総説論文

TVRSJ Vol.5 No.4 pp.1071－1078, 2000

多様なインタフェースデバイスを統合するソフトウェアアーキテクチャ IOA (Interaction-Oriented Architecture)

岩田洋夫[*1]

IOA: A Software Architecture for Integrated Interface Devices

Hiroo Iwata[*1]

Abstract This paper describes software architecture for integration of interface devices. Interface devices, such as haptic interface, are implemented in various configurations. In most cases, software of virtual environment is tightly connected to control program of those interfaces. This problem is a hazard for development of advanced applications. This paper proposes a modular software architecture called IOA (Interaction-oriented Architecture), which supports various types of interface devices. Sensors and displays are easily reconfigured by exchanging modular software. The IOA also supports tools for content development. Implementation of IOA is exemplified through audio haptics and networked haptics.

Key Words: interface device, software tool, modular software, virtual environment

1. はじめに

バーチャルリアリティが他のメディア技術と最も異なる点は、人間の様々な感覚器官に対して合成的情報を呈示することにある。そのため、この研究領域では多種多様なインタフェースデバイスが開発されてきた。人間は豊かな感覚受容能力を有しているため、これを活用するようなメディア技術を開発することは非常に重要である。従来コンピュータと人間の接する場はキーボードやマウスといった極めて狭い感覚チャネルを通じて行われていた。現在標準となっているこれらのインタフェースデバイスは人間の知的活動を制限しているとみなすことができる。つまり、インタラクションを行う環境が、記号的には複雑だが物理的には単純という歪を包含している。自然界で人間が活動する場は物理的な複雑性をもっており、それが人間に智恵をもたらしてきた。メディア技術を用いた知的活動も同様に物理的複雑さが必要である。

インタラクションの場に物理的複雑さをもたらすためには、人間の身体性に対応した多種多様なインタフェースデバイスを協調的に動作させることが必要である。そのためには、ハードウェアの開発に歩調を合わせたソフトウェア基盤の整備が不可欠である。本論文はこのような視座に立って、著者の経験をベースにＶＲのソフトウェアアーキテクチャのあるべき姿について論じるものである。

*1 筑波大学 機能工学系
*1 Institute of Engineering Mechanics and Systems, University of Tsukuba

2. VR用基本ソフトウェアの展開

米国ではバーチャルリアリティの技術は、主としてソフトウェアの問題としてとらえられている。これまでに研究機関におけるプロトタイプや、市販ソフトウェア、フリーウェア等が多数作られてきた。それらを網羅することは本論文の目的ではないが、後述する技術課題の背景となる主たる動向を概観してみたい。

VR ソフトウェアのルーツは米国の軍用シミュレータをネットワーク化するための SIMNET に見出すことができる。SIMNET は１９８０年代中盤に米国陸軍によって開発されたもので、複数の戦車シミュレータを通信回線で接続し、低コストで陸上戦の模擬を行うことを目的としていた[1]。９０年代に入ってこのプロジェクトは海軍大学院大学における NPSNET に展開した[2]。NPSNET は歩兵をも含めた多数の戦闘員が参加することを想定した戦闘シミュレータであり、これがＶＲソフトウェアの一つの基本構造を与えることになった。NPSNET の提案以降複数ユーザーによるＶＲ空間の共有はこの研究領域の一つの主流になり、大学等の研究機関において

研究が進められ、Paradise、DIVE、BrikNet 等を始めとして多くのプロトタイプが報告されている[3]。

９０年代の初頭は商用ＶＲソフトウェアが誕生した時期でもあり、Sence8 社の WorldToolKit [4] やオートデスク社の Cyberspace Developer Kit [5] 等が開発された。これらのソフトウェアは市販ＨＭＤと３Ｄポインターによるインタラクションを前提に、ＶＲ空間のモデリングが容易にできることを目指していた。両ソフトウェアはその後のビジネス展開で大きく明暗が分かれ、前者が生き残り、後続の市販ソフトウェアの範となった。一方日本では１９９１年に廣瀬らがバーチャル物体のモデリングに加えて、それに挙動を与えるシミュレーションエンジンを付加した Vis-Age を開発した[6]。

この時期、著者はハプティックインタフェースのハードウェアに関する研究に腐心していたが、アプリケーションを高度化する上でソフトウェアツールの必要性に迫られていた。力覚フィードバックを行うソフトウエアはハードウエアに密着していたため、プログラムの再利用が難しかった。このため、新たに製作したハプティックインタフェースに入れ替える場合や、新たなアプリケーションを開発する場合に振り出しから始めなければならない、といった多大な労力が必要となり、高度なシステムを組むことの障害となっていた。そこで、著者はソフトウェアの主要な部分をモジュール化し、力覚ディスプレイのハードウェアや、頻繁に使うソフトウェア部品を交換可能にした。このようなモジュール化されたソフトウェアアーキテクチャにＶＥＣＳという名前を付けた[7]。これが本論文で述べるＶＲソフトウェア設計思想の出発点である。

１９９０年代中盤以降になるとＶＲソフトウェアは様々なものが開発されるようになった。まず、この時期の最大の特徴は、ハプティックインタフェースの製品化が進んだことである。Sensable Devices 社は PHANToM を用いたアプリケーションを作成するためのソフトウェアツール GHOST を開発した[8]。また、研究用のものとしてはノースカロライナ大学の ARMLib がグラフィックスとハプティックインタフェースの融合問題を取り上げている[9]。

ＶＲソフトウェアの別の方向性として、実世界との融合という課題が今日注目を集めている。長い歴史を持つテレイグジスタンスの技術はその例であるが、近年舘らは遠隔ロボットの臨場感制御を行うためのソフトウェアツールＲＣＭＬの開発を進めている[10]。また、最近のトレンドとして複合現実感に関する研究例が増えており、ＣＧを実世界に畳重したり、実写映像をバーチャル世界に取り込む手法が開発されているが、ソフトウェアツールとしてはまとめる段階には至っていない。

3. VR用基本ソフトウェアの技術課題

これまでのバーチャルリアリティ研究における動向と著者自身が行った力覚ソフトウェアに関する経験を総括すると、ＶＲ用基本ソフトウェアの主要な技術課題を以下の３つに整理することが可能である。

(1) 世界記述の支援

バーチャルリアリティのアプリケーションを作るということは、バーチャル世界の内容を何らかの方法で記述するということを意味する。その方法は、大きく分けると実世界の計測データを再構成するものと、バーチャル世界をすべてソフトウェアで構成するものに分かれる。手法の整備が進んでいるのは後者の方で、視覚情報に関しては基本的にはＣＧの領域で研究されてきたモデリングと物理法則シミュレーションと同様である。ただし、それを実時間で行うことに大きなバリアーがあり、実時間化の手法が鋭意研究されている。力覚情報の呈示に関しては後述するように実時間性の要求が厳しいため、別の技術課題が存在する。

実世界を再構成する手法はセンシング技術と並行して進められているが、聴覚情報の呈示においてはこの手法が主流といえる。

いずれにしても上記のような世界記述を支援するためには、重要な機能をモジュール化して、再利用可能な形態に整理することが不可欠である。

(2)多様なインタフェースデバイスのサポート

冒頭にも述べたように、バーチャルリアリティでは多種多様なインタフェースデバイスが開発され、それらをいかに活用するかが重要である。バーチャル空間で人が相互作用を行う場合、行動に合わせて感覚情報の呈示を受けなければならない。このような感覚情報の生成は広義のレンダリングと呼ばれる。視覚、聴覚、力覚にそれぞれ別個のレンダリング技術が存在し、それに必要なバーチャル物体のモデルもどのような感覚情報を生成するかのよって要求される仕様が異なる。また、感覚ディスプレイのハードウェアを駆動するためのデバイスドライバも無論必要である。これらのレンダラー、物体モデル、デバイスドライバをモジュール化し汎用性を持たせることが必須である。

人間がインタラクションを行うためには、行動のセンシングが不可欠であり、ＶＲの研究領域でも各種の行動計測用センサーが開発されている。それらのセンサーにはそれぞれデバイスドライバが存在し、センサから取得されたデータから人間の行動を認識するためのエンジンが必要である。これらの機能もモジュール化しなければならない。

さらに、上記の各種入出力デバイスを統合するた

めには、それらの間で適切なデータ交換を行う機能モジュールが不可欠である。その実現方法が重要でありかつ困難な課題である。

(3)ネットワーク化

前述のように複数のユーザーによるＶＲ空間の共有は、今日における主要な研究テーマになっている。バーチャル世界におけるユーザーや物体のデータベースをどのように管理するか、時間遅れにどのように対処するか、といった問題に対して多くの研究者が取り組んできた。ただし、それらの研究のほとんどがジェスチャ入力を行いＣＧで表示するといった、８０年代終盤にＶＲが登場した当時のインタフェースデバイスを前提にしている。視覚以外の感覚情報を呈示する場合には、モデルとレンダラーが異なるため、まったく違う議論になる。特に、厳しい実時間性が要求される力覚呈示をどのようにネットワーク化していくかは、挑戦的な課題である。

4. 力覚と視覚を融合したVR環境構築ツール LHX

著者の研究室では，力覚フィードバックに対応したＶＲ環境構築ソフトウエアの開発に 1991 年より着手し、様々なアプリケーションを通じて熟成を重ねてきた。このソフトウェア群は初期のものはＶＥＣＳ、最近のものはＬＨＸと呼んでいる[11]。ＬＨＸは筆者の研究室で開発を行った各種の力覚ソフトウェアを整理して、再利用し易くするための概念的フレームワークである。研究の進展に伴って手法として確立してきたものをモジュール化することによって新たなアプリケーションの開発をしやすくするのが主たる目的である。現在のところ図１に示すような７つのモジュールにまとめている。以下に各モジュールの役割を説明する。

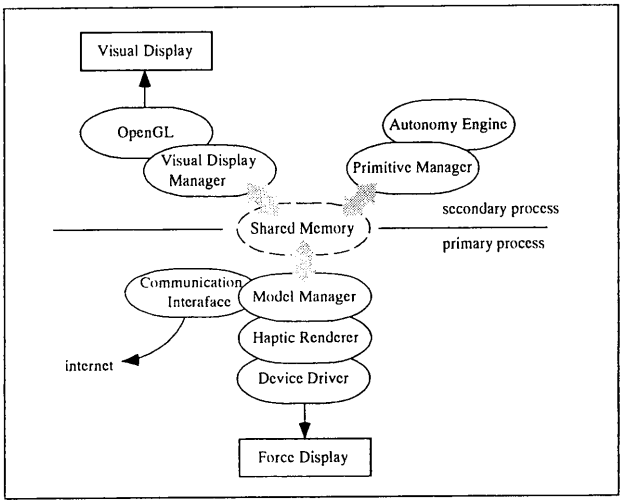

図１　ＬＨＸの基本構造
Figure 1. Basic structure of LHX

バーチャル物体を記述するためには、その３次元形状を与える他に、重さや硬さといった力覚特性を定義する必要がある。これを与えるのが力覚モデルである。この力覚モデルをもとに、人間が感覚器官を通じてバーチャル物体を知覚できるよう力覚情報をハプティックインタフェースの動作指令に変換する必要がある。これが、力覚レンダリングである。ＬＨＸにおける Model Manager と Haptic Renderer がこれに相当する。力覚レンダリングには大きく分けると、サーフェイスレンダリングとボリュームレンダリングがある。サーフェイスレンダリングは、バーチャル物体の表面の硬さ、表面粗さ等を表現するものである。例えば、硬さを表現するために、力覚モデルとして、バネモデルを与えることがよく使われる。パラメータとしてバネ定数を与えておくと、力覚レンダリング部では物体とハプティックインタフェースの効果器との接触判定をし、力覚モデルに基づきめり込み量に比例した反力を呈示する。このパラメータを調節することで、固い物体から柔らかい物体までを表現することができる。一方のボリュームレンダリングは、物体内部の密度や速度、応力分布等の属性を表現するものである。この場合、力とトルクをそれぞれの属性にマッピングする。著者の研究室ではこのボリュームレンダリングに力を入れており、Volume Haptics Library というソフトウェアの開発を行っている[12]。

力覚レンダラーの計算した呈示力の値をもとにハプティックインタフェースを駆動するのがデバイスドライバーである。ハプティックインタフェースのハードウェアは多様な形態をとるため、制御ソフトウェアはそれらのハードウェア構成の違いを吸収しなければならない。そのためには、それぞれのハードウェアに対応したデバイスドライバとレンダラーを用意する必要がある。そのような機能を整理した基本ソフトウェアを開発する試みが著者を含めて４大学共同で進められている[13]。このソフトウェアはＨＩＰとよばれ、市販のものや研究段階のものも含めて現在６機種をサポートしている。

最も基本的な力覚フィードバックの機能は以上の３つのモジュールで実現されるが、実際のアプリケーションはそれだけではすまない。まず、力覚情報は必ずといってよいほど視覚情報と一緒に提供される。視覚情報との連携をやろうとすると、更新速度の問題が出てくる。一般にＣＧの生成は計算機にとって重い処理であり時間を要する。視覚情報は１０Ｈｚ程度の更新速度でも何とか動画に見えるが、力覚情報は数１００Ｈｚから数ＫＨｚというオーダーになる。著者の経験では５０Ｈｚを切ると破綻が起きてくる。したがって、重い視覚ルーチンが走った時でも力覚フィードバックのアップデートが落ちない工夫が必要である。そのため、ＬＨＸで

は視覚情報を生成する Visual Display Manager を別のプロセスに分け、力覚系のモジュールが走るプロセスに優先権を与えることによりこの問題に対処している。Visual Display Manager にはＨＭＤ等の各種の視覚ディスプレイに映像を出力するコンポーネントが格納される。因みに、近年筆者の研究室で最も力を入れている視覚ディスプレイは没入型球面ディスプレイであり、歪み補正のプログラムがこのモジュールに格納されている[14]。

アプリケーションを作っていく上でプログラマーが最も努力を傾注するのが、ＶＲ世界の記述である。そこで、様々なアプリケーションに共通に用いられるバーチャル物体は再利用可能にする必要がある。ＬＨＸではこれまでに作られたバーチャル物体の中で重要なものをプリミティブとして登録し、移動、変形等の加工や力覚特性の変更が自由にできるようにしている。図 1 における Primitive Manager がそれである。プリミティブには球や立方体等の単純なものから、自由曲面のメッシュやボリュームデータといった複雑なものまで含まれている。また、ユーザーインタフェースに用いられる各種の力覚付き３Ｄアイコンもプリミティブとして登録されている。

ＶＲ空間に存在する物体は人間が関与しなくても変化が起きることがある。例えば、動物が動いたり植物が成長したり、といった現象を導入するアプリケーションもある。そのような場合、生物の発生や行動の仕組みを記述するソフトウェアが必要になる。著者は以前、デザイン作業に用いることを想定して、軟体動物や盆栽をＶＲ空間に作ったことがあるが[15]、それらは Autonomy Engine というモジュールに格納している。これは、プリミティブに自律的な挙動を与えるという位置付けである。

ＶＲの重要なアプリケーションに臨場感通信があるが、著者の研究室では複数のユーザが力覚を共有する研究を行ってきた。力覚の共有によって高度な技能を教示することが可能になる。そのようなアプリケーションにおいては遅延のない通信が不可欠である。最近ではＴＣＰ／ＩＰベースの通信回線の性能が急速に向上しているので、著者の研究室ではそれを用いた力覚通信ソフトウェアの開発を進めている。それが入るモジュールが Communication Interface である。

以上でＬＨＸのすべてのモジュールを一通り紹介したが、力覚を用いたアプリケーションは概ねこのフレームワークに納まるはずである。ＬＨＸの７つの機能モジュールは前章で述べた３つの技術課題に対して一通りの回答を与えている。まず、世界記述の支援を行うのが Primitive Manager と Autonomy Engine である。次に多様なインタフェースデバイスの連携を行うのが Model Manager と Haptic Renderer と Device Driver と Visual Display Manager である。そして、ネットワーク化に対応するのが Communication Interface である。

5. IOAの基本構造

本論文で提案するＩＯＡとは、Interaction-Oriented Architecture の頭文字をとったもので、多様なインタフェースデバイスの密な連携を実現するために必要なソフトウェアアーキテクチャを意味するものである。ＬＨＸはハプティックインタフェースのアプリケーション開発を主目的にしたものであったが、ＩＯＡはより広い意味で、人間の多様な行動を支援し拡張するための充実したインタフェース群の構築を目指すものである。

図２はＩＯＡの基本構造を示したものである。これは前述のＬＨＸをベースにしているものの、３章で述べた技術課題により本格的に対応すべく拡張が行われている。

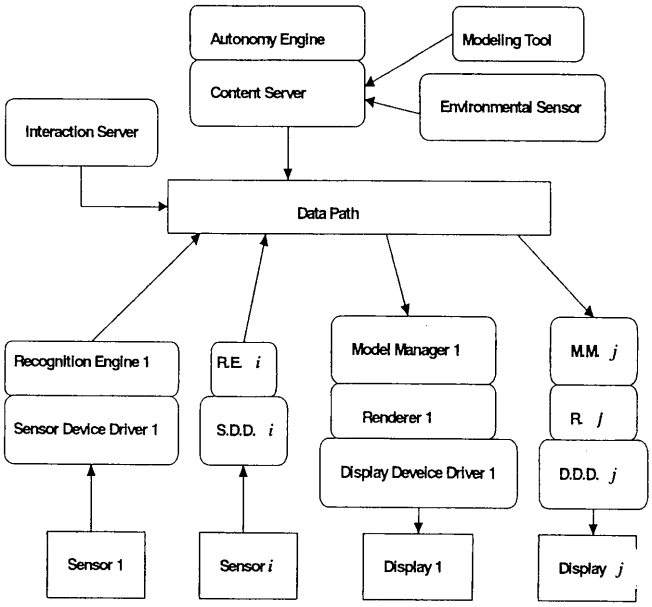

図２　ＩＯＡの基本構造
Figure 2. Basic structure of IOA

インタフェースデバイスのハードウェアは、ユーザーの行動を計測する Sensor i と、感覚情報を呈示する Display j に分けて一般的な記述をしている。これらのデバイス群は一人のユーザーが使うこともあれば、多数のユーザーが関与するという形態もありうる。行動計測用のセンサーにはデバイスドライバー（図中の Sensor Device Driver）に加えて、センサーデータから人間の行動を抽出する認識エンジン（図中の Recognition Engine)を備える。感覚情報を呈示するディスプレイにはデバイスドライバー（図中の Display Device Driver)に加えて、ＶＲ

世界のモデルを格納するModel Managerと物体からの応答を呈示物理量に変換するRendererを備える。

上記のインタフェースデバイスは通信を行うハードウェア（図中のData Path）で相互に接続される。Data Pathの実体はシステムの実装形態によって大きく異なり、コンピュータ内部の共有メモリやEthernet等の通信回線等がありうる。インタフェースデバイスが複数になるとそれらの間でデータ交換の管理を行う必要が起こる。それを司るのがInteraction Serverである。これには単なるデーター交換のルールが記述されるだけでなく、認識エンジンや世界モデルの上位的管理部が格納される必要もある。特に多人数のユーザーをサポートする場合はこの部分が重要である。

ＶＲ世界とのインタラクションは上記のモジュールで完結するが、高度なアプリケーションを構築するためには世界記述を支援するツールが必要である。それに対応するのが図中のContent Serverである。これはＬＨＸのPrimitive Managerを拡張したものと位置づけることができ、複雑な世界を記述するためのベースとなるモデルや素材をユーザーインタフェースと共に提供する。そして、モデルの作成を支援するModeling Toolと、実世界からの素材の獲得を支援するEnvironmental Sensorを備える。Environmental Sensorに相当する例としては、遠隔カメラヘッドから画像や距離情報を取得するものや、音をサンプルするものが挙げられる。Content Srever はそれらのモデルや素材のオーサリングツールも包含する。さらに、モデルに自律的な挙動を与える法則エンジン（図中のAutonomy Engine)が加わる。

6. IOAの試験的実装例

ＩＯＡは構想段階のものではあるが、最近著者の研究室で開発を行ったシステムで、この概念を説明する実例としてふさわしいものがあるので、それを２つ紹介する。

(1) Audio Haptics

ＶＲのアプリケーションにおいて音を生成する場合は、サンプルしたものを再構成するのが一般的であった。一方、視覚と力覚は定義されたモデルから生成するものが多い。実世界から取得されたものを再構成するのは忠実度という面では有利であるが、インタラクションを行う場合に大きな制約を受ける。バーチャル空間における音を呈示する場合、音場の再現という観点では多くの研究例があるが、バーチャル物体を手で叩いた時にどのような音がするか、という問題はほとんど取り扱われたことがない。Audio Haptics とは視覚と力覚に加えて、音もモデルから物理法則によって発生させることを行うものである[16]。

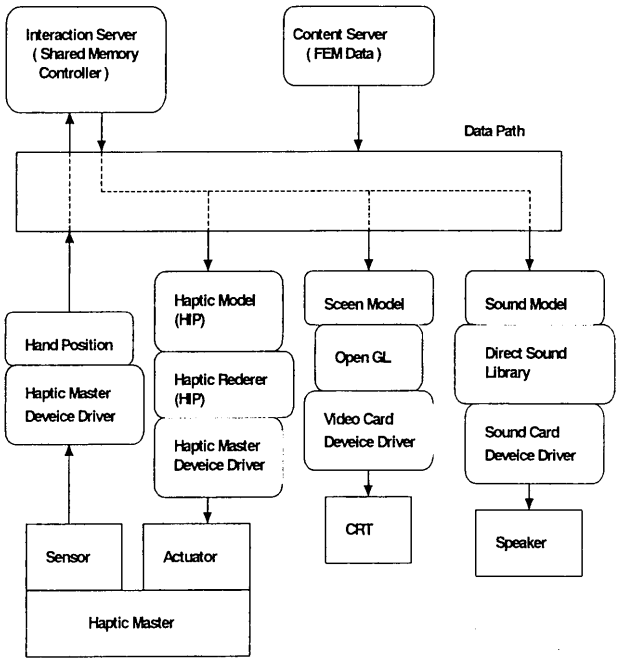

図３ Audio Haptics の構成
Fugure 3. Structure of Audio Haptics

図３はＩＯＡの枠組みにAudio Hapticsのシステムを当てはめたものである。力覚呈示には著者の研究室で開発された６自由度フォースディスプレイHapticMasterを用い、音はHapticMasterのグリップ部に取り付けたスピーカーから出す。視覚情報はＣＲＴに表示される。動作計測はHapticMasterの関節角度センサーを用いて行われる。この場合、認識エンジンは関節角度センサーから手の位置を計算する座標変換ルーチンである。力覚呈示系のModelManagerとRendererは前述のＨＩＰを用いており、視覚呈示系のそれはOpenGLである。これらのデバイスは単一のＰＣに接続されるため、各デバイス間のデーター交換は共有メモリを介して行われる。したがって、本システムにおいてはIntercation Serverには共有メモリ管理部が相当する。

聴覚情報を生成するために必要なモデルは、物体の振動と空気の振動を表すものがそれぞれ必要である。前者は有限要素法によって計算されるもので、これは一般に多大な時間を要するため、予め求めておきContent Serverに格納する。後者は速度ポテンシャルの計算によって求められる。この部分はリアルタイムで行われるため、バーチャル物体の任意の場所を叩いた時の音を聞くことができる。

以上のように、一つの物体モデルから視覚、聴覚、力覚という３つの感覚チャネルに同時に情報呈示

するシステムの構築がＩＯＡによって可能であることがわかる。

(2) Networked Haptics

ＶＲソフトウェアにおいてネットワーク化は重要課題の一つであることはすでに述べたが、力覚をフィードバックする環境をネットワークを介して共有すると体験や技能の伝達が可能になり、新しい応用が広がることが期待できる。著者は１９９０年代の初頭よりこの問題に取り組んできて様々な試みを行ってきており、それらを総称してNetworked Hapticsと名づけている[17]。

この技術は、更新レートや時間遅れ等が非常にシビアに影響する問題があり、従来の視聴覚情報のみの呈示を行うシステムとは異なるアーキテクチャが要求される。著者の研究室で行った最も新しい方式は、力覚呈示用世界モデルをサーバー・クライアントモデルを用いることによって実装し、多数のユーザーに力覚帰還環境を共有させるものである[18]。

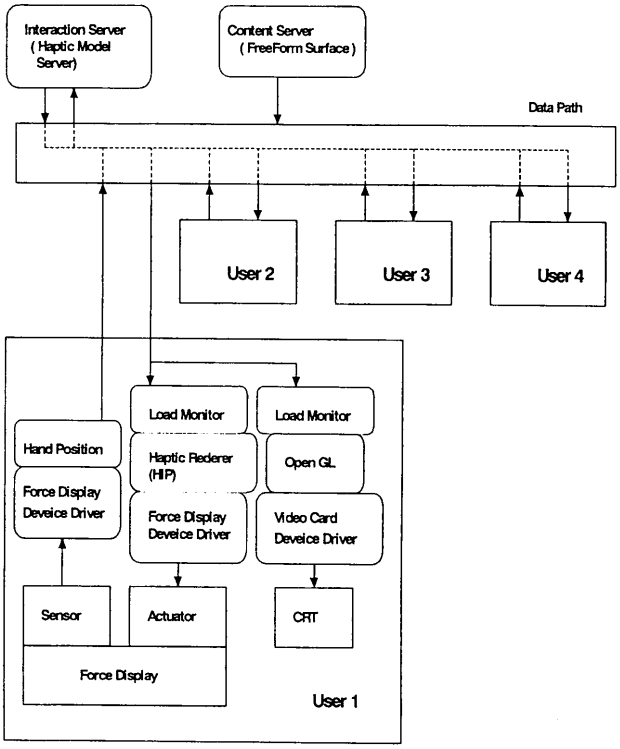

図４ Networked Haptics の構成
Figure 4. Structure of Networked Haptics

図４はこのシステムをＩＯＡの枠組みにあてはめたものである。力覚呈示には著者の研究室で開発された３自由度ペン型フォースディスプレイを用い、映像はＣＲＴに表示される。動作計測はHapticMaster と同様の方法で行われる。力覚のRenderer は前述のＨＩＰであり、視覚のそれはOpenGLである。この力覚・視覚呈示系は一つのＰＣに実装され、試作システムではそれを４ユーザー分をネットワークで接続している。４人のユーザーによって共有されるバーチャル世界のモデルはInteraction Server に置かれる。これは別のＰＣに実装される。各感覚呈示系ＰＣはクライアントとなり、サーバーからバーチャル世界の情報を受け取る。このシステムでは Data Path の実体は Ethernet (100Base-TX)である。

各クライアントの Model Manager はバーチャル物体のデータは持たないが、更新レートの監視を行う。視覚系プロセスと力覚系プロセスは異なるスレッドに分けられ、OpenGL に比べて軽量な HIP は高速でアップデートされる。これらの２つのスレッドは更新速度に極端な偏りが発生しないようにする必要がある。これは、いずれか一方のスレッドが遅くなるともう一方がますます速くなるという現象が起こるためである。その対策として、各クライアントの Model Manager は視覚系と力覚系のスレッドの更新速度に偏りが出ないように、負荷の調整を行う。具体的には、更新速度の差が一定限度を超えて広がった場合には、ダミーの負荷を加えて差が広がらないようにしている。

同様な機能は Interaction Server にも備えられる。各クライアントの更新速度の監視を行うスレッドがあり、クライアント間の更新速度に偏りが出ないように調整している。これはクライアントＰＣの性能が均一でない時に特に有功である。

近年ＴＣＰ／ＩＰベースの通信回線はギガビットの時代を迎えており、上記のようなＩＯＡに基づく共有ＶＲ空間は実用的なレベルに達しつつあるといえるだろう。

7. これからの課題

本論で述べたＩＯＡがソフトウェアアーキテクチャとして熟成していくためには、より多くの入出力デバイスを接続して問題点を洗い出す必要がある。著者の研究室では視覚と力覚の呈示装置に関しては数多くのデバイスを開発し、それらの主要なものは統合することに成功している。しかし、それら以外の感覚呈示装置に関して、ＩＯＡが有効かは確認していない。他の感覚呈示装置への対応はModelManager と Rederer の内容を適切に設計する必要がある。また、センサー系に関しても複数のものを用いた動作認識などに対してＩＯＡがどう対応するかは未知の課題である。複数センサーの同期や計測結果の統合を行う場合には、Interaction Manager に複数の認識エンジンを管理・統合する機能を加えることが必要があろう。

さらに、将来の課題としてＩＯＡに基づくソフトウェアの開発環境の整備が考えられる。現状ではツ

ールキットという形態での整備を進めているが、体系的な記述言語が必要になることが予想される。バーチャルリアリティの記述言語として現在普及しているものは、初歩的な世界記述をサポートしたＶＲＭＬがあるが、その機能は本来のバーチャルリアリティのアプリケーションを組むには程遠いものである。ＩＯＡのメリットを十分に活用できるような記述言語を開発することが今後の重要なテーマになるであろう。

8. まとめ

本論文では、多様なインタフェースデバイスを統合するソフトウェアアーキテクチャＩＯＡの設計思想について述べ、実例を交えてその内容を説明した。ＩＯＡの最大の特徴はシステムの全体構成においてインタフェースデバイスが中心に据えられていることである。人間を中心にしたシステムの重要性が唱えられて久しいが、実際に人がシステムに接する部分であるインタフェースデバイスについては少なくともＶＲ以外の領域では軽視されている傾向にある。冒頭にも述べたようにインタフェースデバイスこそが将来のメディア技術のキーであり、ソフトウェアはそれらの発展を支援しなければならない。

ＩＯＡの最大の特徴は新しいインタフェースデバイスができた時に、それを容易にシステムに統合できるアーキテクチャになっていることである。さらに重要なことは、従来コンピュータの中心技術であったＣＰＵやＯＳは、ＩＯＡにおいては付加物にすぎないことである。このアーキテクチャが本格的に実用化されるようになれば、情報産業自体が大きな構造的変化を起こすであろう。

文献

[1] Pope,A.: The SIMNET Network and Protocols, BBN Report No.7102 (1989)
[2] Zyda,M. et.al.:NPSNET:Constructing a 3D Virtual World, Computer Graphics,Special Issue on the 1992 Symposium on Interactive 3D Graphics (1992)
[3] Singhal,S. and Zyda,M.: Networked Virtual Environments - Design and Implementation, ACM Press Books (1999)
[4] http://www.sense8.com
[5] Cyberspace Developer Kit, Brochure of AUTODESK Inc. (1993)
[6] 木島、廣瀬：人工現実感の研究ー仮想空間エディタ Vis-Edit の開発、第７回ヒューマンインタフェース・シンポジウム論文集 (1991)
[7] Iwata,H. and Yano,H.: Artificial Life in Haptic Virtual Environment, Proceedings of ICAT'93 (1993)
[8] GHOST General Haptics Open Software Toolkit, Brochure of SensAble Technologies Inc. (1996)
[9] Randolf,M. et.al.: Adding Force Feedback to Graphic System: issues and Solutions, Proceedings of SIGGRAPH'97 (1997)
[10] 柳田、川上、舘：ＲＣＭＬ：アールキューブ操作言語の開発（第１報）、日本バーチャルリアリティ学会大会論文集 Vol.2 (1997)
[11] Iwata, H., Yano, H. and Hashimoto, W.:"LHX: An Integrated Software Tool For Haptic Interface", Computers & Graphics,Vol.21,No.4,pp.413-420(1997)
[12] Hashimoto,W. and Iwata,H.: A Versatile Software Platform for Visual/Haptic Environment, Proceedings of ICAT'97 (1997)
[13] 廣瀬他：触覚用共通ソフトウェア(HIP)の開発、日本バーチャルリアリティ学会論文誌、Vol.3,No.3 (1998)
[14] 橋本、岩田：凸面鏡を用いた球面没入型ディスプレイ：Ensphered Vision、日本バーチャルリアリティ学会論文誌、Vol.4, No.3 (1999)
[15] 矢野、岩田：自律的自由曲面を用いた仮想環境における協調作業、電気学会論文誌Ｃ編、１１５巻、２号 (1995)
[16] 矢野,岩田:"AudioHaptics: 物理法則に基づく聴覚と力覚の融合",ヒューマンインタフェース学会研究報告集 Vol.2 No.2, pp.19-24(2000)
[17] Iwata,H.: Challenges in Networked Haptics, Workshop Note of IEEE VR 2000 (2000)
[18] 木村、矢野、岩田：サーバーを用いた力覚帰還型協調仮想環境の開発、日本バーチャルリアリティ学会大会論文集、Vol.3 (1999)

（2000 年 8 月 7 日 受付）