

マルチメディア生成系における プロセス間情報交換モデルの検討

長嶋洋一* 由良泰人** 藤田泰成*** 片寄晴弘* 井口征士***

* イメージ情報科学研究所

** 京都芸術短期大学

*** 大阪大学

プロセス間情報通信IPC(Inter-Process Communication)に関して、アートの視点から検討を行った。マルチメディアアート生成支援環境の研究として、個々のワールドモデルに従ったサウンド生成系、ビジュアル生成系、制御系などの複数のプロセスが、有機的に相互作用するためのモデルを検討した。

具体的には、多対多コネクションによるマルチエージェントモデル、「黒板」や「場」を介在させたモデル、促進と抑制の相互フィードバックモデル等について、具体的な生成実験まで含めて考察した。あわせて分散環境における実現という視点から、ネットワークを利用したシステム化の検討を行った。

A Study of Models for Inter-Process Communication in a Compositional Environment with Multi-Media

Y.Nagashima*, Y.Yura**, Y.Fujita***, H.Katayose*, and S.Inokuchi***

*Laboratories of Image Information Science and Technology,

Kyoto College of Art, *Osaka University

This paper is intended as an investigation of a study for inter-process communication in a compositional environment with multi-media. This project will be open to the public and will be provided for artists who are creating multi-media arts. The platforms are SGI workstations connected with networks and MIDI. We report on the latest situation of the development, and discuss some kinds of models for inter-process communication.

1. はじめに

本研究ではコンピュータ音楽(Computer Music)創造環境の構成要素として、これまでにGranular Synthesisとニューラルネットワーク[1]、カオス[2]、アトラクタ[3]、マルチメディア[4]等について検討するとともに、具体的な作品として実験的な応用を試みてきた[5]。そして新たな段階として、「汎用の芸術創造／表現のためのプラットフォーム」の実現を目標とした[6]。これは、音響のモデルと画像のモデルとを対等なオブジェクトとして統合的に駆動する、ネットワーク化されたオブジェクト指向型マルチメディア情報生成環境の構築を目指すものである。従来の研究においても、音響のモデルと画像のモデルとを結びつ

けるアプローチは行われてきた[7][8]が、本研究においては、ターゲットを「汎用の芸術創造／表現のためのプラットフォーム」と規定する際にキーワードとして「目で聴き、耳で観る」という、本質的にマルチメディアなコンセプトを掲げ、図1のようにアルゴリズム作曲の考え方でシステム内の情報を扱う。

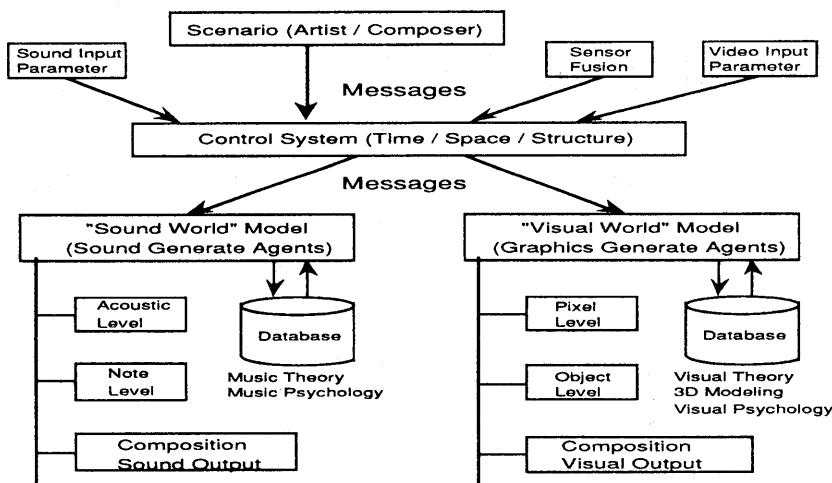


図1:システムの概念図

2. システムの概要と動作

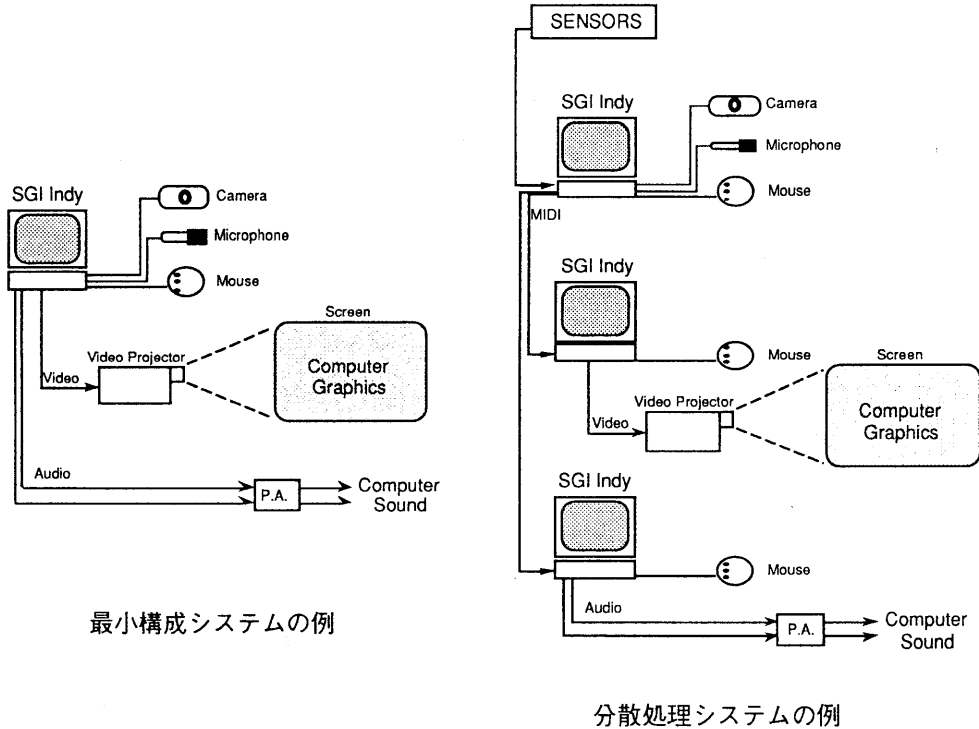
システムを実現するためのプラットフォームとしては、コンピュータ音楽研究の分野で世界的に標準となりつつあるSGI社のWSによる分散処理を採用した。具体的には、イーサネットおよびMIDIによる情報ネットワークを利用して、複数のIndigoやIndyによってサウンド系およびビジュアル系の処理をリアルタイムに並列処理する(図2)。SGIマシンはDSPによるサウンド処理、MIDIによる情報通信、Open-GLによるグラフィクス処理など、汎用の環境としての本システムには好適であり、いずれプラットフォームとしてシステムを公開し、アーティスト等に作品創造の環境を提供していく、という本研究の目標にとっても有意義なものであると考えている。

システムの動作としては、まず時間芸術である音楽や映像という感性情報を時間的空間的に制御するコントロール系に対して、入力情報として楽譜・コマ割り・Sequence Data等の「古典的パフォーマンス情報」と、刻々とサンプリングされる音響/画像や動作/操作のセンサ出力等の「リアルタイム入力情報」とをメッセージとして与える。コントロール系では、これらの情報を統合的に取り扱うアルゴリズムである「シナリオ」(芸術的ワールドモデル)によってリアルタイム処理し、サウンド系およびビジュアル系のプロセス群にメッセージとして送出する。物理モデルや心理モデル等に従うサウンドやグラフィクスの生成処理の部分は、システム内で自由に生息するマルチエージェントとして配置される。ここでは各々の世界モデルに従い、たとえばサウンド系では「個々の音響レベル」「古典的なNoteレベル」「より高次の音楽プリミティブ」等の階層ごとに、コントロール系からの時間情報と同期して、音楽を構成する要素を並列処理し独立に生成する。

3. 開発環境とマルチメディア処理

システムの開発環境としては、SGI社の提供するIDO(IRIX Development Option)およびIRIX Media Libraryを使用した。ここにはX-Window環境としてXlib、X Toolkit、OSF/Motifが標準サポートされており、GUIとしてMotifの優れた「道具」を利用できるだけでなく、後述するX-Windowの「Atom」によるプロセス間通信によってパラメータ交換を容易に実現できる。また、SGI Indyが標準装備しているメディア機能(動画/静止画カメラIndyCam、サンプリング用マイク、グラフィクスエンジン、オーディオ処理DSP等)を、

Cプログラム用のライブラリによって簡単に利用できる。現在のところ、Media Libraryの中で具体的に利用しているのは、オーディオ、MIDI(バージョン5.2にはバグがあるためカスタム化した)、オーディオフィール、ビデオ、IndigoVideo等のライブラリであり、今後はムービー、Compression等のライブラリも用いる予定である。



最小構成システムの例

分散処理システムの例

図2:システムの構成例

デジタル・オーディオ・ライブラリを例にとると、従来であれば複雑なDSPマイクロプログラミングを必要としたリアルタイムのデジタル信号処理が、UNIXのタイミングを考えずに単にライブラリ関数の呼び出しで実現でき、すでにGranular SynthesisおよびGranular Samplingを実験した。また、やや高速のシリアル通信であるMIDI処理については、特に送信については完全に「パソコン感覚」で処理でき、受信についてはSensor系からのトラフィックを制限することで対応している。今後はさらにIRCAMの提供するSGI-MAXの環境ともリンクしていく予定である。

ビデオ映像と共に重要な視覚的要素であるComputer Graphicsについては、IRIX-GLおよびOpen-GLがサポートされているために、従来であればコマ単位に非実時間処理していたCGが実時間で生成され、この描画処理のパラメータに対してMIDIやSensorの情報を反映させることで、リアルタイムのマルチメディア・パフォーマンスが実現できる。GLの環境はX-WindowのXGLというIntrinsicsによって完全に取り込まれ、統一された環境で開発できる。

4. プロセス間の情報交換モデル

4.1 プロセス間通信 IPC (Inter-Process Communication)

マルチタスクのUNIX環境では、種々の処理を複数のプロセスとして記述して、それらに相互作用させるシステムが有効である。ここで問題となるのが、複数のプロセスの間での情報交換の手法である。筆者のうちの一人が従来行ってきた一つの方法としては、図3のように、複数に分割したシステム要素の間での情報交換はすべてMIDIを使用する、というものがあつた[4]。この方法の利点としては、(1) プロトコルを特別に規定することでオーバーヘッドの不要な生データの通信を行える、(2) ケーブルの増設や分配が容

易、(3)MAXを用いることでデバッグやプロトタイプングが容易、などがある。ここでは図4のように、MIDIデータの言葉を用いてMAXで情報のリンクを制作している。しかし、UNIX環境とMIDI通信との相性はそれほど密接ではなく、この方法は最善のものとは思われない。

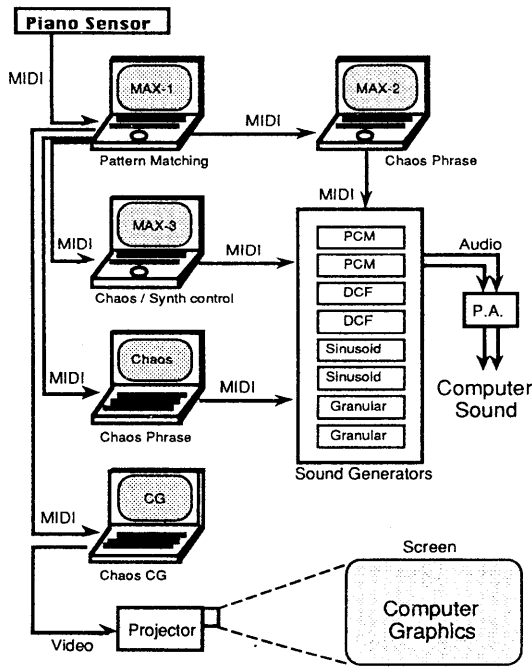


図3:MIDIを用いた実例
(作品 "Strange Attractor")

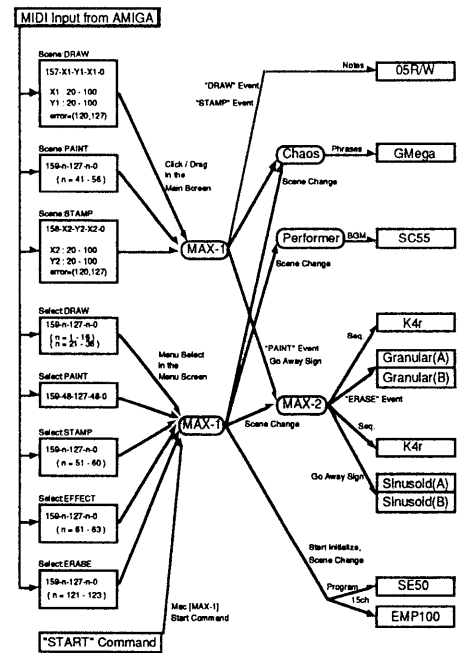


図4:情報リンクの実例
(作品 "Muromachi")

そこで本研究ではまず、UNIXの提供するいくつかのプロセス間通信の手法として、(a) 半二重パイプ、(b) FIFO、(c) 全二重ストリームパイプ、(d) メッセージキュー、(e) 共有メモリ、(f) クライアント・サーバ、という6種類について調べた。その結果、(a)はデータ方向の制限、(b)(c)はファイル経由による処理の遅さ、(d)は蓄積されて消えない情報、(e)はデータ参照の同期処理の必要性、という問題点があり、現実的には(f)のクライアント・サーバ方式、具体的にはX-Windowの「アトム」を用いたクライアント間の通信を採用した。ここではサーバはすべてのプロセスに共通のX-Window環境そのものであり、各プロセスは「プロパティ」と呼ばれるデータ集合を独自に定義して、自在に情報交換することができる。IPCの機構としては、情報提供側の各プロセスは起動時に生成したアトムによって、サーバのプロパティに刻々とデータを送る。情報を受ける側は、同じアトムの宣言によってIDを与えられ、実際にはコールバック関数においてPropertyNotifyイベントハンドラによって、データに変化があれば自動的に処理が呼び出される。これはイベント検出ループを監視し続ける古典的なCプログラムの処理方法よりも記述が容易であり、構造化に適している。

4.2 「多対多」モデル

複数のプロセス間の情報交換モデルとしてもっとも一般的なのは、図5上図のように、それぞれのプロセス間にすべて情報交換のリンクを張ることである。実際には、図5下図のようにA、Bという2つのプロセスで双方向の情報交換をする場合にも往復で2つのルートを確認する必要があり、プロセスの数が増加するに従ってリンクの数は非常に多くなるという欠点がある。また、次々とプロセスが生成されたり消滅するような動作における、「動的なプロセス管理」の方法も複雑になる。

このような問題はニューラルネットワークにおいても課題となったものであり[1]、機械的な結合でないエージェントの相互作用を目指す本システムにおいては、あまり得策ではない。しかし複数の要素がそれぞれカオス的な振舞いをするシステムにおいて、このような相互結合によって全体の系が大局的に興味ある挙動をする例もあり[2]、音響プリミティブ生成などの階層においては、採用の可能性も検討している。

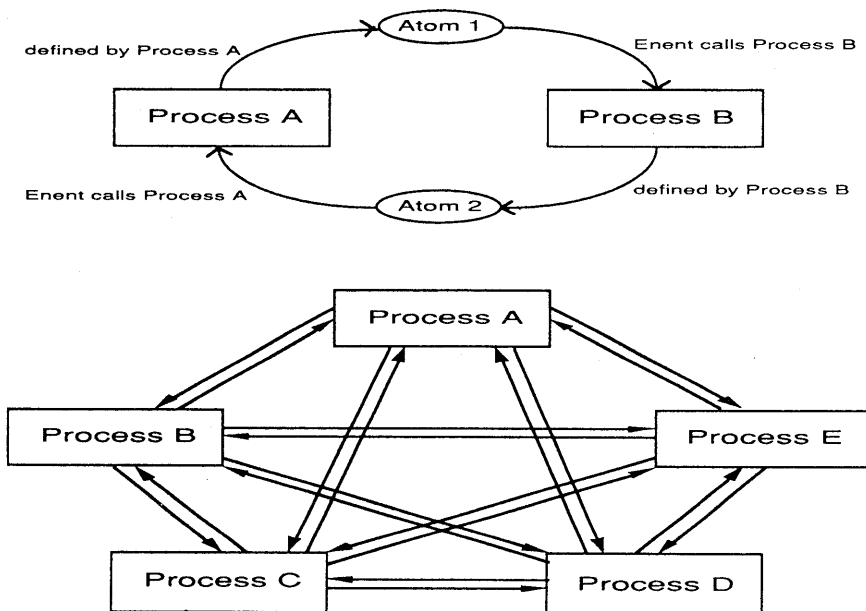


図5: 「多対多」モデルの構成例

4.3 「黒板」モデル

「多対多」モデルのようにすべてのプロセスを対等に扱う場合の無政府状態を避け、さらに独裁的な指導者プロセスを置かず民主的な相互作用空間を実現するためには、パラメータをそれぞれが書き込み、さらにそれぞれが読み出す、という「黒板」のような情報共有システムを置くことが有効である。ここでは「黒板」という受動的な存在であることが重要であり、あれこれと「命令」せず、単に情報を共有するだけの「消去可能な掲示板」に徹するところがポイントである。図6は、このような「黒板」モデルの例である。

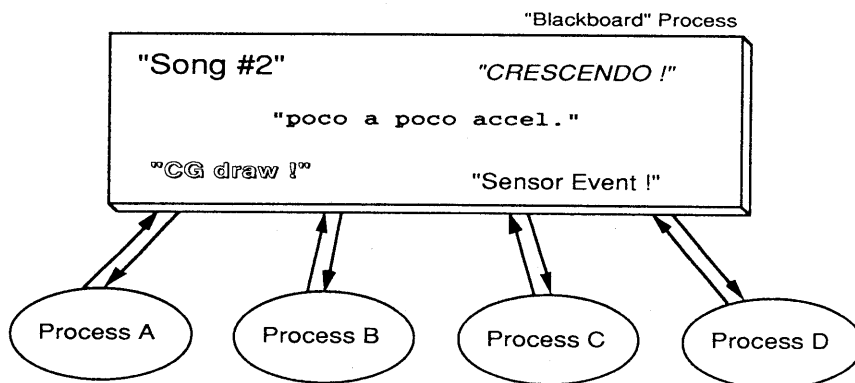


図6: 「黒板」モデルの構成例

ここでは、音楽構造や音楽演奏に関する情報を提供するプロセスからの情報が書き込まれたり、書かれた情報から音楽生成のプロセスが必要なパラメータを得るために機能する。この図のシステムでは、見たところ単純に「共有メモリ」で情報を交換する方法とあまり違いがないが、図7のようにネットワーク化した場合に非常に有効になる。すなわち、ネットワークを介して遠隔地の「黒板プロセス」同士が情報交換を行うことで、異なった場所のプロセスの間でも有機的な相互作用を行うことが可能となる。

なお「黒板」モデルにおいては、個々の情報の同期管理や個々のプロセスのスケジューリングが重要であるが、この制御系については現在のところ実験中であり、当面はUNIXのプライオリティ管理を利用している。

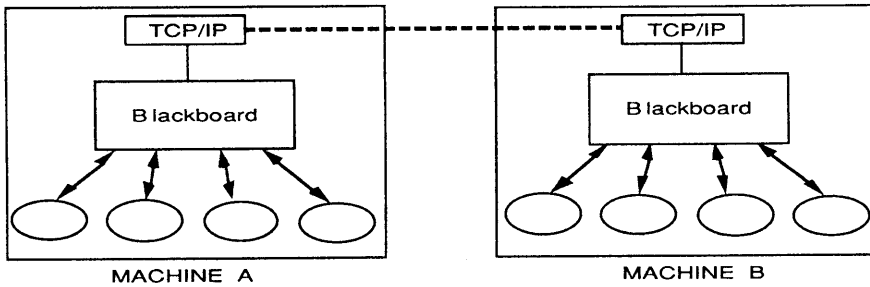


図7: 「黒板」モデルによる分散処理

4.4 「場」モデル

「黒板」モデルが完全な情報の掲示板であるのに対して、よりシステムを現実の世界モデルとして機能させるために検討しているのが、電磁場や重力場と同じように一種の「音楽パラメータの場」を構築する方法である(図8)。各プロセスは、自律的な振る舞いの中で発生した情報を「場」に供給するが、この部分は「黒板」モデルにおける書き込みと同様である。しかし複数のプロセスから「場」に与えられた情報は「環境へのAction」として累積され、たとえば「各プロセスからの音量が閾値を越えたので decrescendoすべきである」「盛り上がり状態が続いたのでそろそろ落ちつくべきである」というような共通のフィードバック出力を生成する。ただしこの出力情報はあくまで「参考意見」である。

このような「場」は、黒板ほど受動的ではなく、「指揮者」ほど威圧的でもない。すなわち、環境として各プロセスからの影響を総合してフィードバックしているだけであり、このフィードバックをどのように解釈して対応するか、は各プロセス内のモデルに依存している。また、「黒板」モデルでは各プロセスが「黒板に書かれた情報の解釈と判断」を行う必要があるのに対して、「場」が共通の解釈を代行してくれるために、処理の負担が軽減される。これは、アンサンブルを行う個々の楽団員が完全に指揮者と同じように演奏をフォロー(解釈/判断)しているわけではない、という現実にも即している。さらに、個々のプロセス内部のルールをかなり単純化しても、このようなフィードバックによって系全体が複雑なカオス的挙動を簡単に起こすことが知られており、今後の実験テーマの一つと考えている。

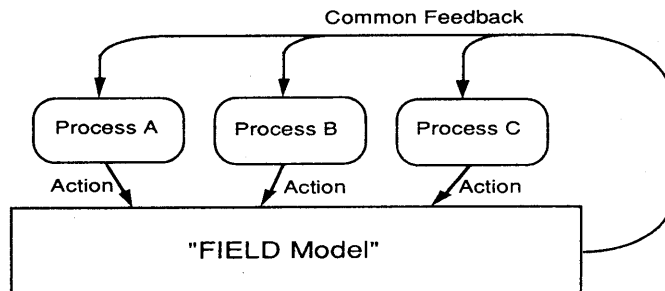


図8: 「場」モデルの構成例

4.5 「相互フィードバック」モデル

「場」のモデルをより現実的なものとしたモデルとして、図9のように相互に逆方向の働きを持つ二つの「場」を組み合わせた「相互フィードバック」という機構を検討している。これは、音楽感性情報処理においては、促進的および抑制的という相反する感性パラメータが数多く登場する[9]ことに対応している。実験中及び構想中の具体的な例としては、

- ・「上昇音型で音量が増加する(一般的傾向)」と「下降音型で音量が増加する(バッハ的傾向)」
 - ・「テンポアップによって音量が増加する」と「テンポダウンによって音量が増加する」
 - ・「テンションノートが増えると音数が増える」と「コードノートが増えると音数が増える」
- 等がある。

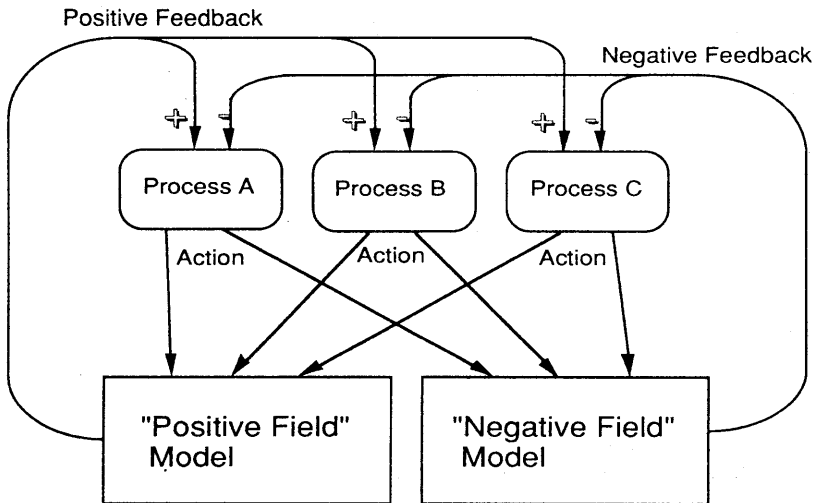


図9: 「相互フィードバック」モデルの構成例

またこのようなモデルにおいては、関係する複数のパラメータを「逆方向」の性質に限る必要はなく、「演奏の盛り上がり(テンションパラメータ)」と「音色の変化」とか、「カオス音列生成の音域」と「カオス状態のフラクタル次元」など、アルゴリズム作曲に関連した、感性空間での直交関係が検証されていない多くの実験課題が考えられている。さらに本研究のテーマであるメディア間における関係づけとして、「音楽系のテンポ」と「グラフィクス系の輝度」とか、「音楽系の和声」と「グラフィクス系の色彩」とか、「音楽系のテンション」と「グラフィクス系の動き」など、アートのモデルとして検討してみたい課題は多く、今後、具体的に実験していく予定である。

5. コントロール系についての検討

本稿で紹介したのは、主にサウンド生成系及びグラフィクス生成系における各種の処理を実現するプロセスについての情報交換モダルの検討であった。しかし、システムの実現において重要な課題となるのが、図1において「Control System」とある、「アートのシナリオ」に従って全体を制御するプロセスである。現在のところは、プロダクションシステムとして記述する方法を検討しているところであるが、この部分は本研究の「プラットフォームとしてシステムを公開し、アーティスト等に作品創造の環境を提供する」という目標からも、十分に検討する必要がある。つまり、プログラミングなど計算機の専門知識を要求しないユーザ・インターフェースが重要であり、広義のHuman Computer Interfaceを提供することが必須だからである。

プログラミング言語による記述でない「シナリオ」の表現方法として、具体的には「テキストファイル形式のプロセス群リスト」「OSF/MotifによるボタンやスライダーのGUI」「時間軸に沿ったタイムチャート形式」などを実験的に試作しているが、最終的には「MAXライクなGUI」を実現する必要性についても

検討している。もはやMAXは一つのツールというよりも「文化」として定着しており、世界的な研究の傾向としても、GUIの規範的概念として尊重する流れにあるからである。

6. おわりに

UNIXの世界としては一般的であるプロセス間情報通信IPC(Inter-Process Communication)について、アート的な応用の視点から検討を行った。マルチメディアアート生成支援環境として、個々のワールドモデルに従った生成系や制御系などの複数のプロセスが有機的に相互作用するためのモデルとして、具体的には「多対多」モデル、「黒板」モデル、「場」モデル、「相互フィードバック」モデル等について考察した。

本研究は「汎用の芸術創造／表現のためのプラットフォームの実現」を目標にしており、成果はステップごとに具体的な作品として発表し、実験と検証を並行していく予定である。現在、1995年10月20-21日に京都で開催される「日独アートフェスティバル」(後援：情報処理学会)で最初の実験的作品を発表する予定である。ここでは本システムを具体的に活用した、リアルタイムにサンプリングされたビデオ映像を用いた画像Sensor、パフォーマーの動作を検出するSensor群、Computer Musicシステム、Computer Graphicsシステムなどが統合された、マルチメディア・インタラクティブ・パフォーマンスを実現するための開発が進んでいる。

また、ある程度システム開発が進んだ段階では、さらに一般のアーティストに公開して評価を求め、あるいはパフォーマンス形式でなく、シナリオに従って自動的に人間に対応する「インタラクティブ・インストール」システムとしての応用についても、併せて検討していきたい。

参考文献

- [1] Y.Nagashima : Real-Time Control System for "Pseudo" Granulation. Proceedings of ICMC, pp.404--405, 1992.
- [2] Y.Nagashima, H.Katayose, S.Inokuchi : PEGASUS-2: Real-Time Composing Environment with Chaotic Interaction Model. Proceedings of ICMC, pp.378--390, 1993.
- [3] 長嶋洋一, 片寄晴弘, 井口征士 : Attractor Synthesisによる楽音合成システムの検討. 情報処理学会平成6年度前期全国大会講演論文集I, pp.379--380, 1994.
- [4] 長嶋洋一 : マルチメディア Computer Music作品の実例報告. 情報処理学会研究報告 Vol.94, No.71 (94-MUS-7), pp.39--44, 1994.
- [5] Y.Nagashima : Multimedia interactive art: system design and artistic concept of real-time performance with computer graphics and computer music. Proceedings of HCI International, Yokohama, 1995.
- [6] 長嶋洋一, 片寄晴弘, 由良泰人, 井口征士 : 画像情報と統合化されたコンピュータ音楽創造環境の構築. 情報処理学会平成7年度前期全国大会講演論文集I, pp.363--364, 1995.
- [7] Tapio Takala, James Hahn, Larry Gritz, Joe Geigel, and Jong Won Lee : Using physically-based models and genetic algorithms for functional composition of sound signals, synchronized to animated motion. Proceedings of ICMC, Tokyo, 1993.
- [8] Mon-chu Chen : Toward a new model of performance. Proceedings of ICMC, Aarhus, 1994.
- [9] N.Saiwaki, H.Katayose, and S.Inokuchi : An Approach to a Computer Assisted Composition System with KANSEI Parameter. Proceedings of 2nd International Workshop in AI and Music, 1989.