

生体情報を利用した人工現実感スキーシステム (2)

— 処理モジュール間の通信と制御 —

7E-4

中村暢達 根本啓次 金子朝男 篠原克也

NEC 情報メディア研究所

E-mail: nakam@JOKE.CL.nec.co.jp

1 はじめに

人工現実感 (VR) システムでは、入力データ処理、仮想世界データ更新、CG 映像生成といった多くの処理を行なうので、1 台の計算機 (特に PC) では負荷が重い。そのため、複数の計算機を用いて負荷を分散させ、協調動作させる分散環境が好ましい [1][2][3]。

分散環境の VR システムでは、各処理がモジュールごとに分割され、分散した各処理モジュールが通信を行いながら協調動作する。また、各処理モジュールを独立に開発でき、システム構成の変更に対応して自由に追加・削除できるようなモジュールの独立性が求められている。本スキーシステムでは、こうした処理モジュールの独立性を特徴とする分散 VR 環境 [4] を適用した。

本稿では、分散 VR 環境における各処理モジュール間の通信方式、及び各処理モジュールを独立に開発し、システム全体を構築する手法について述べる。さらに分散 VR 環境の本スキーシステムへの実装について述べる。

2 分散 VR 環境

2.1 モジュール間通信

従来の分散環境の VR システムにおけるモジュール間通信とは、ネットワーク通信のみが対象であり、その通信方式としてはトークンリング、ブロードキャストなどが用いられてきた。しかし、これらの方式では、RS232C 接続など多様なシステム構成に対応することが困難であったり、ネットワークに負荷をかけるため WAN に対応できないなどの問題点があった。

本スキーシステムに適用した分散 VR 環境におけるネットワーク通信は、通信管理モジュールがサーバとなり、各ネットワーク上の処理モジュールがクライアントとなるサーバ/クライアント方式を用いている。さらに、通信管理モジュールは、ネットワー

ク通信だけでなく、RS232C、プロセス間通信等、すべてのモジュール間通信を集中的に管理する。通信管理モジュールが、種々の接続形態に対応することで、各処理モジュールにとって適した接続形態、計算機環境を選択でき、多様なシステム構成に対応することが可能となっている。

分散 VR 環境は、図 1 に示すように、さまざまなモジュール間の通信接続形態 (Ethernet, RS232C, GPIB, 同一 CPU におけるプロセス間通信, 等)、多様な計算機環境 (UNIX/WS, MS-DOS/PC, MS-Windows/PC, 等) に対応できる。各処理モジュールが通信管理モジュールのみとデータ交換を行うことで、モジュール間通信は実現されている。

2.2 独立モジュールの開発とシステム構築

通常のサーバ/クライアント方式では、各処理モジュールの開発者がサーバ側プログラムを開発し、他モジュールとの協調動作を考慮する必要がある。また接続形態を変更した場合にプログラムも大きく変更する必要があるなど、モジュールの独立性が小さく、開発者の負担は大きい。

そのため、各処理モジュールが共通して用いることができる共有データを用意し、共有データを介して各処理モジュール間の通信や協調動作を実現する。そして、分散 VR 環境としては、共有データにアクセスするための関数ライブラリ (C 言語) を提供して

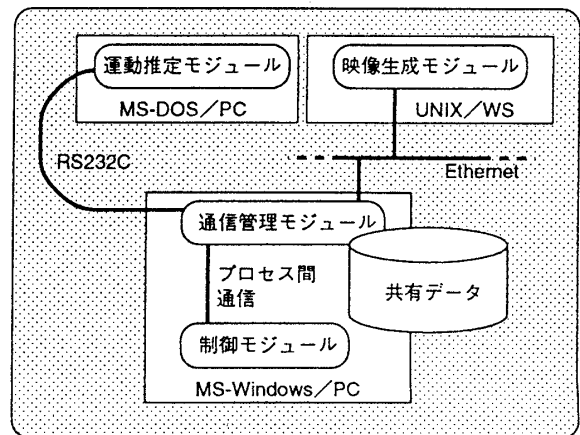


図 1: 分散 VR 環境における計算機環境

A Virtual Reality Skiing System using Physiological Data - Network communications and management for distributed modules -

Nobutatsu NAKAMURA, Keiji NEMOTO, Asao KANEKO, Katsuya SHINOHARA

Information Technology Research Labs., NEC

いる。

各処理モジュールは、共有データを参照することで、他のモジュールと協調動作する。他のモジュールと直接通信を行わないので、各処理モジュールの独立性が高く、開発が容易となる。いくつかの関数を例として示す。

- InitNewSharedData(short key\_id, long size)  
共有データの識別子と大きさ(バイト数)を引数にして共有データ領域の確保を行う。
- GetSharedData(short key\_id, char \*addr, long size)  
共有データの識別子、格納するバッファアドレス、大きさを引数にして共有データ領域のデータを得る。
- PutSharedData(short key\_id, char \*addr)  
共有データの識別子、共有データ領域に置きたいデータのバッファアドレスを引数にして、データを共有データ領域に置く。

これらの関数は、通常のプログラムのメモリ確保、コピーと見かけ上同じように機能するが、実際には通信管理モジュールと通信を行い、共有データ領域の確保、送受信などを要求する。この時、開発者は各処理モジュールがどのように通信管理モジュールと接続されるかは考慮しなくてよい。プログラム作成時にリンクする関数ライブラリを接続形態によって変更するからである。システム構成を変更した場合でも、プログラムの変更は非常に少ない。したがって、各処理モジュール開発者は、プロセス間通信、ネットワーク通信といった処理に煩わされることなく開発を行うことができる。

### 3 分散 VR 環境の実装

VR スキーシステムの処理モジュール基本構成は、CG 映像生成 (CG)、データベース管理 (DB)、重心解析・運動推定 (motion)、運動評価 (judge)、音響生成 (sound)、システム制御 (control)、通信管理 (manager) を 4 台の PC に分散させた構成になっている。図 2 に示すように、基本構成から、高品質 CG グラフィックスを得るように GWS を用いた構成に変更したり、CG、DB、motion、judge の各処理モジュールを 2 セット用意し、ランデブースキーに拡張することも容易である。

本システムでは、制御情報、斜面情報、運動情報、評価情報の 4 つを共有データとして用いている。分散 VR 環境は、汎用的に設計されているので、スキー VR システムとしての制御機能は持っていない。各処理モジュールの動作制御のためには、共有データの制御情報を用いる。

制御情報には STATUS と MODE があり、STATUS は、モード選択、キャリブレーション、レディ、ランといったシステムの状態を示し、MODE は初級・中級・上級、タイムレース・ランデブースキー

といった難易度や種別を示す。これらの情報は主にシステム制御モジュールによって管理される。各処理モジュールは、この制御情報を参照し、斜面情報、運動情報、評価情報を処理し、処理モジュール間の協調動作を実現し、VR スキーシステムを構成する。

### 4 おわりに

VR スキーシステムのモジュール間通信を実現するために、分散 VR 環境を用いた。分散 VR 環境では各処理モジュールは通信管理モジュールとのみ通信を行い、また各種のモジュール間通信の手順が隠蔽化されるので、各処理モジュール開発が容易となる。また、処理モジュールの独立性により各処理モジュールの追加や変更が容易となるので、複数の滑走者が同時に滑走動作を行うランデブースキーなど、様々なシステム構成への柔軟な対応が可能である。

### 参考文献

- [1] The Cognitive Coprocessor Architecture for Interactive User Interface, G.G. Robertson, Xerox, UIST'89
- [2] Dialogue Structure for Virtual Worlds, J.B. Lewis, IBM, CHI'91
- [3] The Decoupled Simulation Model for Virtual Reality Systems, Chris Shaw, Univ. of Alverta, CHI'92
- [4] 分散型仮想現実システム構築環境の一検討, 中村暢達, 第 4 6 回情報大全, 1994.3

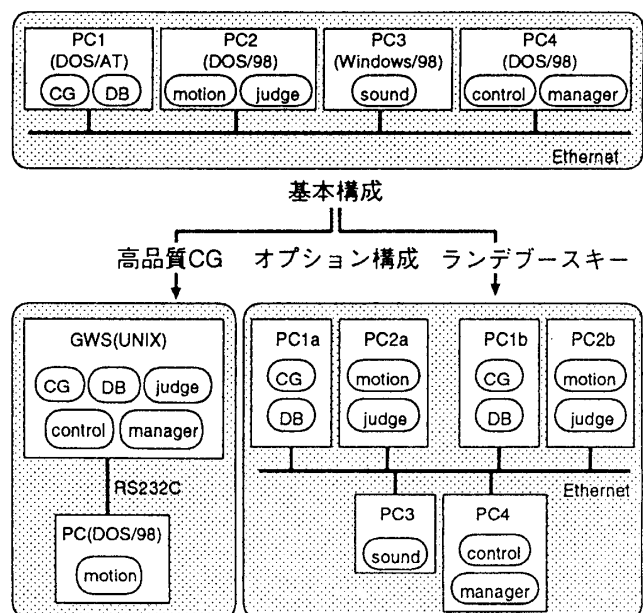


図 2: システム構成拡張例