performance with Computer Graphics and Computer Music ; Proceedings of the 7th IFAC/IFIP/IFORS/IEA Symposium on Analysis, Design, and Evaluation of Man-Machina Systems, IFAC, pp.83-88, (1998)
[5] 後藤真孝, 他 : MIDI制御のための分散協調システム - 遠隔地間の合奏を目指して ; 情報処理学会研究報告 Vol.93,No.109, 情報処理学会, pp.1-8, (1993)
[6] Masataka Goto, et al : RMCP - Remote Music Control Protocol -- Design and Applications ; Proceedings of 1997 International Computer Music Conference, ICMA, pp.446-449, (1997)
[7] 後藤真孝, 他 : RMCP - 遠隔音楽制御用プロトコルを中心とした音楽情報処理 ; 情報処理学会論文誌 Vol.40, No.3, 情報処理学会, pp.1335-1345, (1999)
[8] 長嶋洋一, 他 : "Improvisession" - ネットワークを利用した即興演奏支援システム ; 情報処理学会研究報告 Vol.97,No.67, 情報処理学会, pp.25-30, (1997)
[9] 中村文隆, 長嶋洋一 : 多数決制を取り入れたリアルタイムネットワークセッション --"Improvisession"から"Democratyrant"へ-- ; 情報処理学会研究報告 Vol.98,No.74, 情報処理学会, pp.87-94, (1998)
[10] 後藤真孝, 根山亮 : 不特定多数による遅延を考慮した遠隔セッションシステム ; 情報処理学会研究報告 Vol.98,No.74, 情報処理学会, pp.95-102, (1997)
[11] 後藤真孝, 根山亮 : Open RemoteGIG - 遅延を考慮した不特定多数による遠隔セッションシステム ; 情報処理学会論文誌 Vol.43,No.2, 情報処理学会, pp.299-309, (2002)
[12] Yoichi Nagashima : "IMPROVISESSION-II" : A Perfprming/Composing System for Improvisational Sessions with Networks ; Proceedings of International Workshop on Entertainment Computing, IFIP/IPSJ, pp.237-244, (2002)
[13] http://cnmat.cnmat.berkeley.edu/OSC/
[14] W.Matthew et al. : OpenSound Control - A New Protocol for Communicating with Sound Synthesizers ; Proceedings of 1997 International Computer Music Conference, ICMA, pp.101-104, (1997)