

7J-05 WWW 上の異なる計算機環境で動作する 音楽セッションシステム

— VirJa Session on WWW へ向けて —

根山 亮 後藤 真孝 村岡 洋一

早稲田大学 理工学部

1. はじめに

近年, 計算機とのジャズセッションを実現するシステムに関する研究が多くなされてきた [1]-[5]. しかし, これらのシステムは, ある特定の計算機環境でしか動作せず, またそれぞれのシステムで用意された特定の計算機演奏者としてセッションを行なうことができなかった.

本稿では, ユーザが新しい計算機演奏者を容易に開発・カスタマイズして WWW 上に公開したり, 計算機演奏者を WWW 上から自宅のパソコンのような身近な場所にダウンロードし, セッションを行なうことを可能とする枠組を提案する. 特にダウンロードした計算機演奏者がセッションをする際に必要となる通信を, 安全性を確保して行なうための方法, 新しい計算機演奏者の開発労力を削減する方法について述べる.

本システムは Java で記述されており, Java 対応の WWW ブラウザ上で動作するので, 計算機環境に依存しない. 我々は演奏者の実装例として, ジャズのピアニストとドラマーを実装した. ユーザはピアノを演奏することにより, 計算機演奏者と即興演奏を楽しむことができる. 実験の結果, 実際に HTTP サーバからダウンロードしたピアニストとドラマーを WWW ブラウザ上で実行し, 人間のピアノの演奏にあわせてリアルタイムに演奏させることができた. また本システムが異なる複数の計算機環境で動作することも確認した.

2. 実現上の課題と解決法

まず, 本システムの主要な実現上の課題を二つ述べ, 次に我々の解決法を示す.

2.1 実現上の課題

(1) 安全性の確保

計算機演奏者がセッションを行なうためには, 入出力の演奏情報を他の演奏者と通信する必要がある. しかしダウンロードした計算機演奏者は信頼できるとは限らず, 機密情報を盗んだり, ポートを不法に占有した

り, ネットワークを停止させるなどの危険性がある. 実際 Java 対応の WWW ブラウザでは, Java アプレットに, ダウンロード元のホストとの通信しか許していない. しかし, 計算機演奏者がセッションを行なう場合, ダウンロード元のホストを介して通信したのでは, 遅延が大きく現実的ではない. そこで, 安全性を確保しながら, 計算機演奏者に, LAN 上で通信させるための方法が必要となる.

(2) 計算機演奏者の開発労力の削減

一般に計算機演奏者を開発する場合, 本質的な演奏理解・生成処理以外に, 演奏情報の送受信やその時間管理などを実装する必要がある. しかし, 演奏者の本質的な部分とは無関係なこれらの処理は, 全ての計算機演奏者に共通である. そこで, 計算機演奏者の演奏理解・生成部の一部, または全部を入れ換えるだけで新しい計算機演奏者を作ることが容易にできる開発環境があるとよい.

また, 計算機演奏者の開発にはその内部状態の変化の確認が不可欠であるため, 内部状態を容易にモニタリングできる設計になっているべきである. 例えば我々の従来のシステム [5] では, 内部状態を表すパラメータの変更箇所すべてに対しモニタリング用のコードを加えなければならず, 開発効率が悪かった.

2.2 解決法

我々は, 図 1 のように計算機演奏者の演奏理解・生成部と演奏情報送受信部を分離することによって, 安全性の確保と開発労力の削減を同時に実現する枠組を提案する.

演奏理解・生成部と演奏情報送受信部は, それぞれスレッドとして非同期に動作し, 共有メモリ (Java の静的変数) を用いて情報を受渡しする. WWW 上からダウンロードする対象を演奏理解・生成部のみにし, ローカルディスク上の信頼できる演奏情報送受信部により通信を行なう (通信には RMCP [6] を用いる) ので, 安全性を確保できる.

また, 新しい計算機演奏者を作る際に演奏理解・生成部のみを入れ換えればよいので, プログラマは必要最小限の開発に専念できる. 演奏者の内部状態をすべて共有メモリに格納することで, 新たにモニタリング用のスレッドを起動すれば, 元のプログラムに変更を

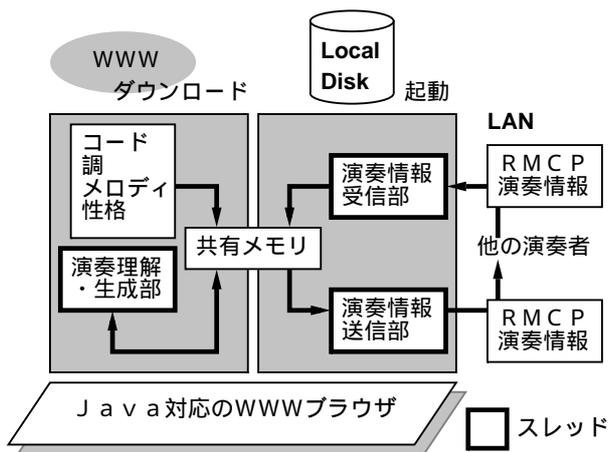


図 1: システムの概要

加えることなく内部状態を確認できる。

3. 計算機演奏者の実装

本システムでは、計算機演奏者の実装例として、VirJa Session (Virtual Jazz Session) [4][5] の計算機演奏者をオブジェクト指向で設計し直し、ジャズのピアニトリオのベーシストとドラマーを実装した。ユーザはピアノを演奏して計算機演奏者と即興演奏する。

本システムを開発環境として利用すれば、計算機演奏者の開発者は、演奏理解・生成部のクラスのメソッドを書き換えるだけで、新しい計算機演奏者を容易に実装できる。例えば、演奏生成部においてリズムや音高を決定するメソッドを、大局的な演奏の流れを考慮して決定するように書き換えることで、異なる演奏をする計算機演奏者を実装できる。

また、共有メモリ上の計算機演奏者の性格を表すパラメータは、ユーザが GUI を用いて演奏前、演奏中を問わず、自由にカスタマイズすることが可能である。現時点で設定できる性格パラメータは、文献 [5] 中の、主張度の初期値、目標主導率、達成満足時間、達成締め期間、主張度を演奏モードへ対応付けるための閾値であり、これらはピアノソロ、ベースソロ、フォーパースなどのシナリオごとに設定できる。これらのパラメータを変更すれば、「常に主導権を握ろうとする演奏者」や「自分がソロの時のみ主張し、それ以外は他者にあわせて演奏しようとする演奏者」などいろいろな性格の計算機演奏者を実現できる。

4. 実験結果

本システムがリアルタイムに動作をすることを確認するため、処理時間を計測した。現在の実装では、計算機演奏者は一拍ごとに処理を行なっているので、計測対象は一拍分の処理時間とした。計測は Sun Ultra 1 (Solaris 2.5) および Netscape Navigator 3.01 を用い、

曲のテンポは 187 (一拍の間隔: 321ms) で行なった。その結果、処理時間はベーシストの場合、平均 7.1ms, 最大 69ms, ドラマーの場合、平均 11.4ms, 最大 105ms であった。これから、本システムはリアルタイムに動作しており、さらに機能拡張の余地があることが分かった。

次に、本システムを実際に運用して、音楽セッションを行なった。対象は、テンポ一定で演奏される 4 ビートのジャズスタンダード曲「Take the "A" Train」とした。実験の結果 LAN 内の HTTP サーバからダウンロードしたベーシストとドラマーを Netscape Navigator 3.01 上で Java アプレットとして実行し、ピアニストのリアルタイムな演奏にあわせて、即興演奏ができた。

また、本システムが異なる複数の計算機環境 (Windows95, Solaris 2.5, IRIX 5.3, Linux 2.0) で動作することも確認した。

5. おわりに

本稿では、Java による異なる計算機環境で動作する音楽セッションシステムについて述べた。本システムにより、誰もが新しい計算機演奏者を容易に開発でき、それを WWW 上に公開したり、WWW 上の計算機演奏者を集めてセッションを行なうことが可能となった。また、計算機演奏者が安全性を確保して通信するための枠組を提案した。

現在の実装では、RMCP による MIDI 入出力の部分は、特定の計算機環境に依存している。今後はこれらの部分もより多くの計算機上で動作させる予定である。また、今回提案した安全性と開発環境の枠組は一般的であるので、他の分野にも応用していきたい。

謝辞 本システムの実験に快く協力して頂いた緒方 美音子氏、数多くの御指導、貴重なコメントを頂いた日高 伊佐夫氏、早稲田大学村岡研究室諸氏に深く感謝する。

参考文献

- [1] 和気他: テンション・パラメータを用いた協調型自動伴奏システム: *JASPER*, 情報処理学会論文誌, Vol.35, No.7, pp.1469-1481 (1994).
- [2] 金森他: ジャズセッションシステムのための音楽認識処理の一実現法, 情報処理学会論文誌, Vol.36, No.1, pp.139-152 (1995).
- [3] 渡辺他: ニューラルネットワークを用いたジャズセッションシステム — ニューロミュージシャン —, 第 44 回情処会大, 4R-5 (1992).
- [4] 後藤他: すべてのプレーヤーが対等なジャズセッションシステム I. システムの全体構想と分散環境での実装, 情処研報, Vol.96, 音楽情報科学 MUS-14-4 (1996).
- [5] 日高他: すべてのプレーヤーが対等なジャズセッションシステム II. ベーシストとドラマーの実現, 情処研報, Vol.96, 音楽情報科学 MUS-14-5 (1996).
- [6] 後藤他: 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 —, 情処研報, Vol.93, 音楽情報科学 93-MUS-4-1 (1993).