

エネルギー科学研究科
エネルギー社会・環境科学専攻修士論文

題目： ネットワーク上の情報資源を活用した
分散型仮想環境システムの構築

指導教官： 吉川 榮和 教授

氏名： 大坂 融弘

提出年月日： 平成13年2月7日(水)

論文要旨

題目： ネットワーク上の情報資源を活用した分散型仮想環境システムの構築

吉川榮和研究室, 大坂 融弘

要旨：

分散型仮想環境 (Distributed Virtual Environment; DVE) 技術は、仮想現実感技術により計算機の中に仮想の世界を作り出し、その世界をネットワークで複数の人が遠隔地より共有できる利点を有しているため、コミュニケーション、電子商取引、教育訓練等の様々な分野への応用研究が行われている。

しかし現状では、実際に仮想世界の作成を行う際に、作成者の意図に沿った仮想世界を柔軟に作成する上で問題点があり、DVEの普及の足かせとなっている。この問題を解決し、仮想世界作成の自由度が向上すれば、分散型仮想環境技術の様々な分野への更なる貢献が期待できる。特にエネルギー分野では、エネルギー需給システムの根幹を支える様々なプラント運転員の訓練の効率化による機器の安全・信頼性の向上、エネルギー教育等への適用が期待される。

本研究では、ネットワーク上の情報資源、具体的にはホームページやシミュレータプログラム等を活用し、柔軟かつ容易に仮想世界を作成できる、新しい分散型仮想環境の構築手法を提案し、システムの構築を行った。

本論文では、まず分散型仮想環境とその従来研究について展望して、その問題点を述べ、より柔軟性と拡張性に富んだ分散型仮想環境システムの構築法を検討した。次いで、ネットワーク上の情報資源を柔軟に活用することのできる分散型仮想環境の構成を提案した。そして、提案するシステムを実現するための必要機能を抽出し、一方で、仮想世界を作成するユーザのタスクを検討した。そして、並列離散事象シミュレーションにおける同期プロトコルの方法を参考にして、多数の情報資源を本システムに取り入れて分散型仮想環境のシミュレーションを行う際のシミュレーション計算の時間的整合性を動的に管理する新しい手法を考案し、ユーザが仮想世界を作成する際のタスク削減を図った。

さらに、提案したシステムを構築し、実際に風力発電を対象としたエネルギー教育用仮想世界を作成することにより、ネットワーク上の情報資源を組み込むこと、また、作成者が整合性を設定することなく仮想世界のシミュレーションで情報資源間の整合性が管理されていることを確認し、作成者のタスクが削減されたことを確認した。また、システムのリアルタイム性などの諸指標を測定し、システムの総合パフォーマンスを評価した結果、目標を達したシステム開発が行えたことを確かめた。

目次

第 1 章 序論	1
第 2 章 研究の背景と目的	3
2.1 分散型仮想環境技術	3
2.1.1 分散型仮想環境技術の概要	3
2.1.2 分散型仮想環境技術の応用研究	4
2.1.3 仮想世界作成時の問題点	10
2.1.4 ホームページを参考にした問題点の解決法の検討	12
2.2 本研究の目的	13
第 3 章 ネットワーク情報資源活用型 DVE	14
3.1 昨年度のシステムについて	14
3.1.1 基盤システム LASNET	14
3.1.2 連携保守訓練システム NETCOM	15
3.1.3 コンテンツを作成するための問題点	17
3.2 ネットワーク情報資源活用型 DVE の提案	18
3.2.1 ネットワーク情報資源活用型 DVE の概要	18
3.2.2 ネットワーク情報資源活用型 DVE の処理方法	20
3.3 ネットワーク情報資源活用型 DVE の実現に向けた課題	22
3.3.1 処理の設定方法	22
3.3.2 解決すべき課題	23
3.4 因果関係を考慮した処理の手法	24
3.4.1 並列離散事象シミュレーションにおける因果関係の管理手法	25
3.4.2 仮想世界における因果関係の管理手法	26
3.4.3 仮想世界のリアルタイム処理について	35
3.5 処理の流れ	36
3.5.1 ロケールサーバ	36
3.5.2 外部モジュール	38

3.5.3	ブラウザ	38
第 4 章	基盤システム MALIONET の開発	41
4.1	システムの構成	41
4.1.1	ハードウェア構成	41
4.1.2	ソフトウェア構成	42
4.1.3	因果関係の管理を行うモジュールの詳細	46
4.1.4	使用する送信パケット	50
4.2	設定ファイルの仕様	51
4.3	仮想世界の作成と動作	51
第 5 章	動作確認実験	59
5.1	システムの使用例	59
5.1.1	確認項目	59
5.1.2	動作確認時のシステム構成	59
5.1.3	結果	62
5.1.4	考察	69
5.2	DVE システムの性能評価	69
5.2.1	測定項目	69
5.2.2	確認方法	69
5.2.3	動作環境	71
5.2.4	結果	71
5.3	まとめ	73
5.4	今後の展望	74
第 6 章	結論	75
	謝辞	77
	参考文献	78

目 次

2.1	分散型仮想環境の概念	4
2.2	PAW ² の画面例	5
2.3	InstantCommunityの画面例	6
2.4	ジャパネット"VIRTUAL CHAT"タウンの画面例	7
2.5	NETCOMの画面例	7
2.6	HyClassの画面例	8
2.7	3D-IESの画面例	9
2.8	エネルギー需給計画システムの画面例	9
2.9	VR、DVEシステムのシミュレーションの概略	11
3.1	LASNETのシステム構成	15
3.2	仮想世界の分割による負荷分散の概念	16
3.3	NETCOMのブラウザ	17
3.4	情報資源を活用した仮想世界の作成の概念	19
3.5	ネットワーク情報資源活用型DVEの処理例	21
3.6	PDESの構成	25
3.7	PDESを適用したDVE	27
3.8	外部モジュールとロケールサーバの接続例	28
3.9	保守的プロトコルと楽観的プロトコルによるデータの流れの比較	29
3.10	データの流れの例	34
3.11	ロケールサーバの処理の流れ	37
3.12	外部モジュールの処理の流れ	39
4.1	ハードウェア構成	42
4.2	ソフトウェア構成	44
4.3	ブラウザの画面例	47
4.4	外部モジュールの構成	49
4.5	情報リンク管理プロセスの構成	49

4.6	送信するパケット	50
4.7	ロケールファイル	52
4.8	オブジェクトファイル	52
4.9	情報リンクファイル	53
4.10	外部モジュールファイル	53
4.11	仮想世界作成の例	56
4.12	仮想世界への参加	58
5.1	ハードウェアとソフトウェアの動作環境	60
5.2	移動と衝突判定に関する外部モジュールの関係	61
5.3	風力発電のコンテンツに関する外部モジュールの関係	61
5.4	情報提示ウィンドウに関する外部モジュールの関係	62
5.5	サーバ用 PC の画面例	62
5.6	ブラウザ用 PC の画面例	63
5.7	作成した教育用仮想世界の画面例	63
5.8	風速・風向が掲載されている HP の例	64
5.9	CGI プログラムを利用して外部モジュールを作成する例	65
5.10	衝突処理の有無による仮想世界の相違	65
5.11	風力発電と灯台の動作例	67
5.12	情報提示ウィンドウの動作例	68
5.13	ロケールサーバの性能評価に使用する外部モジュールの構成	70
5.14	メッセージ数とロケールサーバの描画数の関係	73

表 目 次

2.1	DVE の分類	12
3.1	NETCOM の拡張機能	16
3.2	保守的プロトコルと楽観的プロトコルの比較	26
4.1	ブラウザ用 PC とその想定性能	43
5.1	移動と衝突判定の因果関係の確認	66
5.2	風力発電のコンテンツの因果関係の確認	66
5.3	情報提示ウィンドウに関する因果関係の確認	67
5.4	ロケールサーバの性能評価に使用する外部モジュール一覧表	70
5.5	パフォーマンス測定に使用する PC の性能	71
5.6	ロケールサーバのパフォーマンス測定結果	72

第 1 章 序論

近年の IT 技術の発達により、新たに生まれた技術に分散型仮想環境技術 (Distributed Virtual Environment; DVE) がある。この技術は VR 技術とネットワーク技術を統合するもので、VR 技術により計算機が仮想の世界を作り出し、ネットワーク技術により遠隔地にいる複数の利用者と仮想世界を共有できる。DVE では、音声と映像等のマルチモーダルコミュニケーションを可能とし、しかもリアルタイムで仮想の世界を共有できる。

この技術が実用化すれば、エネルギー分野では効果的な教育と訓練やエネルギー情報の社会啓蒙に新しい手段を提供できると考えられる。また、アミューズメント分野やビジネス分野など、様々な分野への貢献が期待される。

しかし、実際に仮想世界を作成する際に、現状レベルの汎用的な DVE システムでは、作成したい仮想世界のコンテンツにあわせた処理を容易に組み込めるかわりに、簡単な処理しかできず、またコンテンツに特化した複雑な処理を用意している DVE システムでは、あらかじめシステムに組み込まれている処理しかできないという問題点があり、仮想世界の作成者の意図に沿った環境を容易に作成することは難しい。そのため、仮想世界作成の自由度が高く、作成が容易な仮想環境システムが望まれている。

現在、IT 技術の発展でホームページが社会一般に普及している。ホームページの作成者はそのコンテンツ作りのための素材集など、インターネット上に既にある情報を利用して、自らのホームページの内容を充実させている。分散型仮想環境でもホームページ作成と同様にネットワーク上に大量に存在している情報を利用すれば、容易にかつ柔軟に仮想世界を作成できると考えられる。

本研究では、ネットワーク上の情報を利用して作成者が容易に仮想世界を作成し、仮想世界作成の自由度を向上させるための新しい分散型仮想環境システムの構築を目的とする。

本論文では第 2 章で研究の背景と目的について述べ、次いで仮想世界の作成時の問題点について述べる。第 3 章では、その解決法として、ネットワーク上の情報を活用できる分散型仮想環境システムを提案する。また提案したシステムを実現する際の課題を検討し、その解決手法を提案する。そして第 4 章では提案した手法に基づき昨年度に本研究室で開発した DVE システムを拡張して、ネットワーク上の情報を活用でき

るようにした分散型仮想環境システムの構築を行う。第5章では、構築したシステムを用いて実際に仮想世界を作成し、システムの評価を行う。最後に第6章で本論文のまとめとして、構築したシステムについての考察を行い、今後の展望について述べる。

第 2 章 研究の背景と目的

本章では、まず分散型仮想環境技術の概要を述べ、その応用分野を展望する。次に、分散型仮想環境技術をさらに高度化させる観点から、従来研究での問題点と本研究の目的を述べる。

2.1 分散型仮想環境技術

分散型仮想環境 (Distributed Virtual Environment; DVE) 技術は、複数の利用者がネットワークを利用してコンピュータ上に作り出した仮想空間を共有することによりマルチユーザ・インタラクションを可能にするものである^[1]。この技術は、現在既に様々な分野で応用研究が開始されている。

本節では、まず DVE の概要について説明した後、従来研究について紹介する。最後に、従来研究の問題点について述べる。

2.1.1 分散型仮想環境技術の概要

近年の計算機技術の発達は、コンピュータグラフィックス (Computer Graphics; CG) 技術で生成される 3 次元映像空間の中に人が没入して、現実世界と同様のインタラクションをとることを可能とした。この技術は人工現実感 (Virtual Reality; VR) 技術^[2]と呼ばれ、計算機と人間との新しい関わりをもたらすインタフェースとして期待されてきた。この技術を応用すれば、現実には存在しないものがあたかも存在しているかのように計算機の中に仮想の世界を作り出し体験できるため、設計、科学技術計算、教育、医療、娯楽等の広い分野に活用されている^{[3][4][5][6][7]}。

さらにネットワークの広帯域化により、VR 技術とネットワーク技術を融合する分散型仮想環境技術が研究されるようになった^[8]。この DVE 技術は、図 2.1 に示すように VR 技術を用いることで計算機の中に作り出した仮想世界を、ネットワーク技術を用いて離れた場所にいる複数の利用者が仮想空間を共有し、共に仮想世界を体験できる技術である。DVE の利点として、以下のようなことが挙げられる。

1. ネットワークを利用し利用者間の地理的な制限を排除できる。

2. 双方向かつ実時間でのコミュニケーションができる。
3. コンピュータシミュレーションと統合し、非現実的な環境や危険を伴う訓練を安全に繰り返し体験することができる。

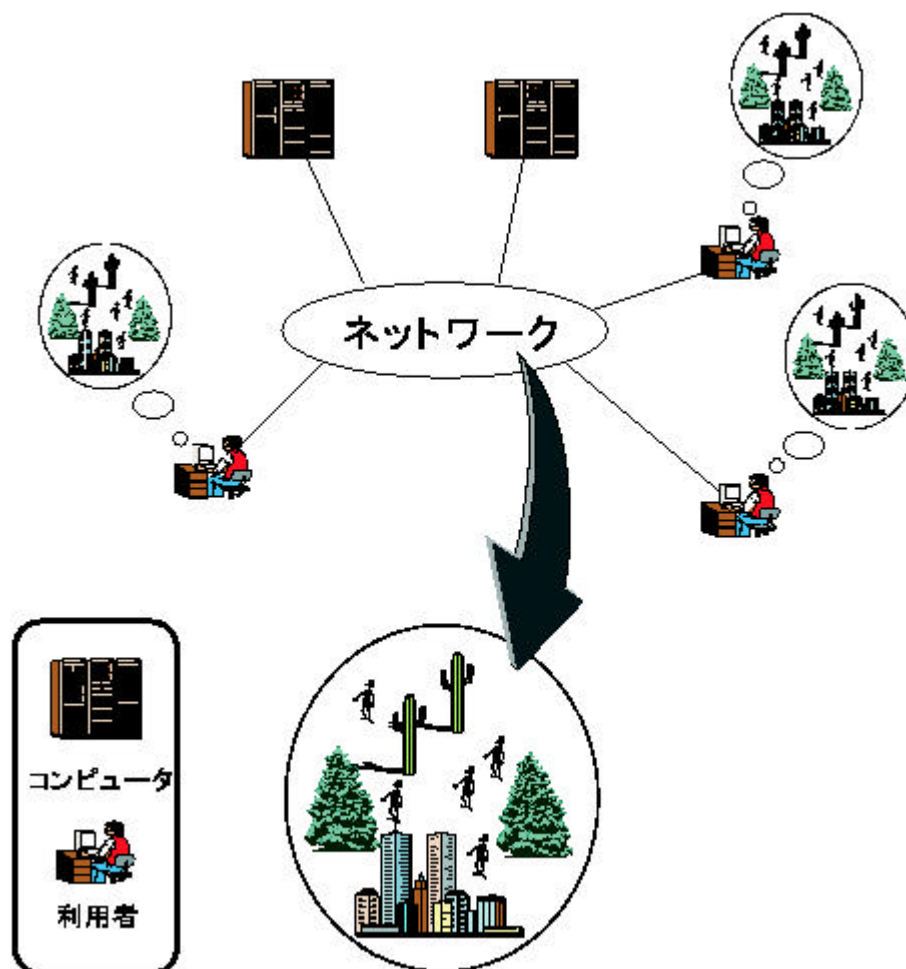


図 2.1: 分散型仮想環境の概念

2.1.2 分散型仮想環境技術の応用研究

2.1.1 項で述べたように、DVE 技術は様々な分野にその応用が期待される技術である。本節では、その応用分野別にいくつかの DVE の実例を紹介する。

コミュニケーションツール 1970 年代から 1980 年代にかけて、米国では MUD(Multi User Dungeons) という新しいタイプのゲームが流行した。MUD は、ネットワークで接

続されたコンピュータを利用して、複数の利用者がテキストでチャットを行いながら、洞穴の中に潜む怪物に挑み冒険を行うロールプレイングゲームの一種であった。計算機技術の飛躍的発展とともに、このような MUD が 3 次元仮想環境とともに提供されるようになったものが、コミュニケーションツールとしての DVE である。

SONY の PAW² (パウパウ)^[9] (図 2.2)、blaxxun interactive, Inc. の Instant Community^[10] (図 2.3) 等は、現在インターネット上でサービスが展開されている DVE システムの代表例である。例えば PAW² では、さまざまな出会いやイベントに参加できる他、かわいいペットの育成を楽しむこともできる。図 2.2 は、その画面例で、左上の画面で 3D 空間が表示されると共に、チャット (右上)、地図 (右下)、ペット情報の表示 (左下) の各ウィンドウがある。



図 2.2: PAW² の画面例

オンラインショッピング IT 技術の進展とともに、ネットワーク化されたコンピュータを用いて商取引を行う電子商取引技術が大いに利用されている。DVE をオンラインショッピングに利用する試みは従来の電子商取引技術と異なり、ウィンドウショッピングを行える、商品陳列の仕方ですれ行きを左右させることができる等、3次元空間の特



図 2.3: InstantCommunity の画面例

色を活かしたショッピングを可能にした。このようなシステムの代表例として、ジャパネットたかたのジャパネット"VIRTUAL CHAT"タウン^[11]を挙げることができる(図 2.4)。このシステムでは他の客とチャットを楽しみながら、ウィンドウショッピングをすることができる。

訓練 医療、軍事、宇宙、原子力等、実機を用いて訓練するには高価あるいは危険な環境を仮想環境として作り出し、その中で複数の利用者が訓練を行うシステムも、DVEの代表的な応用分野である。例えば、図 2.5 に示す昨年度本研究室で開発した NETCOM は、仮想プラントを対象とした訓練システムである。利用者は、自分の分身であるアバタを操作することで仮想プラント内を移動しポンプの保守を行う。また、チャットや音声対話機能を用いることにより、作業員同士が連絡を取り合って設備の保守を行う「連携作業」の訓練を行うことができる^{[12][13]}。



図 2.4: ジャパネット "VIRTUAL CHAT" タウンの画面例

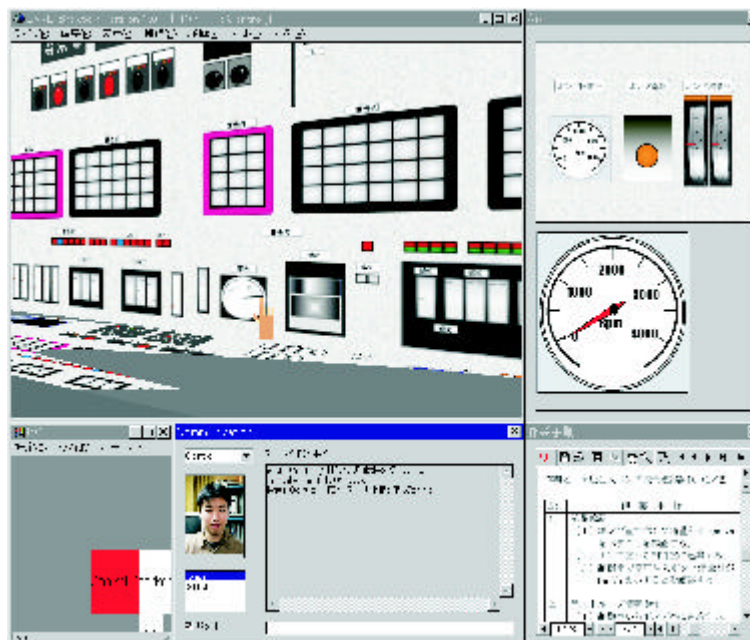


図 2.5: NETCOM の画面例

教育 DVE では計算機で処理した結果を 3 次元空間中に可視化できる点に着目し、教育分野への応用も盛んに研究開発が進められている。例えば、NTT の HyClass^[14] (図 2.6)、野村総合研究所の 3D-IES^[15] (図 2.7) 等が代表的な例である。HyClass や 3D-IES では、双方向のコミュニケーション学習環境を備えたマルチメディア教育システムを提供している。

エネルギー教育の分野においても、DVE 技術を利用することの有効性は指摘されている。例えば長松らは、エネルギー需給計画を題材に、ディスカッションやグループ学習を行うことができるシステムを開発している^[16] (図 2.8)。このシステムでは、利用者が線形計画法を用いる GAMS プログラムを用いてエネルギー需給計画を立案でき、その計画を実行したときのエネルギー需給予測結果が 3 次元で表示され、その結果を閲覧しながら皆でディスカッションすることもできる。

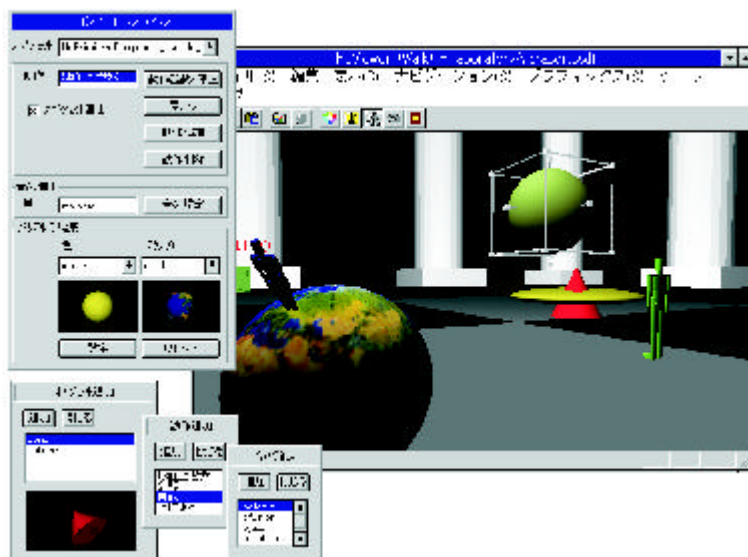


図 2.6: HyClass の画面例

以上のように、DVE は様々な分野での応用が期待され、活発に開発が行われている。



図 2.7: 3D-IES の画面例

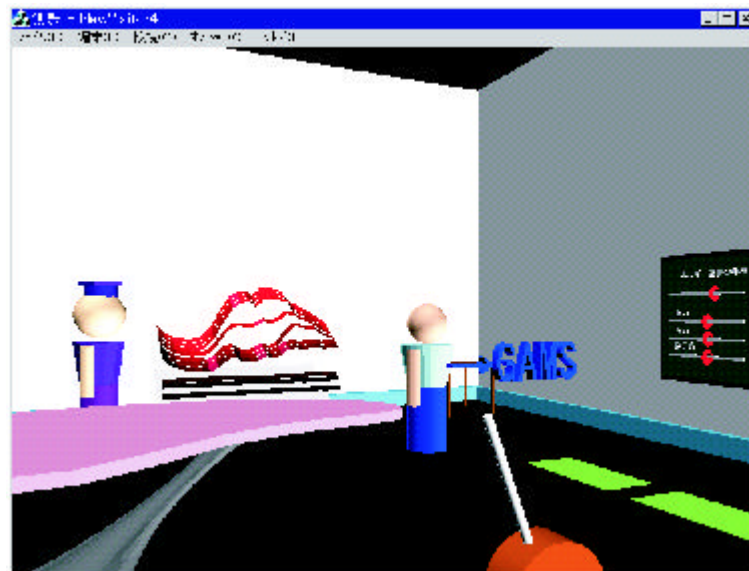


図 2.8: エネルギー需給計画システムの画面例

2.1.3 仮想世界作成時の問題点

VRのシミュレーションは図2.9に示すように、次の様な3つのサブシステムに大別できる^[2]。それらは、(1). 利用者の動作を計測し、その動作を計算機で扱うことができるデータ形式に変換する計測サブシステム、(2). 利用者の動作を仮想世界に反映し、仮想空間内に表示するデータを生成するシミュレーションサブシステム、および(3). シミュレーションサブシステムによって生成されたデータを、音や図表、グラフィックスで利用者に提示するために、提示できるデータ形式に変換し提示する表示サブシステムである。利用者の動作はジョイスティックやデータグローブなどの計測機器で計測し、仮想空間内で処理された結果は、ディスプレイやスピーカなどの提示機器によって利用者に提示される。DVEシステムとは、(2)のシミュレーションサブシステムで生成されるデータを、ネットワークを介して他の利用者と共有できるように拡張したものである。前述したようにVRのシミュレーションでは、(1)、(2)、(3)のパスを経てデータの入出力を行うので、必然的に時間遅延(タイムラグ)が発生する。従って、VRのシミュレーションでは、このタイムラグを人間に認知できない短い時間で処理しなければ、現実世界と同様の世界を体験しているように感じられない。特にDVEでは、ネットワークを介して情報の共有を行っているため、VRよりもタイムラグが大きくなる傾向があり、リアルタイム性の維持は非常に重要視される^[17]。また、利用者が仮想世界で疑似体験を行う際、アバタが物体の中に入り込む等の現実とかけ離れた出来事は利用者に違和感を与えてしまうため、現実性も重要視される。

また、仮想世界の利用者はブラウザと呼ばれるソフトウェアを使用し、仮想世界を体験できる。ブラウザとは、仮想世界を体験するために必要な機能を利用者に提供しており、利用者の化身であるアバタの操作機能、仮想世界の3次元画像提示機能等のほか、仮想環境へ接続機能を備える。

上述(2)でのシミュレーションサブシステムで行う処理とは、仮想世界の中の物体に動きや仮想世界の変化に対する反応を行わずするために必要な処理であり、本研究では、これらには大別して2種類あると考えた。1つはコンテンツに関わらず全ての仮想世界で汎用的に用いられる処理、もう1つは、コンテンツに特化した処理である。ここでいうコンテンツとは、作成者が利用者に対し提供する仮想世界の内容であり、昨年度のシステムNETCOMを例にすると、NETCOMではプラント内の連携作業というコンテンツを提供している。この2種類の処理の区別はシステムの設計方針によって異なるが、例えばNETCOMでは、利用者の分身であるアバタの移動や物体の衝突などの処理は前者の汎用的な処理であり、一方プラント内のシミュレーション、具体的には

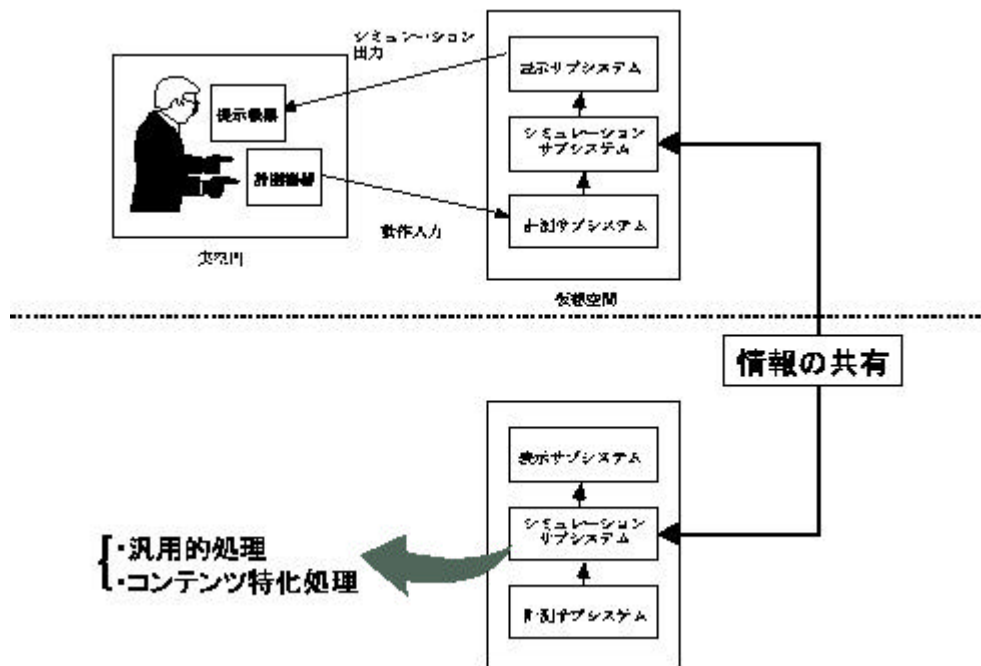


図 2.9: VR、DVE システムのシミュレーションの概略

ポンプの制御や発電量の計算等の処理は後者のコンテンツの処理にあたる。基盤システムでは汎用的な処理しか行っておらずコンテンツの処理を行うことにより作成者の意図する仮想世界を体験者に提供できるため、コンテンツの処理が重要となる。

2.1.2 項で紹介した DVE システムを、このような観点から整理し、その結果を表 2.1 に示す。表 2.1 に示すように、PAW ^2、ジャパネット "Virtual Chat" タウン、および 3D-IES は共通の基盤システム CommunityPlace を用いて開発されている。すなわち、汎用的な処理、例えば利用者同士が仮想都市をウォークスルーし、チャットを楽しむ等の処理は、CommunityPlace を用いている。一方、PAW ^2 におけるペットとの会話、ジャパネット "Virtual Chat" タウンにおけるオンラインショッピング、3D-IES における遠隔教育のような、コンテンツに特化した処理は、システム開発者がコンテンツを追加するためにプログラミング開発を行い、専用のソフトウェアを CommunityPlace に組み込むことによって提供されている。

このようなシステムを用いて新たなコンテンツの仮想世界を作成する際、CommunityPlace に代表される汎用的システムでは、仮想世界の作成者が作成するコンテンツに合わせて処理を追加することができるが、それは既にシステムに組み込まれている汎用的なものに限られる。

一方、PAW^2 では CommunityPlace をベースにペットの動作等の処理を追加し、ペッ

表 2.1: DVE の分類

システム名	基盤システム	コンテンツの処理
PAW^2 (パウパウ)	CommunityPlace ^[18]	ペットとの会話
Instant Community	独自のシステム	コミュニケーション
ジャパネット "Virtual Chat" タウン	CommunityPlace	オンラインショッピング
NETCOM	LASNET ^[19]	連携訓練
HyClass	InterSpace ^[20]	協調的学習
3D-IES	CommunityPlace	遠隔教育
エネルギー需給計画システム	LASNET	エネルギー教育

トというコンテンツを可能としている。しかし、CommunityPlace は外部のシステムを組み込むことを考慮していないため、PAW^2 のようにコンテンツに特化した新しい処理を追加するには、プログラミングによりシステムを新しく構築する必要があり、CommunityPlace に関する知識と大きな労力を必要とする。

このように、基盤システムでは作成者は思い通りの仮想世界を作成できず、一方、コンテンツに特化したシステムでは新しいコンテンツに合わせた処理を追加する際の労力は非常に大きなものになり、いずれの場合も実際に作成者が仮想世界を作成する際の障害となり、DVE を実際に利用する際の足かせとなっている。

2.1.4 ホームページを参考にした問題点の解決法の検討

近年、DVE の研究が進展し実用化に近づいている背景には、計算機技術とネットワーク技術の発展がある。これらの技術の発展は、DVE だけでなく私たちの生活にまで影響を与えている。例えば、従来は手紙のやりとり等で行ってきたメッセージ交換は電子メールを送受信することにより瞬時に行えるようになり、さらにインターネット上のホームページからレシピをダウンロードし調理できる電子レンジ^[21]まで販売されている。

また、DVE の応用研究も行われるようになり、すでに実用段階に入っているものもある。しかし、前項 2.1.3 で述べたように、実際に仮想世界を作成するには大きな労力

を必要とし、広く普及する際の障害となっている。

一方で、近年のホームページの爆発的な普及には目を見張るものがある。これほどの普及をみせた要因として、Web 閲覧者自身も容易にホームページ作成者になれることが挙げられる^[22]。ホームページの作成者は比較的理解しやすいHTML(Hyper Text Markup Language)言語によってホームページを作成することができる。そして、インターネット上にあるホームページ素材集などを利用したり、他のホームページにリンクを貼ることにより、自らのホームページの内容を簡単に充実させることができる。

DVEでもこのようにネットワーク上の情報を簡単に活用することができれば、ホームページ作りと同様に、簡単にコンテンツを充実させることができると考える。

2.2 本研究の目的

本研究では、仮想世界の作成者が容易に仮想世界を作成でき、かつコンテンツに特化した処理を自由に追加できるDVEへの発展を目指している。そして、その目標を達成するために次のステップで研究を行う。

1. ネットワーク上に溢れている様々な情報をコンテンツとして活用することにより、仮想世界作成の自由度の高いDVEを実現する「ネットワーク情報資源活用型DVE」を提案する。
2. 提案した手法に基づき、昨年度の基盤システムLASNETを拡張し、新しいDVEシステムMALIONETを構築する。
3. MALIONETを用いて、コンテンツに特化した処理を行う仮想世界を作成する実例として、実際にネットワーク上の情報を活用してエネルギー教育をコンテンツとする仮想世界を作成する。そして、動作確認によりMALIONETの機能を確認し、また性能面からも実験、評価を行う。

第 3 章 ネットワーク情報資源活用型 DVE

本章ではホームページ等のネットワーク上の情報資源を、コンテンツとして容易に組み込むことができる “ネットワーク情報資源活用型 DVE システム ”を提案する。

そこでまず、昨年度本研究室で開発した基盤システム LASNET をもとに提案する DVE システムを構築するため、LASNET と、それをベースにしてプラント内の連携作業に特化した訓練システムとして開発した NETCOM を例に、DVE システムのコンテンツ作りをどのようにすれば仮想世界作成の自由度が高められるか、またそのために必要な機能は何かを分析する。この分析結果をもとにして今回新たに考案した、ネットワーク情報資源活用型 DVE システムの概要を次に述べる。そして、このようなシステムを構築するための課題を検討した後、その課題を解決するために考案した手法について詳細に述べる。

3.1 昨年度のシステムについて

本研究室では昨年度、連携作業訓練システムを構築するために、まず DVE の基盤システム LASNET(Large Scale Networked virtual Environment) を構築し、次に LASNET をベースに連携作業というコンテンツを提供する訓練システム NETCOM(NETworked DVE based Training system for CoOperative machine Maintenance) を構築した。本節では、まず LASNET と NETCOM の概要について述べた後、仮想世界のコンテンツを作成する方法について整理する。

3.1.1 基盤システム LASNET

一般に、仮想環境の規模が大きくなるにつれ、仮想環境を共有するための通信負荷や、シミュレーションの計算負荷が増加する^[23]。昨年度の研究では、原子力発電プラントに代表される大規模工学プラントの保守作業の訓練支援を目標に、まず大規模仮想環境をシミュレートする基盤システム LASNET を構築した。図 3.1 に LASNET のシステム構成を示す。

LASNET では、仮想環境の大規模化に伴い計算負荷が増大するため、仮想世界を分割し複数のサーバで計算し負荷分散を行っている^[24](図 3.2 参照)。この分割された個々

の仮想空間をロケール、ロケールを管理するサーバをロケールサーバ(図 3.1 中 3)と呼ぶ。また、全てのロケールサーバを統括し仮想世界全体を管理するサーバをワールドサーバ(図 3.1 中 2)、仮想世界全体の情報を格納しているサーバをワールド情報データベース(図 3.1 中 1)と呼ぶ。ユーザはブラウザ(図 3.1 中 4)を利用して仮想世界を体験することができる。各サーバ間は通信方式として、通信路の数が少なく、データの集約が容易なクライアント・サーバ方式を採用している。また、Splineサーバ(図 3.1 中 5)は仮想世界のリアルタイム描画を行うもので、三菱電機(株)情報技術総合研究所が開発中のソフトウェア Spline^[25]を利用して開発した。

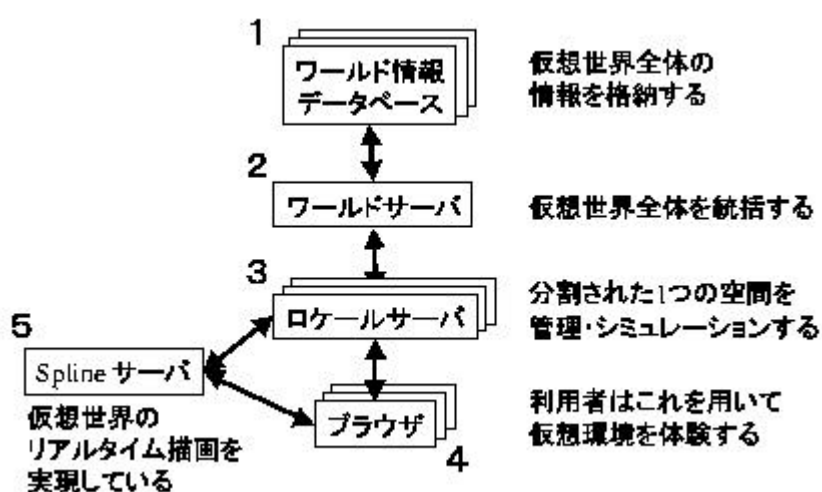


図 3.1: LASNET のシステム構成

仮想世界は各ロケールサーバで独立にシミュレーションされ、利用者はブラウザを用いてロケールサーバに接続することにより、仮想世界を体験することができる。

また、作成者は XML(eXtensible Markup Language)に基づいた記述言語を用いて仮想世界を記述する。XML はホームページを作成する際に用いられる HTML と類似した言語で、これを用いてホームページ作成と同様、DVE による仮想世界を容易に作成できる。

3.1.2 連携保守訓練システム NETCOM

NETCOM では、プラント計測制御システムの連携保守作業というコンテンツを提供するために、LASNET システムを拡張し、表 3.1 に示す 2 つのモジュールを実装した。

プラントシミュレータは、作成した仮想プラント全体の挙動を処理するためのモジュー

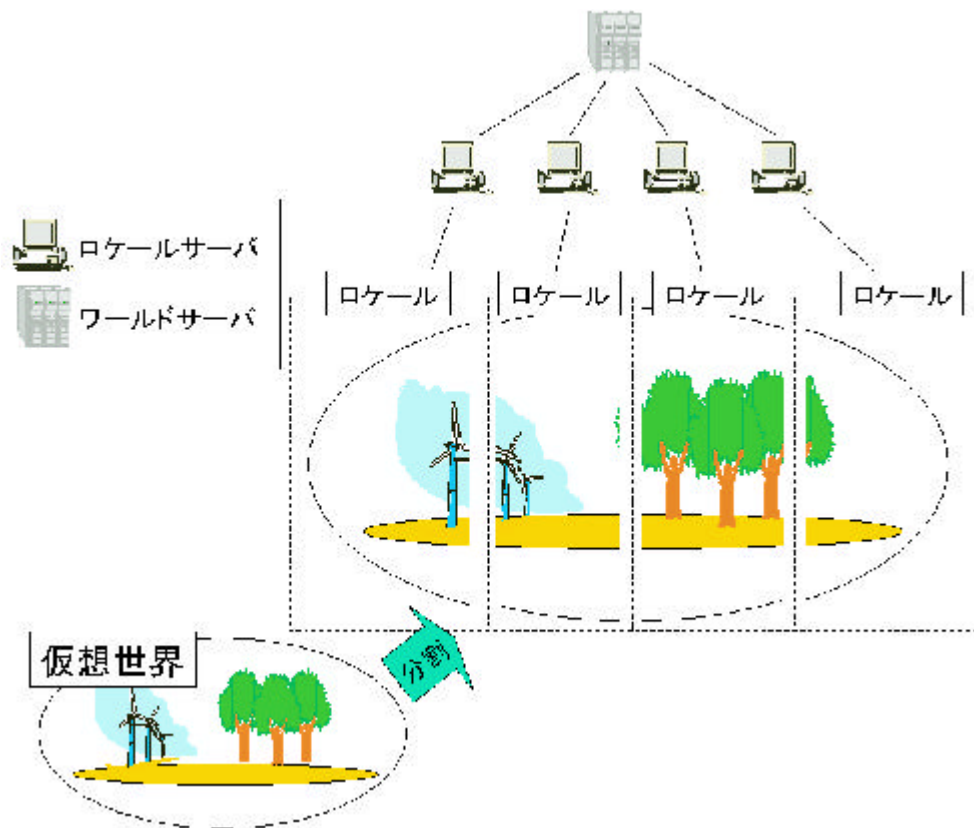


図 3.2: 仮想世界の分割による負荷分散の概念

表 3.1: NETCOM の拡張機能

モジュール名	機能	実装場所
プラントシミュレータ	プラントの挙動の処理	ワールド情報データベース
ウィンドウ	仮想のプラント内の機器の操作	ブラウザ

ルである。LASNETでは、移動する、衝突する等の汎用的な仮想世界の処理を行っているのに対し、NETCOMではこのようなシミュレータをプログラムとして組み込み、保守作業の訓練の場を提供している。

ブラウザは、PC上で使用され連携保守作業時の利用者と仮想世界とのインタラクションを実現する。NETCOMでは、ブラウザに図3.3に示す5つの連携作業訓練用ウィンドウを実装した。各ウィンドウは以下の機能を備える。なお(1)のウィンドウのみ、LASNETの機能として提供される。

- (1) 仮想世界提示ウィンドウ：仮想世界を提示する機能
- (2) 地理情報提示ウィンドウ：仮想世界全体を把握するための地図を提示する機能
- (3) 操作支援ウィンドウ：対象機器の操作を容易に行えるようにするための機能
- (4) コミュニケーションウィンドウ：仮想プラントの他の部屋にいる作業員と対話する機能
- (5) 作業手順提示ウィンドウ：訓練中に作業手順を提示する機能

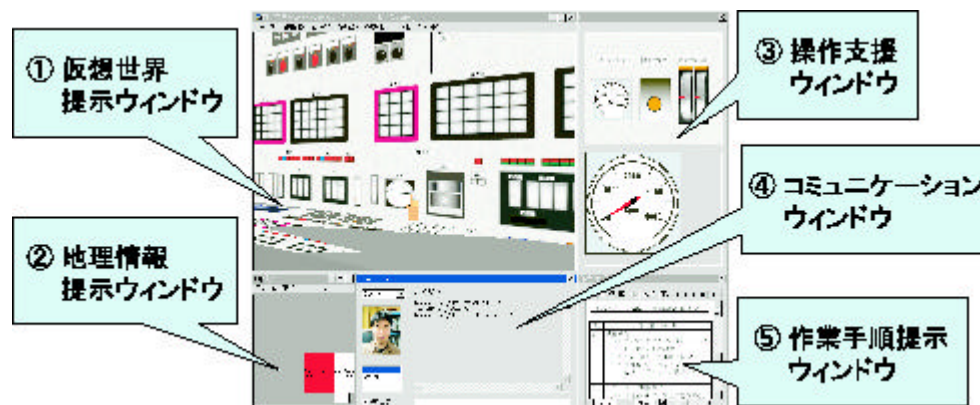


図 3.3: NETCOMのブラウザ

3.1.3 コンテンツを作成するための問題点

3.1.2項で述べたように、昨年度開発したNETCOMは、連携保守作業というコンテンツを提供するために基盤システムLASNETを拡張したシステムである。具体的には、プラントの状況に応じてプラント機器の挙動を処理するために、ワールド情報データ

ベースに新たにプラントシミュレータを組み込んだ。一方、保守作業を円滑に行うために新たなウィンドウをブラウザに組み込んだ。

これは、2.1.3項で述べた CommunityPlace をベースにした PAW² 等の従来型の DVE システムと同様で、新しいコンテンツの仮想世界を作成するには LASNET をベースにプログラミング作業を行って新たにシステムを構築する必要があり、仮想世界の作成者が容易に新たなコンテンツを追加することが困難である。

このような問題点を解決するために、昨年度の基盤システム LASNET をもとに新しい DVE システムの提案を行う。

3.2 ネットワーク情報資源活用型 DVE の提案

2.1.3 項で述べたように、仮想世界を作成するにはコンテンツに特化した処理が重要である。しかし、従来型の DVE システムを用いて自由にそして容易に仮想世界を作成する際に、コンテンツに特化した処理を組み込む際の労力が障害となっている。本研究では、ネットワーク上に存在する情報資源を活用することにより自ら作成することなくコンテンツを追加できると考え、ネットワーク上の情報資源を活用した DVE システムを提案する。これは、従来型の DVE システムがネットワークを介して DVE システム内でのみ情報の共有を行っているのに対し、提案する DVE システムでは、ネットワーク上にある他の情報資源とも情報の共有を行う。

本節では、まず提案する DVE の概要について述べた後、その利点について述べる。次に本研究で提案するシステムに昨年度のシステムを拡張する際の課題について検討し、最後にその解決法について述べる。

3.2.1 ネットワーク情報資源活用型 DVE の概要

近年のインターネット利用の拡大に伴い、インターネット上で発信されるデータ量は急増している。例えば 1999 年 8 月の時点で、国内における Web サーバ総数は 8.5 万台（対前年比 1.57 倍）、Web ページ総数は 3857 万ページ（同 2.15 倍）、Web でアクセスできる総ファイル数は 8570 万個（同 2.35 倍）と推計されている。さらにインターネット上では Web だけではなく、種々のプロトコルで実に多種多様な情報が提供されている^[22]。

本研究は、DVE システムにおいてコンテンツを作成する場合、つまりコンテンツに特化した処理を追加する場合、ネットワーク上の情報資源を活用することにより、仮

仮想世界作成の自由度が高いネットワーク情報資源活用型 DVE というフレームワークを提案する。

ネットワーク情報資源活用型 DVE は、システムに既に組み込まれた計算モジュールではなく、ネットワーク上の情報資源をそのまま利用してコンテンツに特化した処理を行うフレームワークである。情報資源とは、具体的にはホームページ等に掲載されている情報や、プラントシミュレータ等のシミュレータを指す。以下では、DVE に組み込めるようにしたこのような情報資源を外部モジュールと呼ぶ。また情報資源のもつ情報を DVE で活用するために情報をやりとりする通信路を、情報リンクと呼ぶ。

提案するフレームワークにおける、情報資源を活用して仮想世界を作成するという概念を図 3.4 に示す。ネットワーク情報資源活用型 DVE では、ネットワーク上の情報資源を外部モジュールとして利用し、情報リンクを介して DVE に接続して、コンテンツに特化した処理を行うことができる。つまり外部モジュールを柔軟に組み合わせる情報リンクを作成することにより、仮想世界作成の自由度を大幅に拡大し多様な仮想世界を作成することが可能となる。

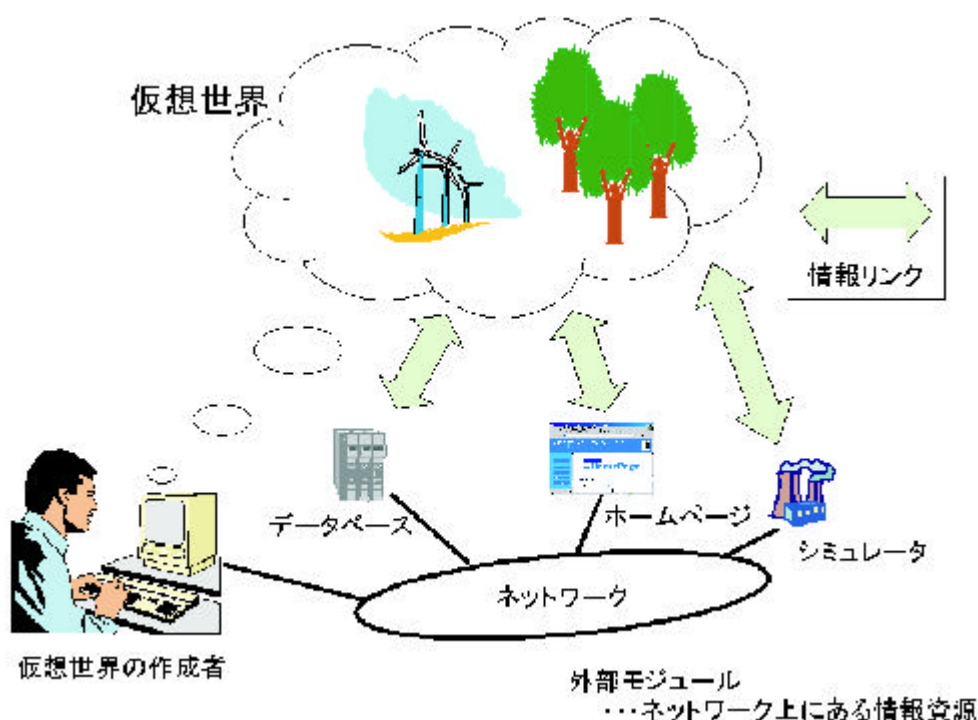


図 3.4: 情報資源を活用した仮想世界の作成の概念

上記のように、外部モジュールと仮想環境の間でのデータの入出力を行うためには、次に示す 3 つの機能を DVE システムに備える必要がある。

機能 A. データの入力：情報資源を外部モジュールとして情報リンクを通じて仮想環境に取り込むことができる機能。

機能 B. データの出力：外部モジュールとして扱う情報資源に情報リンクを通じて情報を送ることができる機能。

機能 C. データの転送：ある外部モジュールから仮想環境に取り込んだ情報を、更に他の外部モジュールに送ることができる機能。

なお、本研究の範囲内では、外部モジュールがネットワーク上に存在すること、あるいは仮想世界の作成者が外部モジュールを簡単に作成できることを前提とした。また複数のロケールから成る大規模な仮想世界ではなく、1つのロケールだけで構成する比較的小規模な仮想世界を作成する場合についてのみ考慮する。

3.2.2 ネットワーク情報資源活用型 DVE の処理方法

仮想世界には、ユーザの化身であるアバタの他に家や木など様々なものが存在する。それら仮想世界に存在する全てのものを仮想オブジェクトと呼び、これらは姿勢・形状など様々な情報を持っている。提案するシステムでは、外部モジュールが持つ情報と仮想オブジェクトが持つ情報を結合（リンク）することによって外部モジュールが持つ情報を仮想世界に活用する。

ここで図 3.5 に示すような仮想世界を例に処理の方法について説明する。図 3.5 は、ホームページ上の情報を用いて仮想世界で風を吹かせ、その風で仮想の風力発電機の発電量を処理し、発電機の羽根を回転させる仮想世界である。ここでは、実際に風を発生させる処理はそれだけで複雑な処理であるが、これを外部モジュール（ホームページ）から仮想世界に与えることにより、容易にコンテンツを増やしている。ここで用いる外部モジュールは、風の強さの情報が掲載されたホームページと、風速データから風車の姿勢データと発電量データを処理する風車シミュレータである。仮想オブジェクトは、風速データを持つ風オブジェクトと、姿勢データと発電量データを持つ風車オブジェクトである。

仮想世界での風車の姿勢と発電量を決定する処理を例に、DVE システムにおいて外部モジュールが持つ情報を仮想世界に利用する方法を以下に示す。

1. 3.2.1 項で述べた機能 A により、ホームページ上のデータを風オブジェクトの風速データとして取り込む。

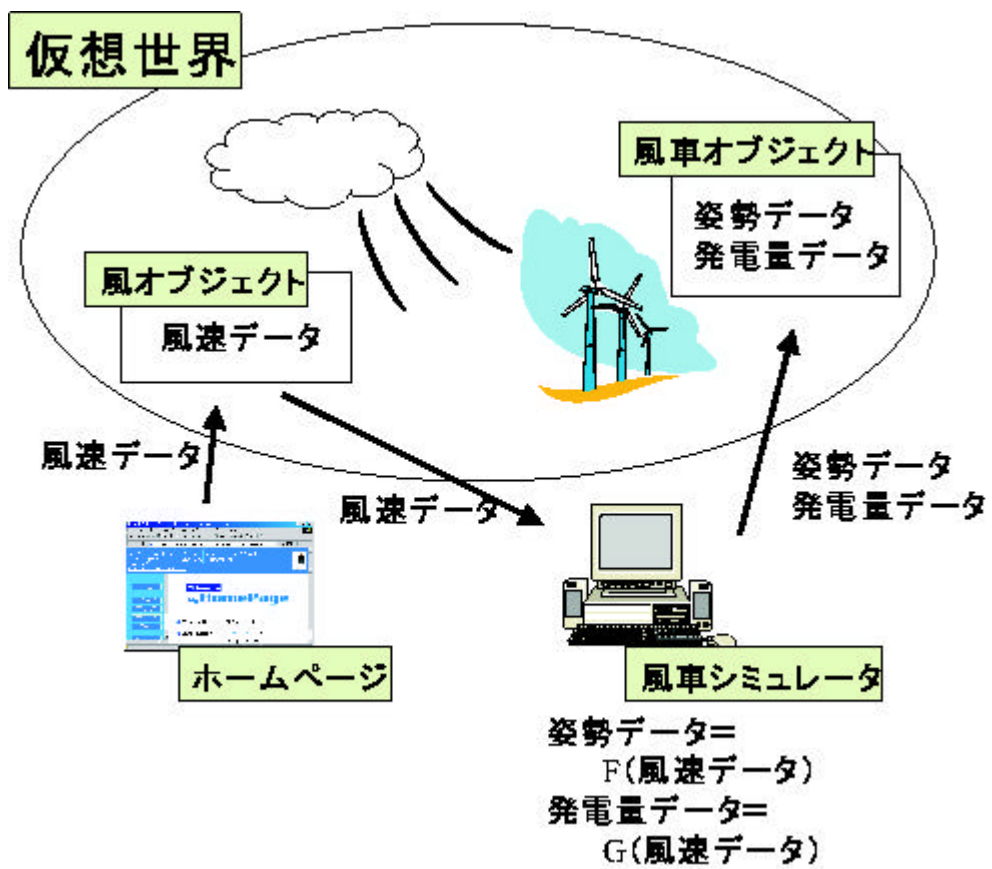


図 3.5: ネットワーク情報資源活用型 DVE の処理例

2. 機能 B/C により、風オブジェクトの風速データを風車シミュレータの風速データとして転送する。
3. 機能 A により、風車シミュレータの姿勢データ、発電量データを、それぞれ風車オブジェクトの姿勢データ、発電量データとして取り込む。

このようにして、ホームページ上の風速データをもとに仮想世界の風車オブジェクトの羽根を回し、発電量を出力することができる。

3.3 ネットワーク情報資源活用型 DVE の実現に向けた課題

3.2 節で述べたように、ネットワーク上の情報資源を活用することにより、コンテンツに特化した処理を行う計算モジュール (3.2.2 項の例では、仮想世界の中に風を発生させる処理) を作成することなく仮想世界を作成することを可能にする。本節では、3.2.2 項のような仮想世界の処理において、DVE システムが行う処理と仮想世界の作成者が設定すべき項目についてまず整理する。次に、より容易に仮想世界を作成可能なシステムを構築するための課題について述べる。

3.3.1 処理の設定方法

3.2.2 項で述べたシミュレーションを例にとると、情報資源を外部モジュールとし DVE に組み込む際に必要な機能を考えると以下の機能が挙げられる。

- ² データの送受信機能：情報リンクを介して DVE システムとデータを送受信する機能。
- ² データの処理機能：DVE システムから受信したデータを保持し、そのデータをもとに処理を行う機能。

一方、DVE システムに必要な機能は以下のものである。

- ² データの送受信機能：情報リンクを介して外部モジュールとデータを送受信する機能。

- 2 情報リンクの管理機能：どのような外部モジュールが接続されていて、どの情報リンクを用いてデータを送受信するかを管理する機能。
- 2 仮想オブジェクトの管理機能：仮想オブジェクトを管理し、送受信するデータを格納する機能。
- 2 仮想世界シミュレーション機能：仮想オブジェクトが格納するデータをもとに仮想世界をシミュレーションする機能。

このような機能を具備するシステムを構築すれば、作成者は次のような手順を経て仮想世界を作成することができる。

1. 仮想オブジェクトの作成：3.2.2 項の風オブジェクトや風車オブジェクトのような仮想オブジェクトを作成する。同時に、風オブジェクトが風速データをもつように、仮想オブジェクトが保持するデータを設定する。
2. 外部モジュールの準備：風速データを得ることができるホームページ等、作成する仮想世界で必要となる外部モジュールを作成、もしくは既存のものを利用する。
3. 情報リンクの作成：仮想オブジェクトと外部モジュールとの関連を記述して、外部モジュールのどのデータと仮想環境システムのどのデータをやりとりするかを設定する。
4. 処理順序の設定：複数の外部モジュールをコンテンツの処理に用いる場合は、処理の順序やタイミングを設定する。例えば風車オブジェクトの発電量は、風シミュレータ上の風速データの値が変化すれば、風シミュレータ 風車シミュレータという順で処理するように設定する。

3.3.2 解決すべき課題

ここまでは、ネットワーク上に存在する情報資源を仮想環境に組み込む点に注目して議論を進めてきた。

しかし、本研究では、より容易に仮想世界の作成が可能なシステムを提供すべきであると考えられる。ここで特に問題と考えたのは、前項で述べた処理の設定方法のうち、処理順序の設定である。すなわち、仮想世界の作成者が、外部モジュールについてその性質を熟知し、仮想世界全体の外部モジュール同士の関係について把握し正しい処理

順序を設定するには、大きな労力が必要になると考えた。よって、本研究では情報リンクを作成するだけで、つまり外部モジュールと仮想オブジェクトの関係を決定しさえすれば、後は自動的にシステム側で正しい処理順序で処理することが可能な DVE システムを構築することを考えた。

このために、DVE システムが自動的に「因果関係」を管理できる手法の考案を、本研究の課題に含めることとする。ここで因果関係とはある過去の出来事(因)により未来の出来事(果)が生じることであり、因果関係が成立する事象では、過去の処理を行う前に未来の処理を行ってはいけないという制限ができる。そのため未来の処理を先に行った場合、その結果に矛盾が生じてしまう(因果律エラー)。本システムのように複数のモジュールで処理する場合、モジュール同士が関連することにより因果関係が発生するため、因果関係の管理をするためにはモジュール同士の関係を管理する必要がある。

この際、提案する DVE ではネットワーク上に存在する複数の外部モジュールを利用して仮想世界のシミュレーションを行っているため、特にリアルタイムで処理を行うことに着目して手法を考案する。なぜなら、ネットワークの状態によって遅延が発生する場合や、大規模な計算を行うため長い処理時間を必要とする外部モジュールを利用する場合に仮想世界のシミュレーションがリアルタイム性を損なうからである。このような遅延は DVE では不可避である。しかし、遅延が小さい場合には利用者に見かけ上のリアルタイム性を提供できるため、本研究ではリアルタイム性を重視しつつその中で因果関係を管理する手法を考案する。

3.4 因果関係を考慮した処理の手法

このネットワーク情報資源活用型の DVE システムでは、コンテンツに特化した処理は全て外部モジュールで行い、多数の外部モジュールが仮想世界のシミュレーションの一部を並行して処理する形態を用いることを前提とする。

本節では、多数の外部モジュールが接続されても処理結果の因果関係に矛盾を生じさせないように、DVE システムと外部モジュールを管理する手法を提案する。この方針として、並列離散事象シミュレーション (Parallel Discrete Event Simulation; PDES)^[26] 分野の研究で提案されている因果関係の管理手法をもとに、ネットワーク情報資源活用型 DVE の処理手法を検討する。

以下では、まず PDES について説明する。次に、外部モジュールの入出力データの

因果関係の管理方法について述べる。最後に仮想世界のリアルタイム処理について検討した後、処理方法について述べる。

3.4.1 並列離散事象シミュレーションにおける因果関係の管理手法

計算機を用いた計算処理手法は、時間の扱い方の観点で分類すると、気象シミュレーションのように時間的に連続して追跡する計算処理と、特定の出来事の生起する瞬間(事象)に着目して、その事象によるシステムの内部状態の変遷を追跡する離散事象シミュレーションに大別できる。PDESとは、離散事象シミュレーションを並列計算機や分散システムを用いて行うもので、対象とするシミュレーションタスクを分割し、それを互いに通信しあう論理プロセス(Logical Process; LP)に割り当て並列に処理を行う。

PDESでは、因果性のある未来の事象処理を、過去の事象処理が終わる前に行なうてはいけないという制約がある。しかし、複数のLPにより並列して処理を進めることにより各LPが生起時刻の異なる事象を処理するため、正しい因果関係を維持できなくなる(因果律エラーが発生する)可能性が生じる。そこで、各LPは事象が発生した場合に他のLPとメッセージを交換することにより因果関係を管理して、並列に処理を行う(図3.6)。メッセージには、事象の内容の他に事象の生起時刻をタイムスタンプとして付加する。このようなメッセージ交換の方法には、保守的プロトコルと楽観的プロトコルという2つの代表的な方法がある。以下にこれらのプロトコルについて説明する。

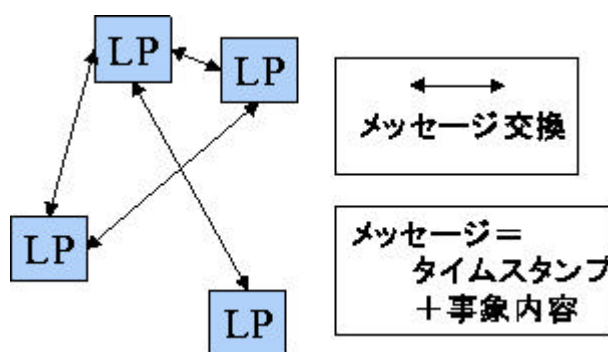


図 3.6: PDES の構成

保守的プロトコル この方法は、常に最小のタイムスタンプを持つ事象(最も古い事象)から処理を行う方法である。そのため、処理する事象のタイムスタンプが最小であることを常に保証する必要がある(以後、最小タイムスタンプを保証するという)。通

常、ある LP に送信する送信元 LP は固定されていて、また送信元 LP から送信されるメッセージのタイムスタンプは常に小さい順に送信されるという仮定のもとに、すべての LP からのメッセージが到着してから比較して最小タイムスタンプを保証する。

しかし、何らかの原因でメッセージが届かなかったりすることにより最小タイムスタンプが保証されなかった場合は、処理が完全に停止してしまう。この現象はデッドロックと呼ばれる。デッドロックを回避するためには、事象が起こらなくてもタイムスタンプのみの空メッセージを送信し、最小タイムスタンプを保証するヌルメッセージプロトコル^[27] や必要に応じて他の LP に次のタイムスタンプを問い合わせる問い合わせメッセージプロトコル^[28] がある。しかし、これらは unnecessary メッセージの送信を行うため、多くのメッセージの発生を導く。

楽観的プロトコル この方法は、最小タイムスタンプが保証されなくても事象処理を行う方法である。そのため、これまでに処理した事象のタイムスタンプより、小さいタイムスタンプを持つメッセージが届いた場合は因果律エラーが発生したことになり、そのタイムスタンプの時刻まで戻って処理を行う必要がある。これをロールバックと呼ぶ。ロールバック処理を行うにはその時刻の履歴が必要であり、そのためメモリを多く消費してしまう。

この2つのプロトコルの比較を表 3.2 に示す。

表 3.2: 保守的プロトコルと楽観的プロトコルの比較

比較項目	保守的プロトコル	楽観的プロトコル
処理時間	プラットフォームや実装方法によって異なる	同左
メッセージ送信頻度	高い	低い
メモリの消費量	少ない	多い
デッドロック	発生すると処理が停止	なし
ロールバック	なし	正しい時刻まで戻り再処理

3.4.2 仮想世界における因果関係の管理手法

DVE システムにおいて考慮すべき点 VR のシミュレーションでは、利用者からの入力に基づいてイベント処理を行い、その結果を用いて仮想世界を計算するという、離散

的なシミュレーションが一般に広く採用されている^[29]。例えば仮想物体の運動を処理する場面では、物体毎に利用者とのインタラクションを規定するフラグを設ける、あるいはペトリネット^[30]を利用する等の方法で、本来は連続事象である仮想物体の運動を、離散的処理と連続的処理を併用することにより行われることが多い。本研究においても、計算や通信の負荷を軽減させる観点から、仮想空間の変化を離散事象としてモデル化し、その処理を行う手法が有効であると考えられる。

一方、ネットワーク情報資源活用型 DVE のシステム構成は、接続された複数の外部モジュールが並行して処理を行うことができる仕組みになっている点に着目すると、仮想世界全体を並列にシミュレーションしていると捉えることができる。

以上の理由から、本研究では PDES におけるプロトコルをもとに、因果関係の管理手法を検討する。PDES を本システムに適用する際、ロケールサーバや外部モジュールが PDES における LP、各外部モジュールの処理結果が内部状態、処理結果の変化が内部状態の変化とそれぞれ対応していると考えられる(図 3.7)。そうすると、DVE での外部モジュールのデータ通信はメッセージ交換に相当し、処理結果が変化したときに送信される。また、DVE を保守的プロトコルで管理する場合、デッドロックを回避するため処理結果が変化しない場合にもデータ通信を行う必要がある。また、楽観的プロトコルでは、処理結果が変化した場合のみデータ通信を行えばよいが、ロールバック処理を行うため過去の処理に用いた入力データを履歴として記録する必要がある。

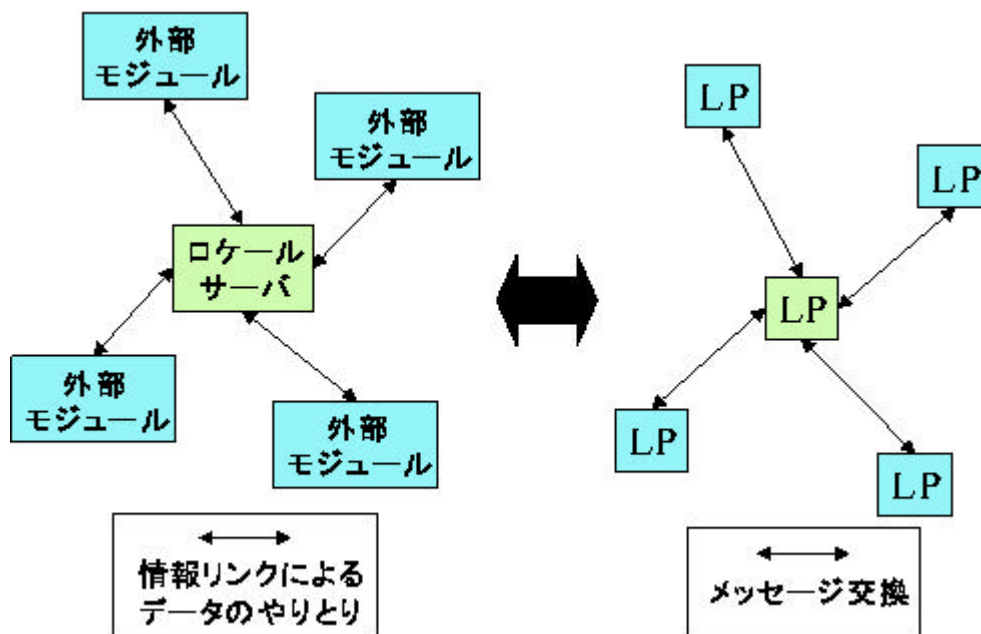


図 3.7: PDES を適用した DVE

ここでまず、外部モジュールとロケールサーバの仮想オブジェクトが図3.8のような接続関係をもつ場合を考える。3.4.1項で述べた2つのプロトコルによって、因果関係を管理するデータの流を比較した結果を図3.9に示す。ここでは、タイムスタンプとしてロケールサーバで作成される仮想世界の提示画像用のFrameに番号をつけて、その番号をタイムスタンプとして用いる。また、図3.9では、横軸が時間で矢印はデータの流を示す。

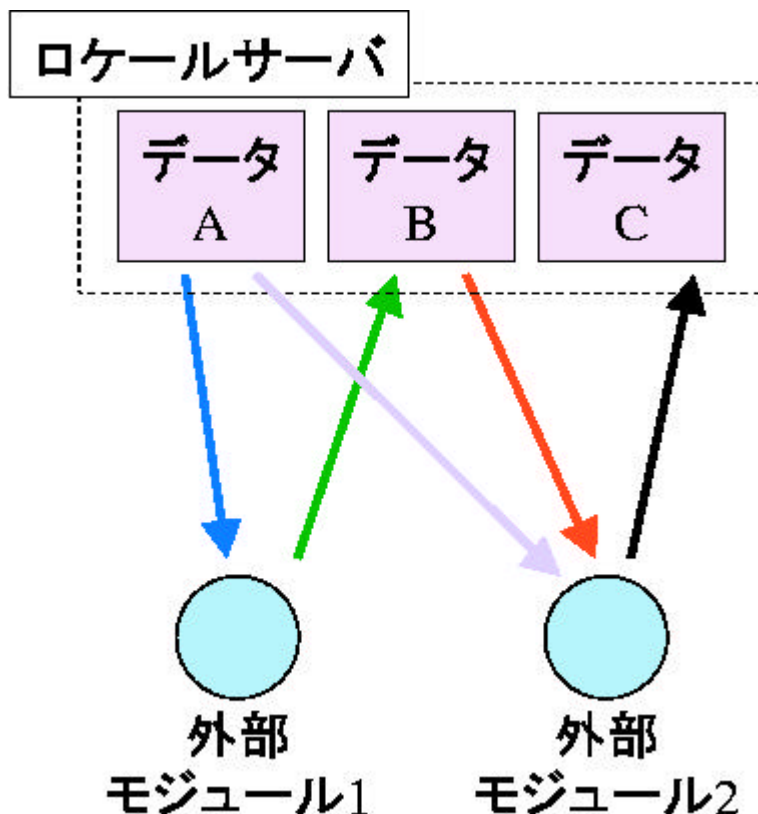
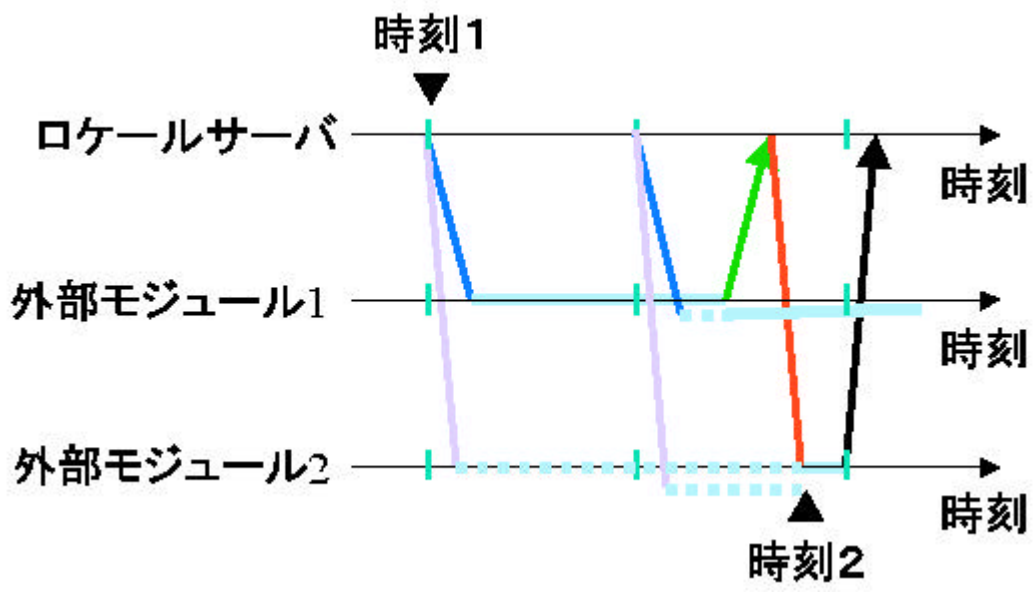


図 3.8: 外部モジュールとロケールサーバの接続例

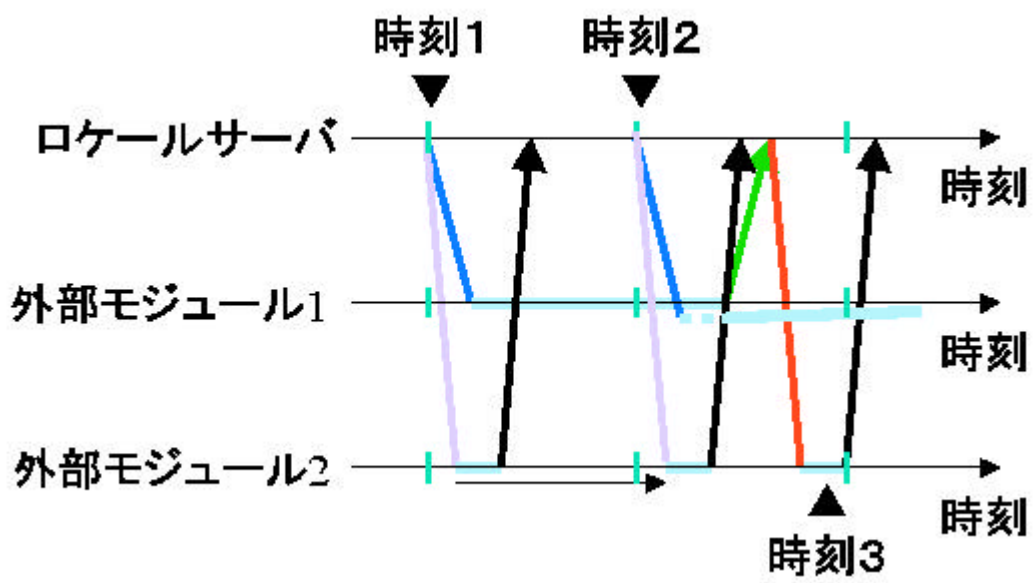
1の時刻でデータAが変化すると仮定すると、そのデータは外部モジュール1、2に送信される。

保守的プロトコルでは、外部モジュール1では入力データがデータAしかないためそのメッセージが最小であると保証されるため処理される。外部モジュール2ではデータBからより小さいタイムスタンプのメッセージが届く可能性があるため処理を行わず待機する。そして時刻2になって初めて処理が行われる。

楽観的プロトコルでは、タイムスタンプが最小であることが確定しなくても処理を行うため、時刻1においてデータAが変化するとき、外部モジュール1、2共に処理を



(a) 保守的プロトコル



(b) 楽観的プロトコル

図 3.9: 保守的プロトコルと楽観的プロトコルによるデータの流れの比較

行う。時刻2についても同様に処理される。時刻3で時刻1のタイムスタンプが付加されたデータBが外部モジュール2に送信されるため、ロールバックが発生する。このとき、外部モジュール2は時刻1のデータまで戻って処理をやり直す。

各プロトコルで因果関係の管理を行った場合には上記のようにデータのやりとりを行うが、その性能は対象となる計算処理の内容やプラットフォームにより大きく異なる。そのためDVEの以下のような場合について考慮し、管理方法について検討する必要がある。

1. 処理時間が長い大規模な処理が外部モジュールとして組み込まれる可能性がある
2. 多数の外部モジュールが互いに関連する可能性がある
3. 外部モジュールはネットワーク上で利用される

これらの特徴が3.4.1項で述べた2つのプロトコルに及ぼす影響について検討する。保守的プロトコルでは処理時間の大きな外部モジュールを組み込んだ場合(1の場合)、最小タイムスタンプが保証された事象のみ処理するため、その外部モジュールの出力データを利用する全ての外部モジュールは、その処理が終わるまで待機するため処理停止してしまう。関連した外部モジュールが多い場合(2の場合)、関連する外部モジュールの出力を待つ必要があり、その時間が余分にかかる。ネットワークで利用する場合(3の場合)、メッセージ交換頻度が高いためネットワークに負荷をかけてしまう。また、最小タイムスタンプが保証されるまで待機するため、メッセージの到着の遅れが処理の遅れに直結するため、性能がネットワーク環境に大きく依存してしまう。

一方、楽観的プロトコルでは、1の場合、最小タイムスタンプが保証されない事象も処理されるため、他の外部モジュールに与える影響が少ないが、ロールバックが発生した場合、仮想世界では間違った処理結果が表示されてしまい、また処理時間が長い分だけそのオーバーヘッドが大きくなり、間違った処理結果が訂正され正しい因果関係が復元されるまでに時間がかかってしまう。2の場合、間違ったデータが次々と送信されるため雪崩的にロールバックが発生し、整合性を著しく損ない、利用者に違和感を与えてしまうことがある。また3の場合、最小タイムスタンプが保証されていない事象も処理し先読み計算を行っているためネットワークの通信時間を短縮できる。しかし、ネットワークの遅延などによるメッセージの到着順の入れ替わりが起こり、ロールバックを引き起こす可能性がある。

本研究では、いずれの方法もユーザに違和感を与えないという現実性、リアルタイム性を兼ね備えていないためネットワーク情報資源活用型DVEには適していないと考

え、現実性とリアルタイム性を考慮した独自の手法を考案した。なお、PDESにおける因果関係の管理手法をVRの分野に応用する研究は、既にHybinetteらにより発表されている^[31]。しかしその手法は、精度の高い画面を提示するために、並列計算機を利用して描画する画面をリアルタイムで処理する場合の手法であり、提案するDVEに適用することはできない。

因果関係の管理方針 DVEでは、仮想世界の時間が逆行する因果律エラーは現実性を損なってしまうため、避けたい現象である。例えば、人がボールに近づいてボールを把持する場合、通常はボールに接触した時点でボールが人の動きに追従する。しかし因果律エラーが発生すれば、人がボールに触れてもボールが人の動きに追従しない、あるいはボールに触れていなくてもボールが人の動きに追従ようになる。現実性の観点からは常に因果関係を考慮して正しい処理を行っている保守的プロトコルの方が優れている。

しかし、仮想世界のリアルタイム性の点から考えると、前述したように保守的プロトコルでは処理が停止する恐れがあり、また保守的プロトコルではネットワーク環境に大きく依存するため、楽観的プロトコルの方が優れている。

2.1.3項で述べたように、DVEシステムではリアルタイムにシミュレーションを処理することが必要不可欠であり、また現実性についてはロールバックの時間が短ければ間違った処理結果を提示するのは一瞬であるため、仮想世界のリアルタイム性に重きを置き、本研究では、DVEシステムには楽観的プロトコルの方が適していると考えた。

しかし、特に多数の外部モジュールが互いに関連する場合には、ロールバックが雪崩式に発生してしまう恐れがある。図3.7、3.6を比較すると分かるように、本システムではPDESとは異なり、ロケールサーバを中心にメッセージ(データ)のやりとりを行っているため、集中的にメッセージ管理することが可能である。よって、ロケールサーバで送受信するメッセージを管理し、因果律エラーを起こしたデータが他の外部モジュールへ送信されることを防ぐことにより、ロールバックが雪崩式に発生することをできる限り減少させることが可能である。

メッセージの管理手法 雪崩式にロールバックが発生する原因は、因果律エラーを起こしたデータを他の外部モジュールに送信するためである。本システムでは、外部モジュールはロケールサーバとのみメッセージ交換を行っている点に注目し、ロケールサーバで全てのデータを管理し因果律エラーを起こしたデータを破棄する、また因果律エラーを起こす可能性のあるデータをすぐに他の外部モジュールに送信せずに、一

時保留することで、雪崩式ロールバックの発生を減少させる。以下、外部モジュールの処理結果を利用者に提示することを外部モジュールの処理結果を仮想世界に反映する、すぐに反映せずにデータを保存することを保留すると定義する。

本研究では、外部モジュールで処理された結果に、確認値をメッセージに付加することによって、データが因果律エラーを起こしているか否かを判断しメッセージを管理する手法を提案する。確認値とは、外部モジュールがどのタイムスタンプのデータを用いて処理をしたかを示すものであり、処理に用いたデータのタイムスタンプの平均値を用いる。時刻 t (タイムスタンプは T_{stamp}) における確認値 $K_{Check;t}$ を以下の式で表す。

$$K_{Check;t} = \frac{\sum_{i=1}^N T_{stamp;t;i}}{N} \quad (i = 1; \dots; N) \quad (3.1)$$

ただし、 N とは外部モジュールの入力データの数、 $T_{stamp;t;i}$ とは時刻 t において処理に使用した i 番目のデータのタイムスタンプである。ここで時刻 t において処理に用いられるデータはそれ以前に変化したデータなので、 T_{stamp} 、 $T_{stamp;t;i}$ となり、確認値は T_{stamp} 、 $K_{Check;t}$ となる。

この方法では、まず、ロケールサーバから外部モジュールにデータを送信する際に、ロケールサーバで送信したデータのタイムスタンプの履歴を記録しておき、その履歴を用いて外部モジュールから送られてくるデータの確認値を予測する (予測確認値 K_{forest})。次に、外部モジュールでは受信したデータを用いて処理し、用いたデータのタイムスタンプより確認値を計算しデータに付加して送信する (実確認値 K_{fact})。ここで、送信したデータが届くまでに時間がかかるため、 K_{forest} 、 K_{fact} となる。ロケールサーバでは、受信したデータの実確認値を用いてそのデータが因果律エラーをしているかどうかを評価する。

具体的には、 T_{stamp} 、 K_{forest} 、 K_{fact} の値を比較してデータを以下の3種類に分類し、それぞれ異なる処理を行うことにより雪崩式ロールバック発生を減少させる。また、 T_{stamp} 、 K_{forest} 、 K_{fact} の大小関係は、 T_{stamp} 、 K_{forest} 、 K_{fact} となる。

確実なデータ $T_{stamp} = K_{forest} = K_{fact}$ を満たすデータ。これは、タイムスタンプ T_{stamp} の処理を行うときに使用した全てのデータのタイムスタンプが T_{stamp} なので、絶対に因果律エラーを起こさないデータである。よって、このデータはすぐに他の外部モジュールに送信する。

不確実なデータ $T_{stamp} \neq K_{forest} = K_{fact}$ を満たすデータ。このデータは送信したデータを用いて処理しているが、タイムスタンプ T_{stamp} の処理を行うときに使用し

た全てのデータのタイムスタンプが T_{stamp} ではない。そのため、他の外部モジュールからタイムスタンプ T_{stamp} のデータが送信された場合は因果律エラーを起こしてしまう。よって、このデータはロールバックの発生する可能性のあるデータである。このデータの扱いについては、後で検討する。

間違ったデータ $T_{stamp} \notin K_{forest} \notin K_{fact}$ を満たすデータ。これは送信したデータとは異なったタイムスタンプのデータを用いて処理しているため、因果律エラーを起こしたデータである。よって、確実にロールバックの発生するデータである。よって、他の外部モジュールに送信せず破棄することとする。

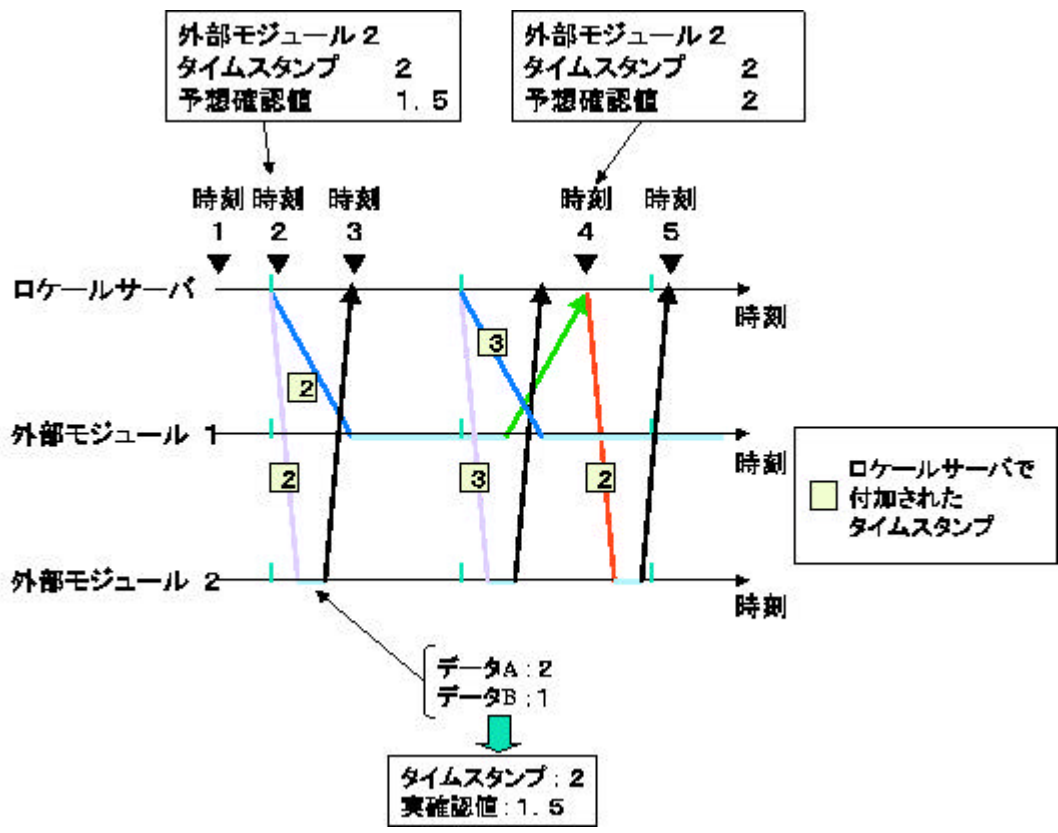
不確実なデータの扱い方について考えると、不確実なデータは因果律エラーを発生させる可能性があるだけで、必ず発生する訳ではない。以下、外部モジュールとロケールサーバの関係が図 3.8 で、データの流れが図 3.10 の場合の具体例を用いて、不確実なデータの扱いについて検討する。図 3.10 では横軸が時間で、矢印はデータの流れを示す。

まず不確実なデータが生じる場面について考える。図 3.10(a) では、時刻 1 で全てのデータのタイムスタンプが 1 であるとする。時刻 2 においてデータ A が変化した時、ロケールサーバはデータにタイムスタンプ 2 を付加して送信する。この時ロケールサーバでは予測確認値を計算する。タイムスタンプ 1 の予測確認値は、外部モジュール 1 についてはタイムスタンプ 2 のデータ A のみを利用して処理するため 2、外部モジュール 2 については、タイムスタンプ 1 のデータ A とタイムスタンプ 2 のデータ B を用いて処理するため $(2 + 1) \times 2 = 1:5$ となる。外部モジュール 2 では受信したデータを用いて処理を行い、実確認値 1.5 を付加して送信する。時刻 3 においてロケールサーバは外部モジュール 2 からデータを受信するが、このデータは予測確認値とは一致するがタイムスタンプとは一致しないので不確実なデータとなる。

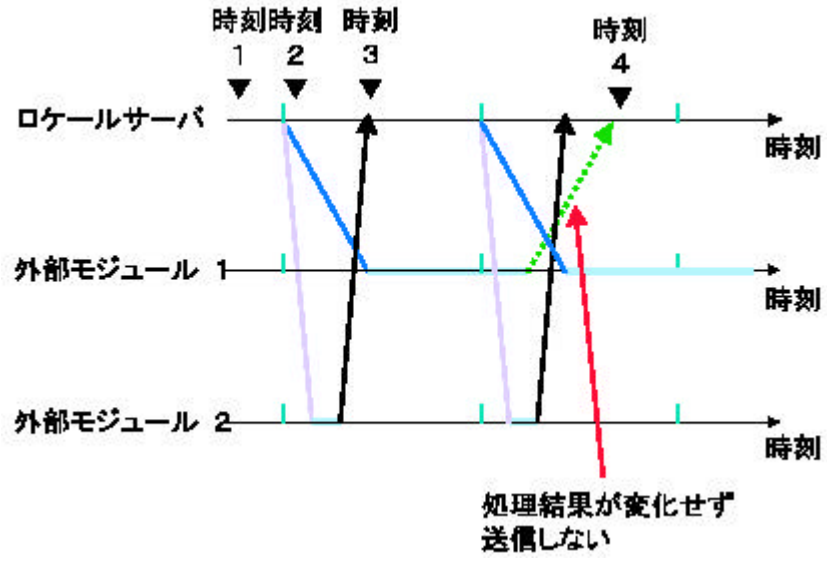
ここで、因果律エラーが発生する場合と発生しない場合について説明する。

例えば図 3.10(a) の場合、時刻 3 で受信したデータをすぐに反映すると時刻 5 で因果律エラーが発生しロールバックを行う必要がある。メッセージ管理を行った場合、時刻 4 で外部モジュール 2、タイムスタンプ 2 の予測確認値が 2 となり、時刻 3 に受信したデータ実確認値と異なるため、因果律エラーが発生したことを検知することができる。この場合では、時刻 3 で受信したデータを時刻 4 まで保留しておき、時刻 4 で因果律エラーを検知した時に破棄することによりロールバックの発生を防ぐことができる。

また、図 3.10(b) の場合、外部モジュール 1 で処理結果が変化しないため、時刻 3 で



(a) 因果律エラーが発生する例



(b) 因果律エラーが発生しない例

図 3.10: データの流れの例

受信したデータを反映しても因果律エラーが発生しない。この場合、時刻 3 で反映した方がタイムラグが小さくなる。

このように不確実なデータは、因果律エラーが発生する場合と発生しない場合がある。前者の因果律エラーが発生する場合はデータを保留した方がよく、後者の発生しない場合はすぐにデータを反映した方がよい。

よって、不確実なデータを時間 T だけ保留してその間に因果律エラーが発生した場合はデータを破棄し、発生しなかった場合は反映させる処理を行う。この時間 T を保留時間と呼ぶ。この場合、因果律エラーが発生するデータを受信してから時間 T 以内に検知できれば因果律エラーを防ぐことができるが、時間 T 以上経過してから検知した場合は因果律エラーを防ぐことはできない。また、因果律エラーが発生しないデータについては、時間 T だけ仮想世界に反映するのが遅れるため、設定する保留時間の値は重要である。しかし、因果律エラーが検知できるまでの時間は、外部モジュールごとに異なり、また、関連する外部モジュールにより影響を受けるため保留時間の最適値を求めることは非常に難しい。本研究では、リアルタイム性に与える影響が比較的小さいと考えられる、仮想世界を 1 Frame 描画するために必要な時間 (本研究では 100ms) を保留時間とした。

3.4.3 仮想世界のリアルタイム処理について

仮想世界のシミュレーションをリアルタイムで行うため、以下のことについて検討を行った。

外部モジュールの処理方法 PDES では、メッセージはそのタイムスタンプの小さいものから 1 つずつ処理されていく。本システムでは、処理時間の大きな外部モジュールの場合、データを受信する速度が処理する速度を上回ってしまう。このような場合、1 つずつ処理していくと受信データが次第に増加し、その外部モジュールで処理されるデータが古いものになってしまう。

よって、多数の受信データが存在した場合は以下のように処理する時刻を決定し、全てのデータを一度に処理する。

1. 受信したデータのタイムスタンプの中で、最大のものと最小のものを求める。
2. 最小のものが現在までに処理した時刻より小さいタイムスタンプであった場合、因果律エラーが発生するため、その時刻まで戻って処理する必要がある。よって

この場合は、最小のタイムスタンプを次に処理する時刻とする。

3. 最小のものが現在までに処理した時刻より大きいタイムスタンプであった場合、因果律エラーは発生していない。よって最近の時刻である最大のタイムスタンプを次に処理する時刻とする。
4. 処理する時刻が決定すると、その時刻に最も近い過去のデータを全てのデータについて履歴から参照し、そのデータを用いて処理を行う。

外部モジュールの待機時間 外部モジュールでは処理の種類により2つに分けれる。1つは時間が経過するにつれて出力が変化するもので、もう1つは時間に関係なく入力に変化しなくても出力が変化するものである。これらは外部モジュールの性質によるもので、前者の場合は一定時間毎に処理する必要がある、後者の場合は新しいデータを受信するまで処理する必要はない。よって、外部モジュールは一定時間ごとに処理を行い、処理間隔は作成者が設定するものとする。この時間を待機時間と呼ぶ。

ただし、ロケールサーバにおける仮想世界の更新間隔より頻繁に処理しても、仮想世界に反映されないため、待機時間は仮想世界の更新間隔より長いものとする。

3.5 処理の流れ

本節では、これまで述べてきた方法で、各モジュールがどのように処理するかをまとめ。以下にロケールサーバ、外部モジュール、ブラウザについて説明する。

3.5.1 ロケールサーバ

ロケールサーバでは、外部モジュールとのデータの送受信とメッセージ管理、仮想世界へのデータの反映を行う。図 3.11 にその処理を示す。

ロケールサーバは以下の処理を繰り返しながら、仮想世界のシミュレーションを行う。ロケールサーバは一定間隔で仮想世界を更新しているため、仮想世界の時刻、つまりタイムスタンプには仮想世界の提示画像用 Frame の番号を用いた。

1. 受信データの有無を確認する。なかった場合は4へ。
2. 受信データから取得した実確認値と予想確認値を比較し、それぞれ処理を行う。確実なデータはすぐに反映する。不確実なデータは保留し4へ。間違ったデータは破棄し4へ。

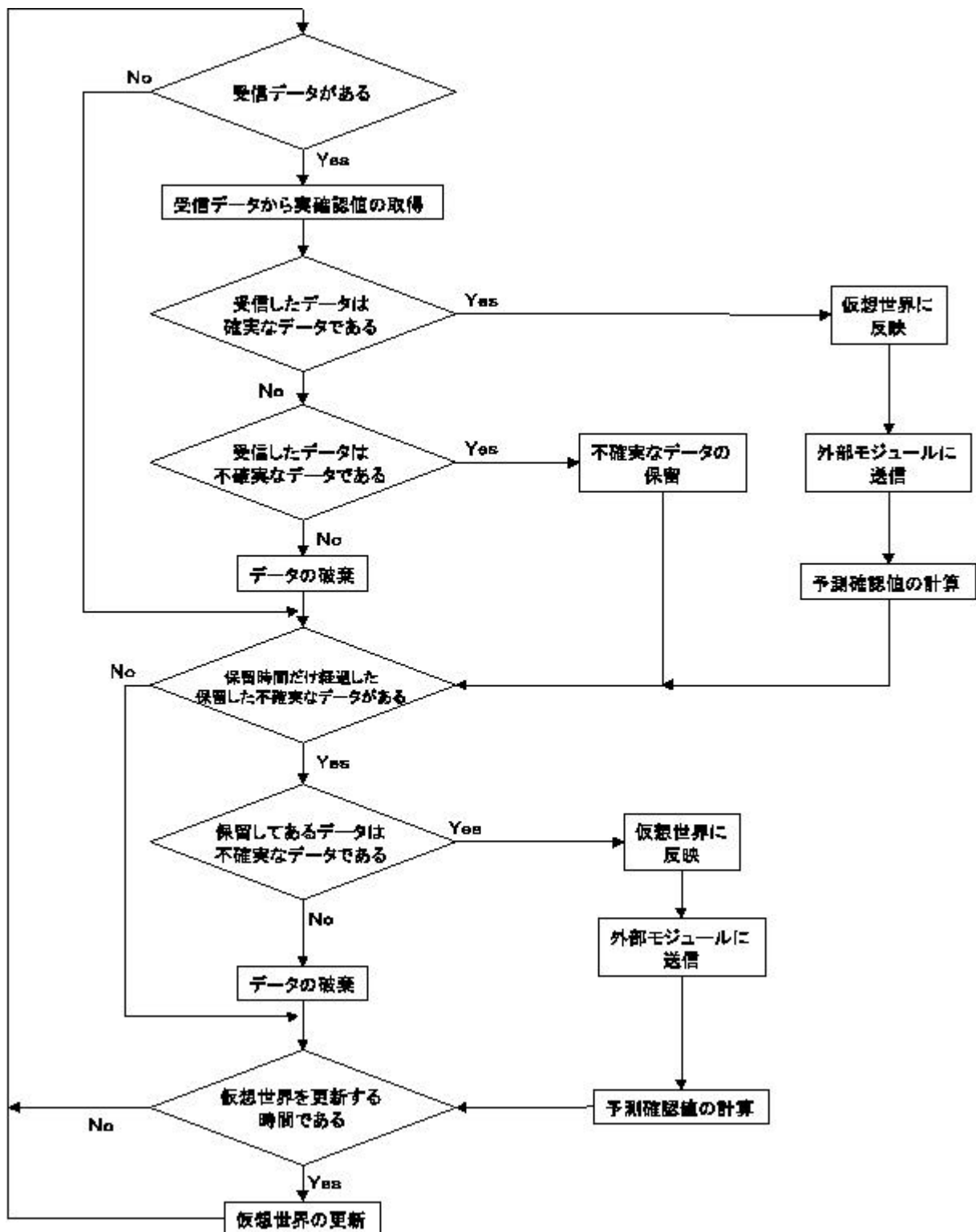


図 3.11: ロケールサーバの処理の流れ

3. 反映したデータを外部モジュールに送信する。このとき、送信するデータをもとに予測確認値を計算する。
4. 保留時間だけ経過した不確実なデータの有無を確認する。経過したデータは、再び予測確認値と比較し不確実なデータと判断した場合は仮想世界に反映する。間違ったデータだった場合は破棄し6へ。
5. 3と同じく外部モジュールへ送信し、予測確認値を計算する。
6. 前回更新してから更新間隔だけ時間が経過したか確認する。経過していない場合は1へ。
7. 仮想世界を更新する。1へ。

3.5.2 外部モジュール

外部モジュールでは、データの送受信と情報資源が持つコンテンツ特有の処理を行う。図3.12にその処理を示す。外部モジュールでは、ロケールサーバから受信するデータのタイムスタンプで仮想世界の時刻を管理する。

1. 受信データの有無を確認する。なかった場合は4へ。
2. 受信データのタイムスタンプより処理する時刻を決定し、その時刻の処理に必要なデータを履歴より参照する。
3. 必要なデータをもとに処理を行う。
4. 時間が前回処理してから待機時間だけ経過したかを確認する。経過した場合は5へ。経過していない場合は1へ。
5. 最後に処理した時刻に(外部モジュールの待機時間)¥(ロケールの更新間隔)加えたものを次の処理をするタイムスタンプとする。2へ。

3.5.3 ブラウザ

ここまでは、ロケールサーバと外部モジュールなど仮想世界のシミュレーションに関係するモジュールについて述べてきた。仮想世界の利用者はブラウザを利用して仮想世界を体験することができるが、仮想環境から見るとブラウザも利用者が入力した

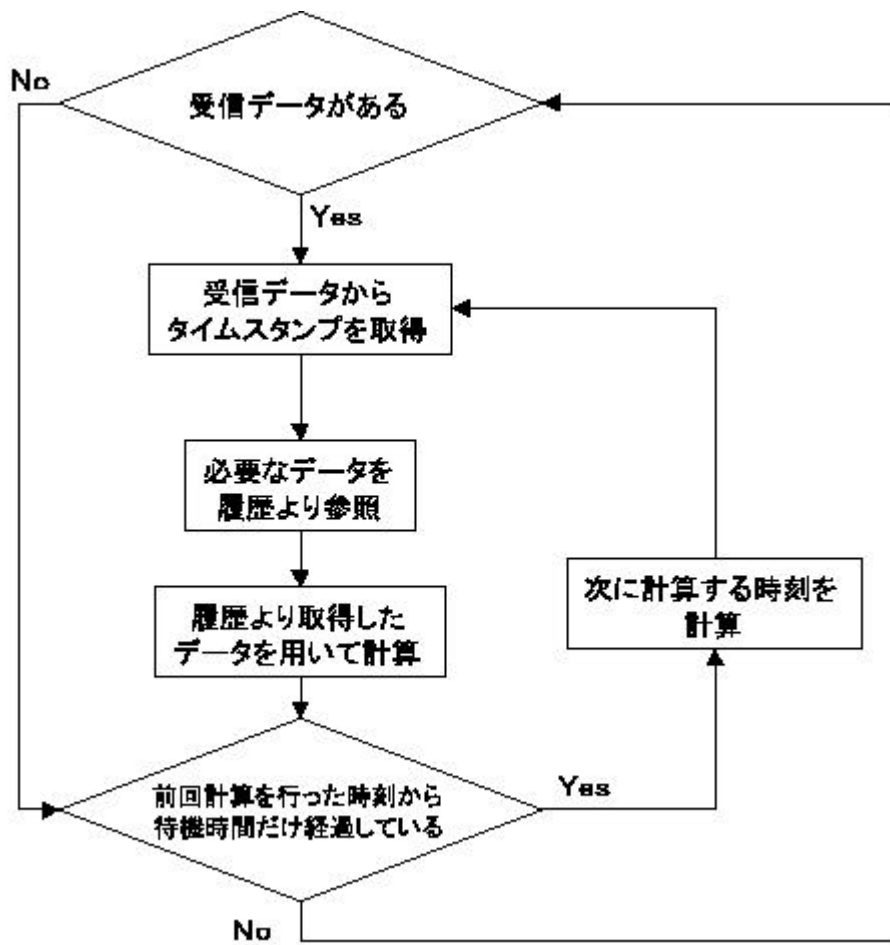


図 3.12: 外部モジュールの処理の流れ

情報を送信し、一方仮想世界の情報を受信して利用者に提示する外部モジュールと捉えることができる。よって、本研究ではブラウザは仮想世界を体験する際に必要であると考え基本的な機能のみを備え、コンテンツに特化した機能は外部モジュールによって拡張するものとする。

基本的な機能として、以下の3項目を挙げる。

1. 仮想世界を3次元画像で提示する機能。
2. 利用者の化身であるアバターを操作する機能。
3. 他の利用者とのコミュニケーションを行う機能。これは音声と文字によるコミュニケーションを行う機能である。

第 4 章 基盤システム MALIONET の開発

本章では、前章で提案したネットワーク資源活用型 DVE を実装した結果について述べる。以下では、構築した DVE システムを MALIONET (Multiple Access of Linked Information Over NETwork) と呼ぶ。まず、DVE システムの構成について説明し、次に仮想世界を定義するために用いる設定ファイルの仕様を述べる。最後に、本システムを用いて仮想世界を作成する手順を述べる。

4.1 システムの構成

本節では、構築する DVE システム MALIONET のハードウェア構成およびソフトウェア構成について述べる。

4.1.1 ハードウェア構成

ハードウェア構成を図 4.1 に示す。本研究で用いたハードウェアは、後述する DVE システムのソフトウェアを動作させる PC(Personal Computer) とその周辺機器、およびネットワーク設備で構成される。PC は全て、Microsoft 社の OS、Windows NT を使用できる PC/AT 互換機を使用した。ネットワーク機器には、100BASE-TX 対応の LAN とネットワークアダプタを用いた。周辺機器は、動作させるソフトウェアに応じて異なる。各種サーバ用の PC では、マウスやキーボード等の、ソフトウェアを操作するための一般的な周辺機器を備えていけばよいが、ブラウザを動作させるブラウザ用 PC は、仮想世界に参加するために以下のデバイスが必要となる。

² 仮想空間映像提示デバイス

仮想空間の映像を画面に提示する為に、グラフィック API (Application Programming Interface) である OpenGL^[32] に対応したディスプレイアダプタが必要である。描画結果はディスプレイを用いて利用者に提示する。

² 音声入出力用デバイス

音声による会話を行うためには、マイクとスピーカが必要である。これらは、PC

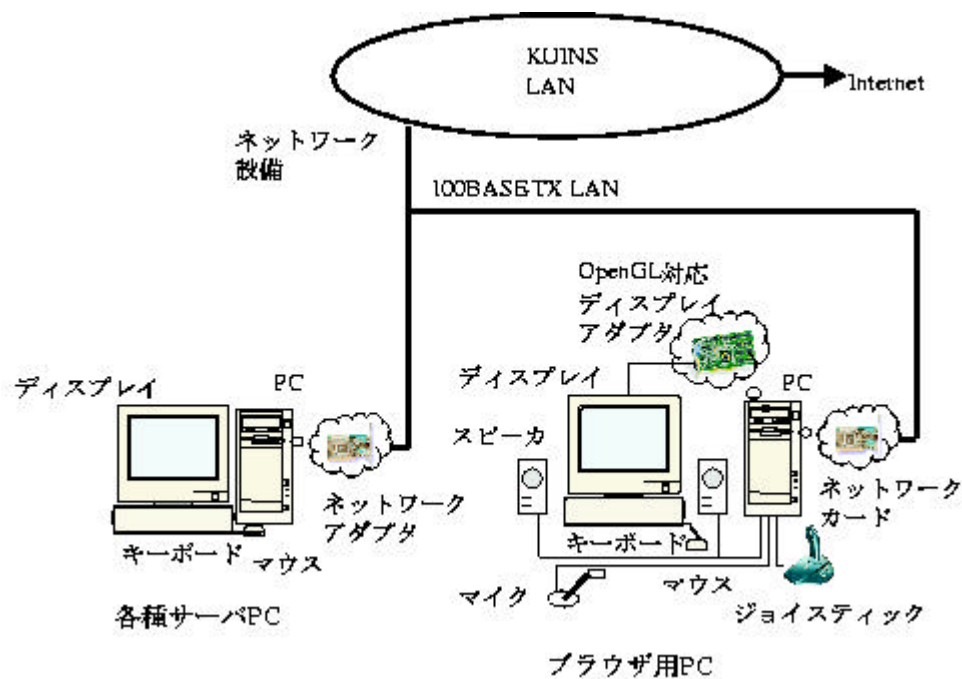


図 4.1: ハードウェア構成

のサウンドカードに接続する。音声の送信と受信を同時に行うため、サウンドカードには全二重対応のものを利用する。

2 アバタを操作する入力デバイス

仮想空間におけるアバタの動きを入力するデバイスとして Microsoft 社の Joystick(SideWinder 3D Pro) を使用した。

2 コミュニケーション用の入力デバイス

コミュニケーションのために様々な入力を行うデバイスとして、PC に接続されたマウスとキーボードを用いる。

なお、本システムは、表 4.1 に示す性能を備え、ネットワークに接続された PC (OS は WindowsNT) 上であれば動作可能である。

4.1.2 ソフトウェア構成

開発したソフトウェアの構成を図 4.2 に示す。ソフトウェアの構成は機能的に、汎用的な処理を行う部分とコンテンツに特化した処理を行う部分に大別される。開発環

表 4.1: ブラウザ用 PC とその想定性能

PC	CPU 500MHz、メモリ 256MB 以上の DOS/V 機
仮想世界提示デバイス	1280x1024 ドット以上の解像度で表示できるディスプレイ
音声入出力デバイス	マイクとスピーカ
入力デバイス	ジョイスティックとキーボード、マウス
ネットワーク設備	100BASE-TX 対応

境には Microsoft 社の Visual C++^[33] を用いた。通信は独自のプロトコルを定義して TCP/IP で実装した。

汎用的な処理を行う部分は昨年度のシステム LASNET をベースに開発した。また、昨年度ロケールサーバで行っていたコンテンツの処理は、本システムでは外部モジュールが行う。汎用的な処理を行う部分は「仮想環境を統括するソフトウェア」と「仮想環境の基本機能を提供するソフトウェア」に分類できる。以下にそれらの内容を述べる。

仮想環境を統括するソフトウェア まず仮想環境を統括するソフトウェアには、ワールドサーバ、ワールド情報データベースがある。複数のロケールが存在する場合にも、全てのロケールに関連する情報をデータベースに記録することにより、情報を共有することが可能である。以下にそれらをまとめる。

2 ワールドサーバ

ネットワークを介して全てのロケールサーバを集中管理するサーバ。ロケールサーバあるいはワールド情報データベースからの要求があれば、他のロケールサーバやワールド情報データベースから情報を取得し、その情報を返す。システム内に1つだけ存在する。

2 ワールド情報データベース

全てのロケールに共通する情報を保管するデータベース。ワールドサーバからの要求があれば、データベースが保管しているデータを返す。複数の種類のワールド情報データベースを作成し、ワールドサーバに登録することができる。

仮想環境の基本機能を提供するソフトウェア 仮想環境の基本機能を提供するソフトウェアには、ロケールサーバ、Splineサーバがある。以下にそれらをまとめる。

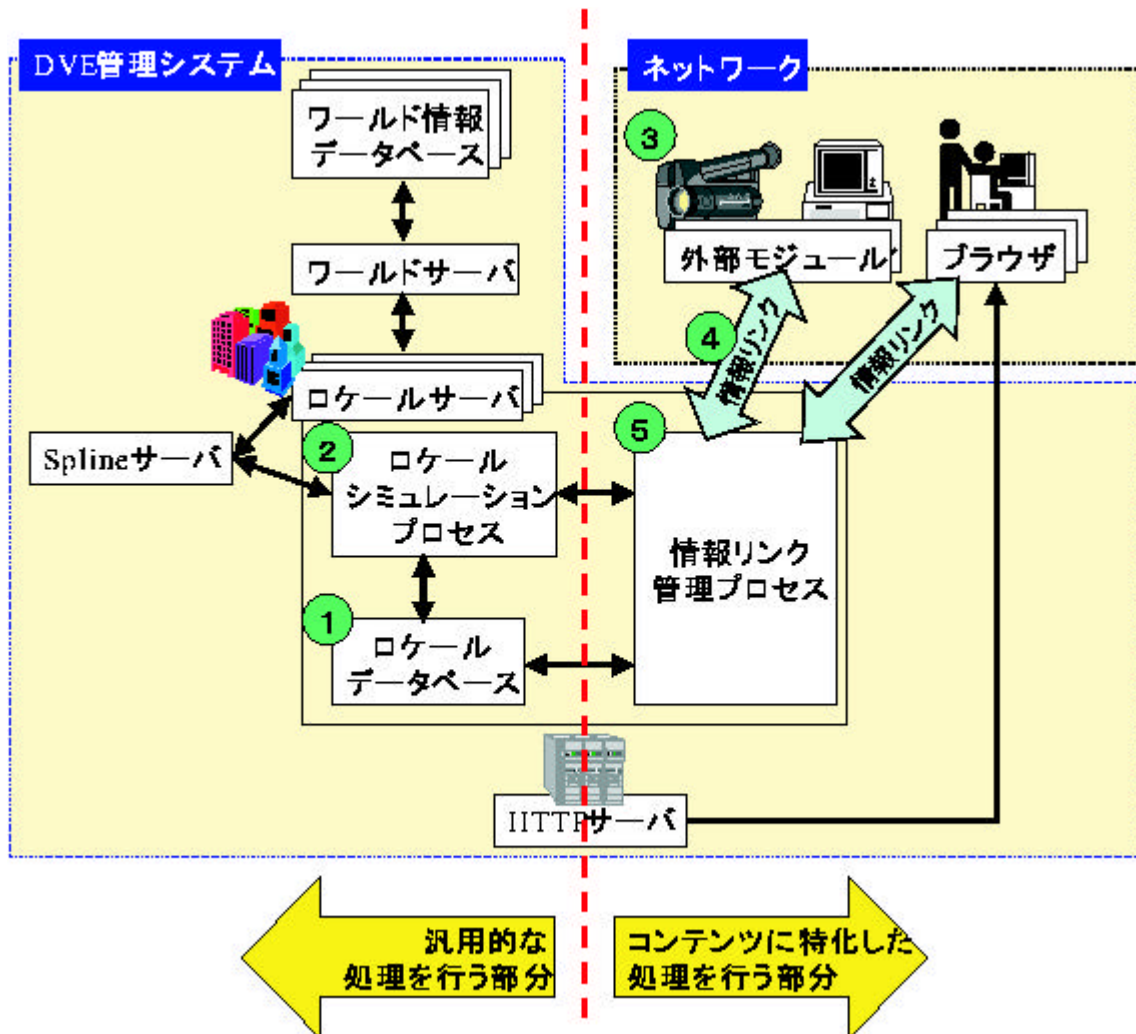


図 4.2: ソフトウェア構成

² ロケールサーバ

ロケールの構築およびシミュレーションを行うサーバ。ブラウザをロケールサーバに接続すれば、仮想世界に参加できるようになる。ロケールの数だけシステム上に存在する。

² Spline サーバ

リアルタイム通信により 3 次元画像を共有するライブラリ Spline3.0 Beta-3 を利用するためのサーバである。Spline の独自プロトコルを用いて、ロケールサーバとブラウザと接続している。システム内に 1 つだけ存在する。

² HTTP サーバ

仮想世界の各種設定ファイルが格納されている。いずれも必要に応じてダウンロードできる。Microsoft 社の Internet Information Server 3.0 を用いたが、他のソフトウェアも利用できる。

なお、これらのソフトウェアはいずれも DVE の管理上必要であるため、システムを提供する側が準備すべきである。このようなソフトウェアを、情報資源やブラウザといったネットワーク上のソフトウェアと区別する意味で、DVE 管理システムと呼ぶ。

以上のソフトウェア構成は昨年度の LASNET と同様であるが、MALIONET ではネットワーク上の情報資源を取り込むためにロケールサーバを大幅に改良した。ここではその機能を説明するため、ロケールサーバのモジュールについて説明しておく。説明には、図 4.2 中の番号と一致する番号を付加した。

- (1) ロケールデータベース 3.3.1 項で述べた仮想オブジェクトの管理機能を担当し、すべての仮想オブジェクトの形状、姿勢等のデータを格納するデータベースである。MALIONET では、外部モジュールと送受信したデータを管理できるように改良した。
- (2) ロケールシミュレーションプロセス 3.3.1 項で述べた仮想世界シミュレーション機能を担当し、ロケールデータベースのデータを目に見える仮想世界に反映させるために Spline サーバに情報を送信するプロセスである。ロケールデータベースの改良に併せて一部を改良した。

コンテンツに特化した処理を行う部分 コンテンツに特化した処理を行う部分は、仮想世界のコンテンツを提供するソフトウェアと仮想世界に参加するソフトウェアに大別できる。

仮想世界のコンテンツを提供するソフトウェア 仮想世界のコンテンツを提供するソフトウェアは、ネットワーク上の外部モジュールと DVE 管理システムのロケールサーバ、それらを結合させる情報リンクである。ロケールサーバに関するモジュールの機能について概略をまとめる。

- (3) 外部モジュール ネットワーク上に存在する情報資源で、データを処理し、送受信する機能をもつ。情報リンクを用いてロケールサーバにデータを送信し、仮想オブジェクトに反映させることができる。複数の外部モジュールを接続できる。
- (4) 情報リンク DVE と外部モジュールを接続し、データのやりとりを行う通信路のことである。外部モジュールの数だけ存在する。
- (5) 情報リンク管理プロセス 3.3.1 項で述べたデータの送受信機能と情報リンクの管理機能を担当するロケールサーバのモジュールである。データを送受信した後、情報リンクの作成、削除や、因果関係の管理など、外部モジュールを用いるための処理全般を行う。LSANET には存在しない。

仮想世界に参加するソフトウェア MALIONET によって提供される仮想世界には、専用のソフトウェアを用いて参加する。

ブラウザ Web ページを閲覧するブラウザと同じような、仮想世界に参加するためのソフトウェアである。ブラウザの機能は以下のとおりである (図 4.3)。

- 1. 仮想世界を 3 次元画像で提示する機能。
- 2. 利用者のアバタを操作する機能。
- 3. 他の利用者とコミュニケーションする機能。

ブラウザは利用者の数だけ存在する。

4.1.3 因果関係の管理を行うモジュールの詳細

本節では、3.4.2 項で提案した因果関係の管理手法の実装方法について述べる。以下では、外部モジュールと情報リンク管理プロセスの構成と動作を詳細に説明する。

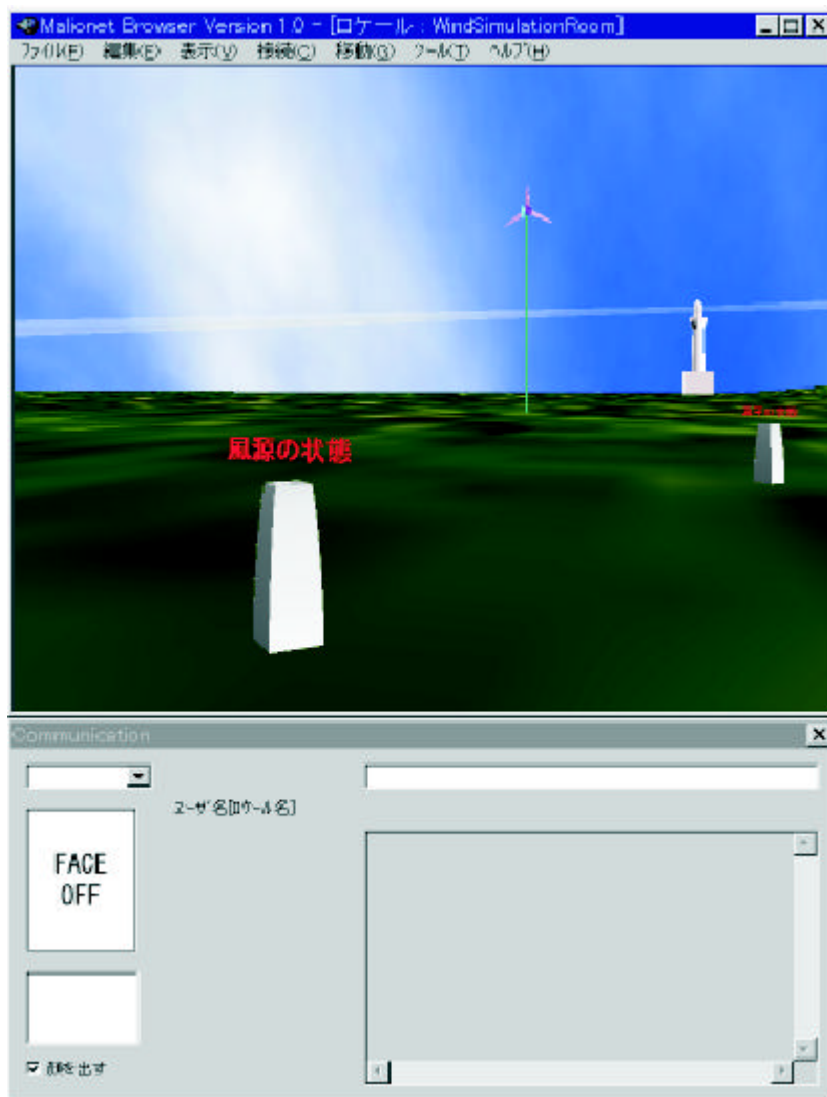


図 4.3: ブラウザの画面例

外部モジュール 図4.4に外部モジュールの詳細図を示す。外部モジュールはMALIONETシステムの「専用ライブラリ」がネットワーク上に存在する情報資源に組み込まれた構成をとっている。図中の専用ライブラリとは、情報資源が外部モジュールとしてDVEに組み込む際に必要な機能(関数)を提供するものである。外部モジュールを作成する際には、既存の情報資源にこの専用ライブラリを組み込み、プログラムに数行記述して提供する関数を組み込むだけでよい。

専用ライブラリの機能は以下の通りである。データの送受信に加え、仮想世界のシミュレーションで正しい因果関係を保つための機能を有する。

機能1 情報リンクに対してデータを入出力する。

機能2 入力するデータをタイムスタンプごと履歴に記録する。

機能3 受信データより処理する時刻を決定する。

機能4 処理する時刻に対するデータを履歴より参照し、情報資源に渡す。

機能5 一定時間経過すると次に処理すべき時刻を計算し、機能3と同様に情報資源にデータを渡す。

機能6 処理されたデータにタイムスタンプと実確認値を付加する。

情報リンク管理プロセス 情報リンク管理プロセスは、その処理内容により3つの部と2つのデータベースから構成される。図4.5に情報リンク管理プロセスの構成を示す。

² 入出力部：情報リンクに対してデータを入出力し、外部モジュールとの送受信を行う。送信の際には情報リンクデータベースより送信先の外部モジュール情報を取得し、履歴データベースより過去の送信記録を取得し、それらに基づいて確認値を予測し履歴データベースに格納する。受信時にはデータの種類により、メッセージ管理部または情報リンク管理部にデータを渡す。

² メッセージ管理部：受信したデータに対する予測確認値を履歴データベースより取得し、受信したデータを評価する。確実なデータは入出力部とロケールデータベースに渡し、不確実なデータは履歴データベースで保留する。保留されたデータはここで呼び出され、確実なデータと同じく処理する。

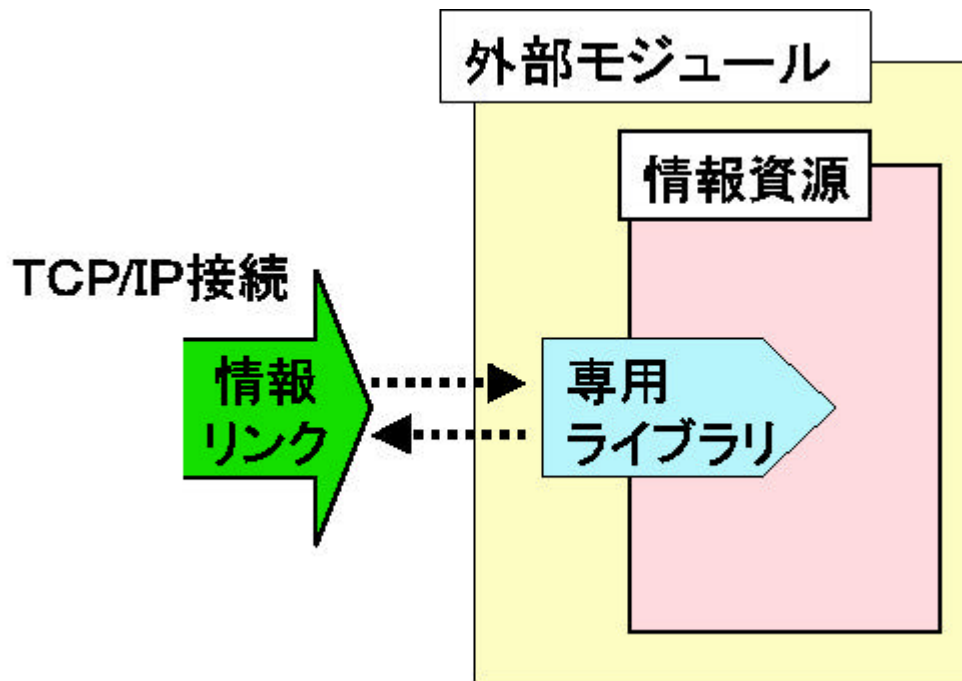


図 4.4: 外部モジュールの構成

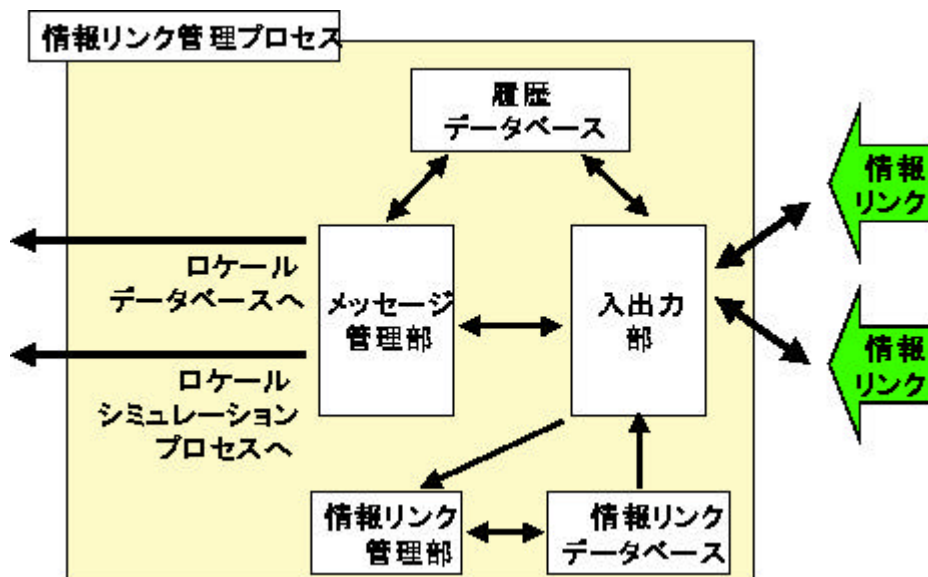


図 4.5: 情報リンク管理プロセスの構成

- 2 情報リンク管理部：情報リンクの生成、削除、変更を行う。情報リンクの変更に関する情報は、情報リンクデータベースに格納する。
- 2 情報リンクデータベース：現在作成されている情報リンクや作成予定の情報リンク等情報リンクに関する全ての情報を格納する。
- 2 履歴データベース：過去に送信したデータとそのタイムスタンプ、予測確認値、保留された不確実なデータを格納する。

4.1.4 使用する送信パケット

本システムでは、各サーバ、モジュール間で通信を行いながら、仮想世界をシミュレートしている。この通信に使用されるパケットは全ての通信で共通であり、図 4.6 に示すように、送信する情報の実体のほかに送信先のサーバの種類やその名前、送信する情報の種類などの制御情報が付加されている。

情報リンクに用いられる送信パケットでは、その情報の実体の中に因果関係の管理用の情報とデータの名前、データの値が含まれている。因果関係の管理用の情報は、ロケールサーバから送信するものと外部モジュールから送信するものでは異なり、ロケールサーバから送信するものはタイムスタンプ、外部モジュールから送信されるものはタイムスタンプと実確認値からなる。

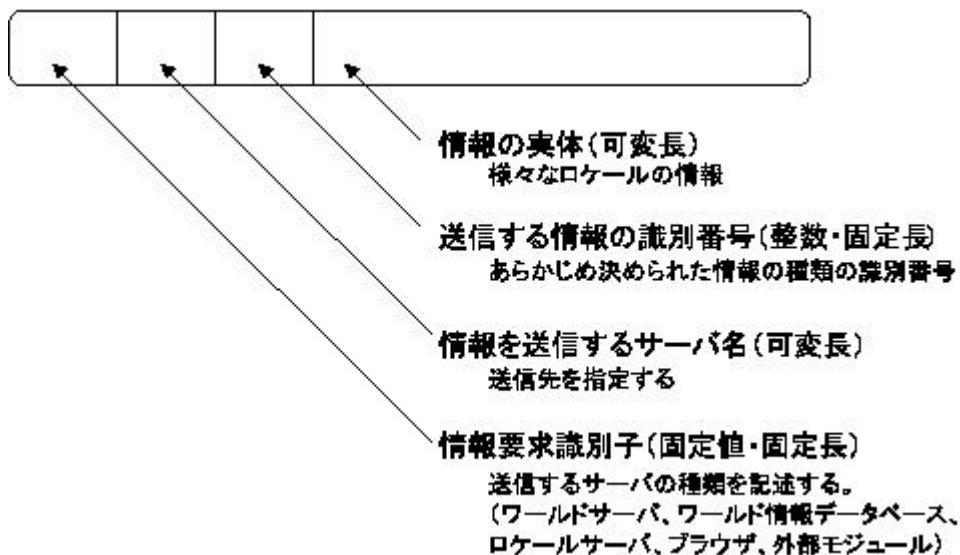


図 4.6: 送信するパケット

4.2 設定ファイルの仕様

本システムで仮想世界を作成するためには、以下に示す 4 種類の設定ファイルを作成する必要である。

ロケールファイル 仮想空間を作成する上で不可欠な設定ファイル。ワールドサーバの IP などシステムに必要な情報や、仮想空間の名前や大きさ、仮想オブジェクトなどコンテンツの情報を記述する。このファイルの中で以下の設定ファイルが引用される。図 4.7 にロケールファイルの例を示す。

オブジェクトファイル 仮想オブジェクトがもつデータを定義する。形状や姿勢などの描画に関するデータや、風速などのその他のデータに分けて記述する。図 4.8 にオブジェクトファイルの例を示す。

情報リンクファイル 情報リンクに関する設定を記述する。外部モジュールのデータと仮想オブジェクトのデータとの対応関係を記述する。利用する外部モジュールと仮想オブジェクトはこのファイルをロケールファイルで引用するとき決定する。図 4.9 は情報リンクファイルの例である。

外部モジュールファイル 外部モジュールが仮想世界に接続する上で必要不可欠な情報を記述するファイル。ワールドサーバの IP や外部モジュールの登録名などを記述する。図 4.10 に設定ファイルの例を示す。

4.3 仮想世界の作成と動作

本節では、実際に仮想世界を作成する手順について述べる。まず、仮想世界を作成する準備について説明し、次に DVE システムを起動し仮想世界を作成する手順について説明する。そして仮想世界のシミュレーションについて述べた後、最後にその世界を体験する方法について説明する。

コンテンツの準備 MALIONET を利用して仮想世界を作成する作成者は、まずコンテンツの準備を行う。具体的には、以下のようにして形状ファイルと外部モジュールを作成すればよい。

ロケールファイル

```
<Locale>
  Name           : ロケールの名前
  WorldServerIP  : ワールドサーバのIP
  WorldServerPortNo : ワールドサーバのポート
  SplineServerIP : SplineサーバのIP
  HttpPath       : HTTPサーバのパス
  ExBoundary     : ロケールの範囲
  LocalePortNo   : ロケールのポート
</Locale>

<Object>
  ObjectName     : 仮想オブジェクトの名前
  ObjectFile     : オブジェクトファイル名
  <ILFile>
    外部モジュール名 : 情報リンクファイル名
    .....
  </ILFile>
  Initialize     : 初期化
    データ : その値
    .....
</Object>
.....
```

図 4.7: ロケールファイル

オブジェクトファイル

```
<Object>
  <spData> ← 描画に関するデータ
    VisualDefinition : 形状
    spTransform      : 姿勢
  </spData>
  <Data> ← その他のデータ
    データの型 : データの名前 : データの値
  </Data>
</Object>
```

図 4.8: オブジェクトファイル

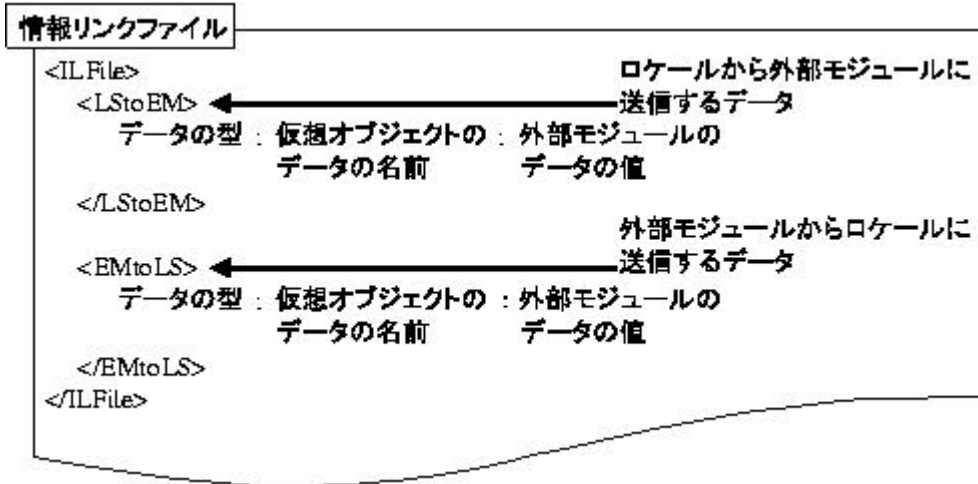


図 4.9: 情報リンクファイル

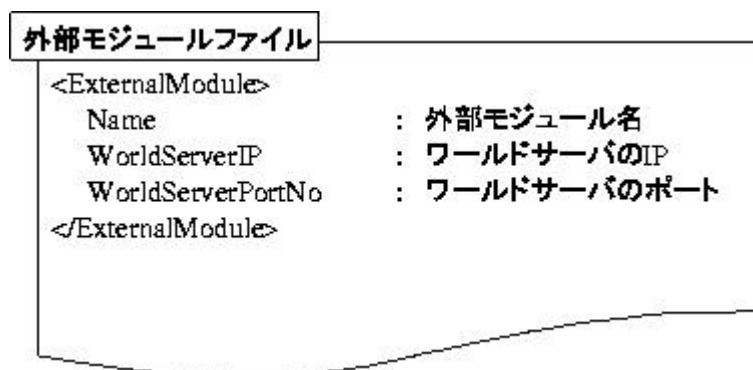


図 4.10: 外部モジュールファイル

- 2 形状ファイル：VRML2.0形式で記述された仮想オブジェクトの形状ファイルを準備する。テキストエディタで直接編集する、あるいは3Dモデリングソフトウェアを使用する等の方法で形状ファイルを作成すればよい。またVRMLはインターネット上で汎用的に利用されているファイル形式であるため、適切なファイルを検索し、ダウンロードして利用することもできる。
- 2 外部モジュール：ネットワーク上に公開されている情報資源をインターネット等で検索または新規作成し、4.1.3項で述べた手法で外部モジュールを作成する。

設定ファイルの記述 次に、仮想世界を作成するために必要な設定ファイルを記述する。記述すべき設定ファイルは、ロケールファイルとその中で引用されているオブジェクトファイル、情報リンクファイル、また必要な外部モジュールの外部モジュールファイルである。

ホームページ上の風の情報を利用して仮想世界の風車の羽根を回転させるという仮想世界(図4.11(a))を例に、必要な設定ファイルについて説明する。この仮想世界を作成する際に必要なロケールファイルと、風オブジェクトに関するオブジェクトファイル、情報リンクファイルを図4.11(b)に示す。

図4.11(b)のロケールファイルには、仮想世界内に作成する風オブジェクトと風車オブジェクトを分けて記述してある。このうち風オブジェクトの記述(図(b)の1)を見ると、その中には仮想オブジェクトの名前(図(b)の2)、オブジェクトファイル名(図(b)の3)、情報リンク(図(b)の4、5)が記述してある。また情報リンクについては外部モジュール名と情報リンクファイル名が記述されている。

仮想世界を作成する際、以下の処理により風オブジェクトに関する仮想オブジェクト、情報リンクが生成される。(図中の番号に対応)

- i. 2の記述により風オブジェクトが生成される。
- ii. 3の記述によりWindという名前のオブジェクトファイルが引用される。それに基づいて風オブジェクトに風速データを追加する。
- iii. 4の記述によりGetWindという名前の情報リンクファイルが引用される。それに基づいて風オブジェクトの風速データにホームページの風速データを反映させるように情報リンクを生成する。

iv. 5の記述により SendWind という名前の情報リンクファイルが引用される。それに基づいて風車シミュレータの風速データに風オブジェクトの風速データを反映させるように情報リンクを生成する。

このように、ロケールファイルで1行記述するごとに仮想世界のコンテンツは豊かになっていく。

システムの起動 コンテンツの準備と設定ファイルの記述が済めば、MALIONET システムを起動することができる。

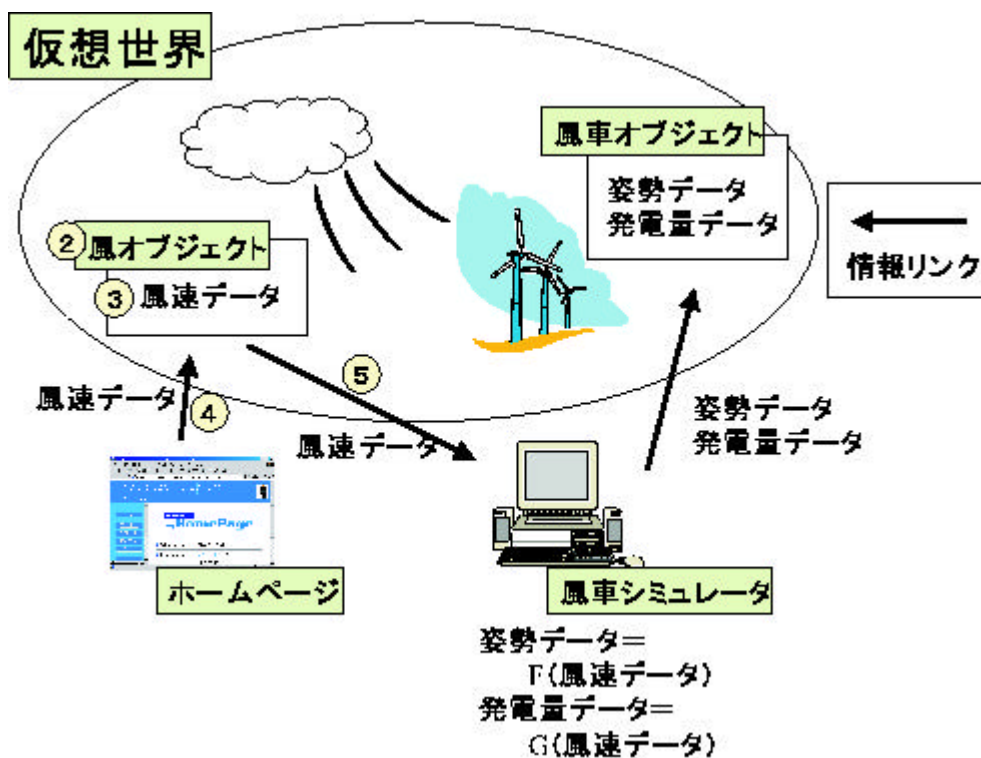
- i. ワールドサーバ、ワールド情報データベースを起動する。
- ii. 仮想世界の作成者が自ら用意した外部モジュールがあれば、記述した外部モジュールファイルを用いて外部モジュールを起動する。起動後はワールドサーバにより管理される。
- iii. ロケールサーバを起動する。このとき、ロケールサーバ自身をワールドサーバに登録し、また現在存在している外部モジュールの情報をワールド情報データベースより受信する。
- iv. ロケール作成時にロケールファイルが読み込まれ、前述したように仮想世界が作成される。このとき、受信した外部モジュールの情報と情報リンクファイルの外部モジュール名を比較し、既に存在している外部モジュールについては情報リンクを生成する。

以上の処理により仮想世界は作成される。

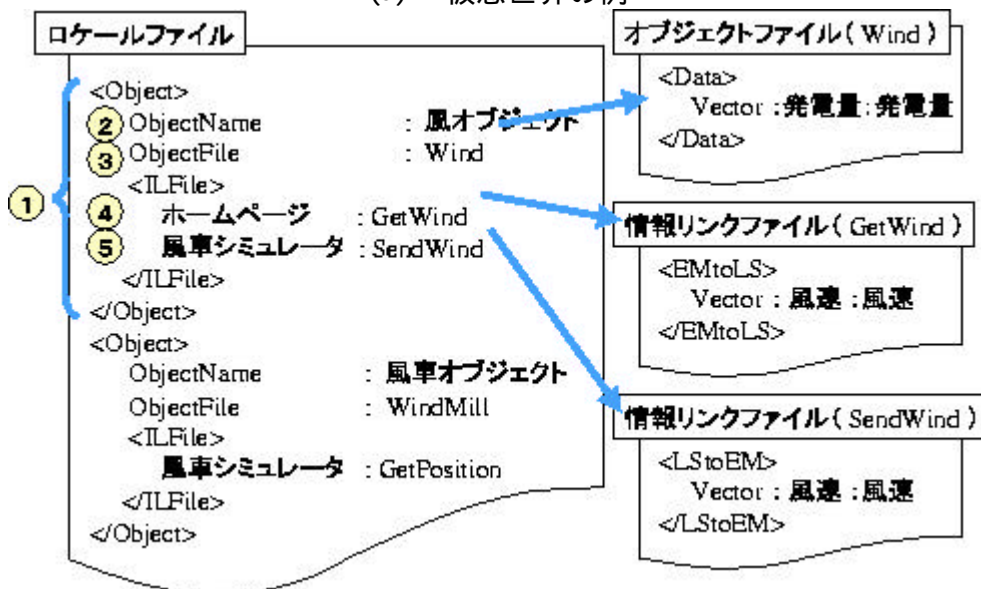
仮想世界のシミュレーション 以上のようにして作成された仮想世界では、以下の手順で因果関係の管理が行われ、フレームレートを落とすことなく、かつロールバックの少ないシミュレーションが行われる。

外部モジュールは、3.5節で述べた手順で処理を行い、ロケールサーバに情報を送信する。以下では4.1.3項で説明したロケールサーバの情報リンク管理プロセスの因果関係の管理手順について述べる。

- i. 情報リンクから送られてきたデータは入出力部に送られる。



(a) 仮想世界の例



(b) 設定ファイル

図 4.11: 仮想世界作成の例

- ii. 入出力部にて、iで送られてきたデータが外部モジュールからのデータの場合メッセージ管理部へ、情報リンクに関する情報の場合は、情報リンク管理部へ送る。
- iii. メッセージ管理部では、実確認値と履歴データベースに格納してある予測確認値を比較し、データを処理する。データを仮想世界に反映する際には、ロケールデータベースと入出力部へデータを送る。
- iv. 入出力部では、メッセージ管理部から受け取ったデータにタイムスタンプを付加し、関連する外部モジュールに送信する。

以上の処理により、因果関係の管理を行い、外部モジュールから入力されたデータが仮想世界内のオブジェクトの位置や姿勢等に反映される。

仮想世界への参加 利用者はブラウザを用いて仮想世界を体験することができる。

ブラウザを起動するとまずワールドサーバに接続をする(図 4.12(1))。このとき、利用者の名前を入力する(図 4.12(2))。ワールドサーバでは現在存在しているロケールの情報をブラウザに送信し、利用者に提示する(図 4.12(3))。利用者はその中から好きなロケールを選択し、仮想世界を体験することができる(図 4.12(4))。

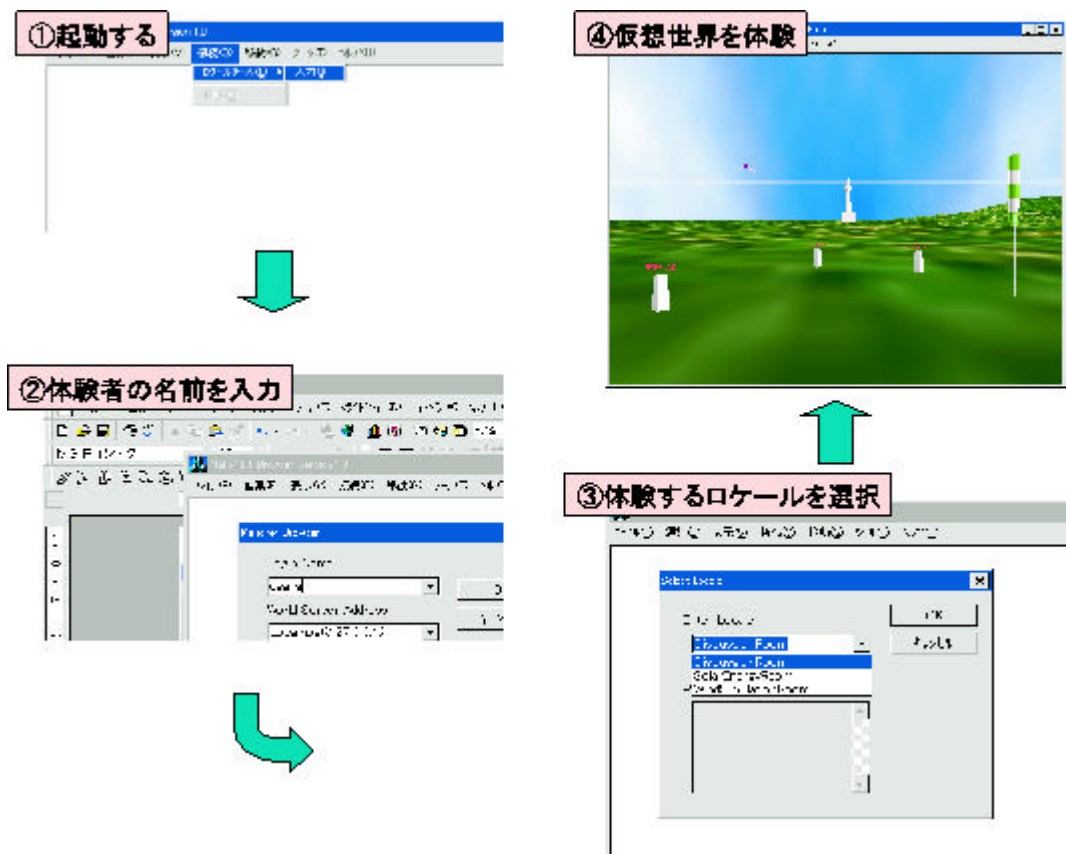


図 4.12: 仮想世界への参加

第 5 章 動作確認実験

本章では、ネットワーク資源活用型 DVE システム MALIONET の動作確認実験を行った結果について述べる。

まずコンテンツの作成例として、エネルギー教育が行える仮想世界を MALIONET を用いて作成し、多数の外部モジュールが接続されても因果関係を管理できることを確認する。次に、性質の異なる複数の外部モジュールを接続してパフォーマンスを調べ、仮想世界のシミュレーションをリアルタイムで処理可能なことを確認する。

5.1 システムの使用例

本節では、MALIONET を用いて作成できる仮想世界の例として、風力発電の体験教育環境 EEWING(Experiential learning Environment for WIND Generator) を作成した。

5.1.1 確認項目

本実験では以下の項目を確認する。

- 2 ネットワーク上の情報資源の活用：ホームページ上の情報等を外部モジュールとして組み込めることを確認する。
- 2 因果関係の管理：多数の外部モジュールを接続し、因果関係の管理が正しく行われていることを確認する。

5.1.2 動作確認時のシステム構成

作成したコンテンツ EEWING 内では、ホームページや計測器等の外部モジュールの情報を利用し、仮想の世界に風を吹かせ、発電した電力により灯台を点灯している。仮想世界を体験するユーザは風力、風車の種類などを変更することができ、それが吹き流しや灯台の光に反映されることによりその変化を視覚的に体験することができる教育環境である。吹き流しは風速、風向をもとに風になびき、風車は風速、風向をもとに

羽根を回転させ発電し、灯台は風車の発電量をもとに光の強弱が変化する。また、仮想オブジェクトのインタラクションを可能とするために、ブラウザに情報提示ウィンドウを実装した。この仮想世界を作成するために用いた外部モジュールの詳細を付録 A に示す。また、情報提示ウィンドウの詳細を付録 B に示す。

このような環境を提供することにより、利用者は仮想世界の風の強さを自由に变化させ、また風車の種類を替えることにより、どの風車でどのくらいの風のとときにどれだけの発電が行われるかを体験することができる。

ハードウェア構成 サーバ用の PC を 2 台、ブラウザ用の PC を 3 台を研究室内 LAN 上で接続して利用した。各 PC は、4.1.1 項で述べた性能を有する。

ソフトウェア構成 4.1.2 項で述べた各ソフトウェアを、上記のハードウェアを利用して図 5.1 のように構築した。EEWING のために作成した外部モジュールは、その入出力の関係により 3 つに分類できる。アバタの移動と接触判定に関する外部モジュールとその関係を図 5.2 に、風力発電のコンテンツに関する外部モジュールとその関係を図 5.3 に、情報提示ウィンドウに関する外部モジュールとその関係を図 5.4 に示す。

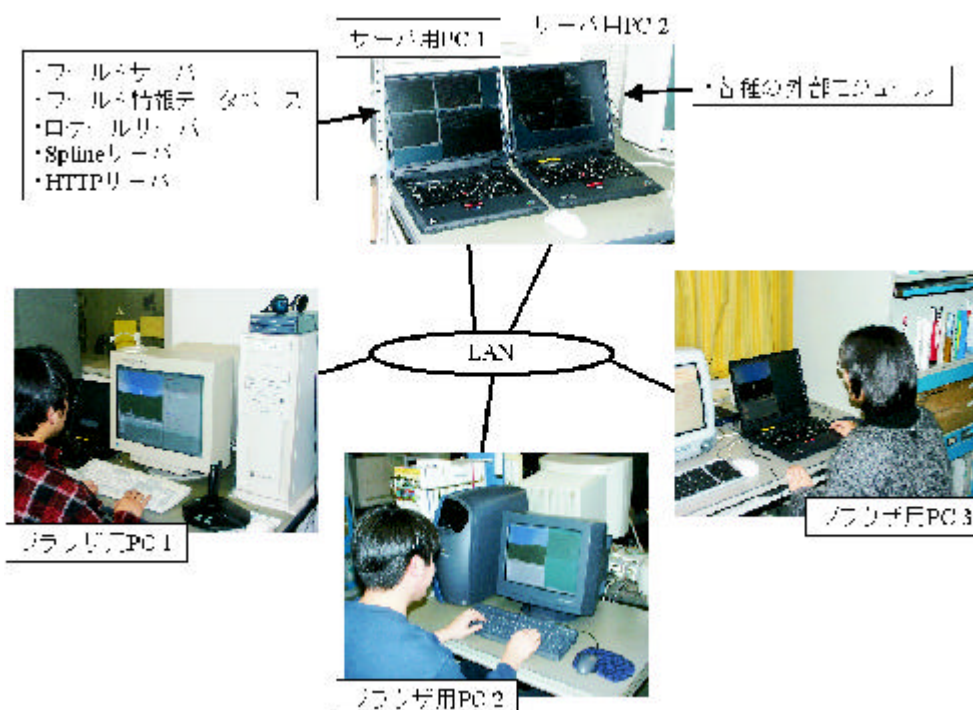


図 5.1: ハードウェアとソフトウェアの動作環境

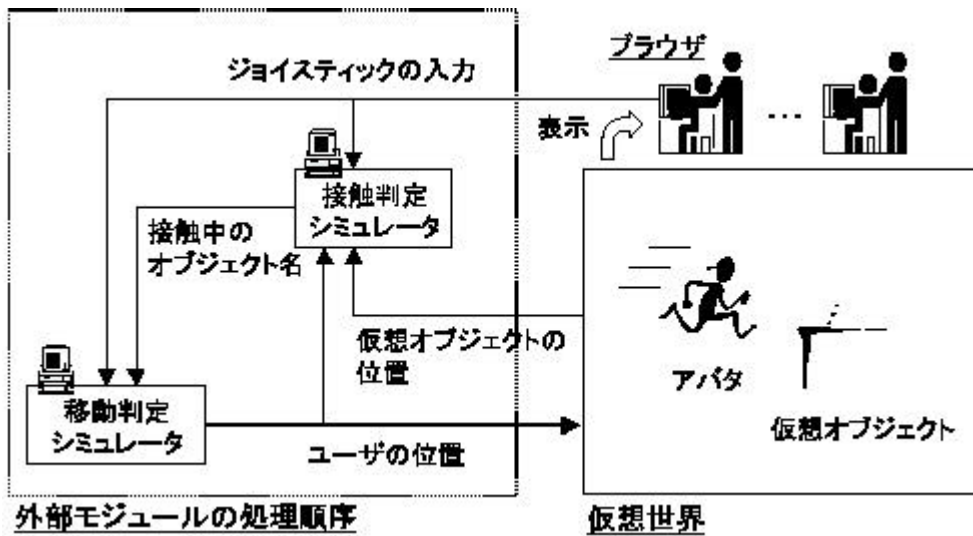


図 5.2: 移動と衝突判定に関する外部モジュールの関係

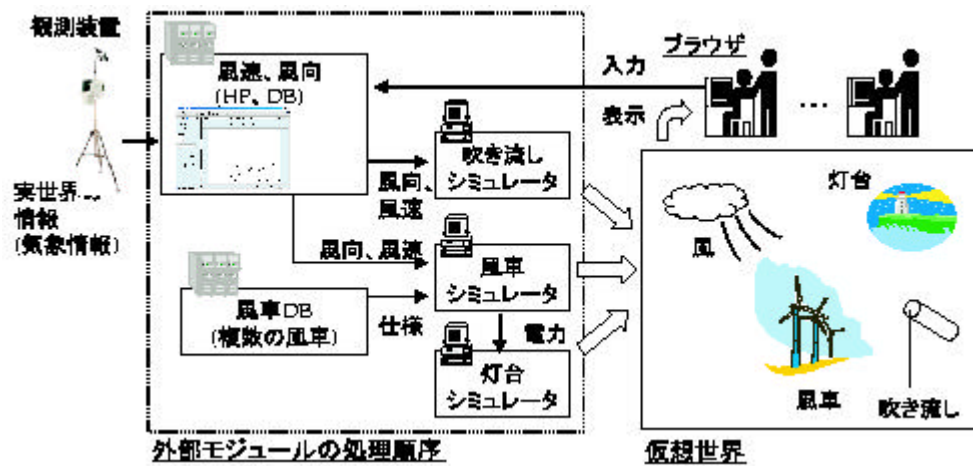


図 5.3: 風力発電のコンテンツに関する外部モジュールの関係

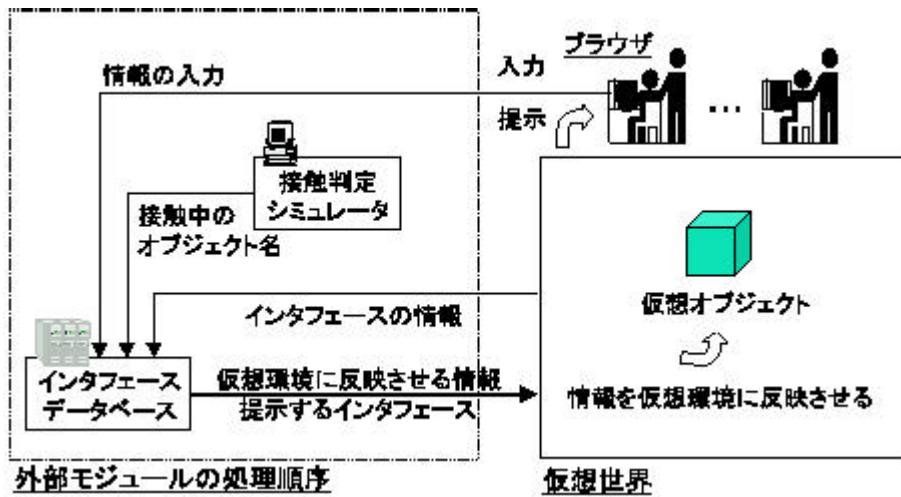


図 5.4: 情報提示ウィンドウに関する外部モジュールの関係

5.1.3 結果

動作状況 図 5.1 の各 PC における動作中の様子を、サーバ用 PC については図 5.5、ブラウザ用 PC については図 5.6 に示す。サーバ用 PC では複数のコンソールアプリケーションが起動している様子を確認できる。またブラウザ用 PC では、仮想の世界が利用者に提示されている。ブラウザには、仮想世界を提示する 3 次元画像提示ウィンドウと、数値の入力や提示を行う情報提示ウィンドウ、チャットに用いるチャットウィンドウの各ウィンドウが提示されている。図 5.7 に、仮想世界の構成を示す。風車、吹き流し、灯台が表示されている。

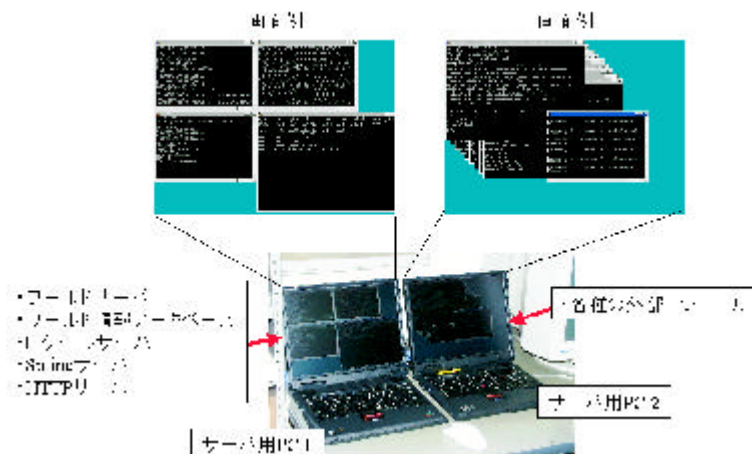


図 5.5: サーバ用 PC の画面例

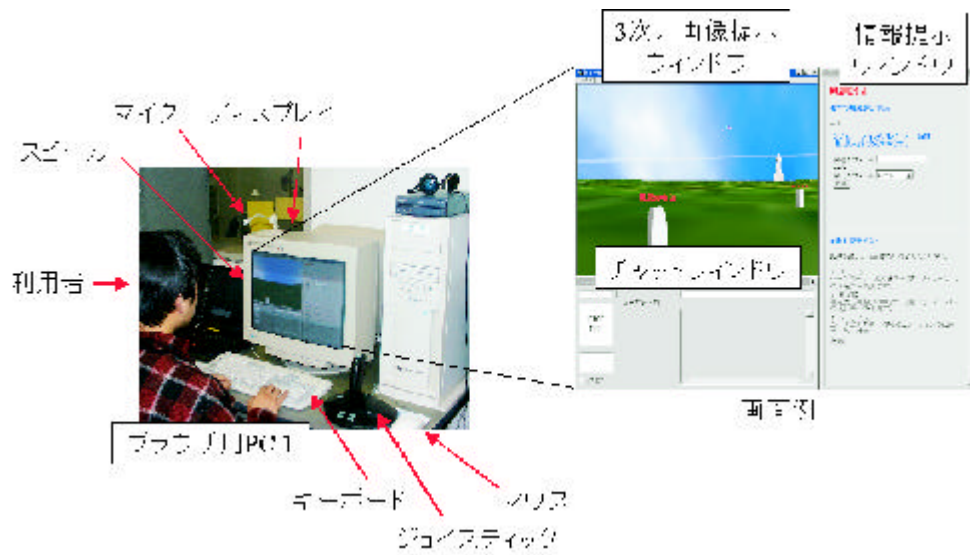


図 5.6: ブラウザ用 PC の画面例

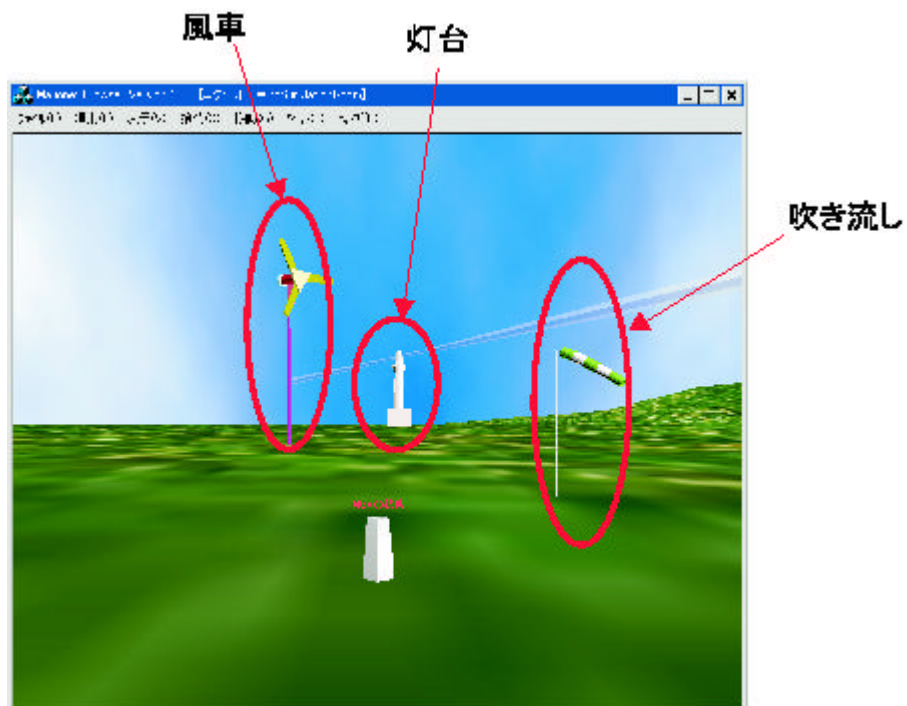


図 5.7: 作成した教育用仮想世界の画面例

ネットワーク上の情報資源の活用 この仮想世界を作成するにあたり、風速と風向のデータをインターネット上のホームページから取得するようにした。図 5.8 は風速・風向が掲載されている HP の例で、EEWING はこれらの HP から情報を取得できる。

外部モジュールの作成は、10 行程度のプログラムを記述し、専用ライブラリを組み込むことにより可能となる。一例として、図 5.9 に示す CGI プログラムを外部モジュールとしてシステムに組み込んだ。

また、外部モジュールの有無により仮想世界で体験できるコンテンツが変化する例を図 5.10 に示す。作成した環境では、衝突処理を行う衝突判定モジュールにより物体に近づいたことを検知し、アバタが仮想オブジェクト内を通過することができない処理を行っている。図中で左右の画面を比較すると、衝突判定モジュールの有無により、仮想オブジェクトに近づける距離が異なっていることがわかる。つまり、外部モジュールを仮想世界に追加することにより、ユーザが体験できるコンテンツがより豊かになっている。

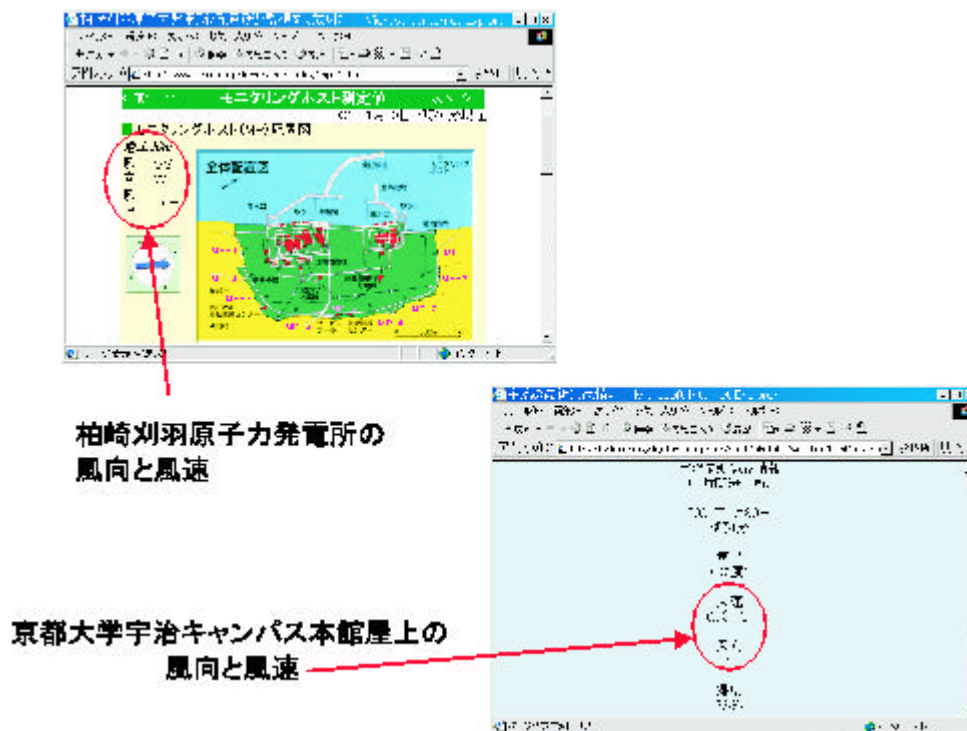


図 5.8: 風速・風向が掲載されている HP の例

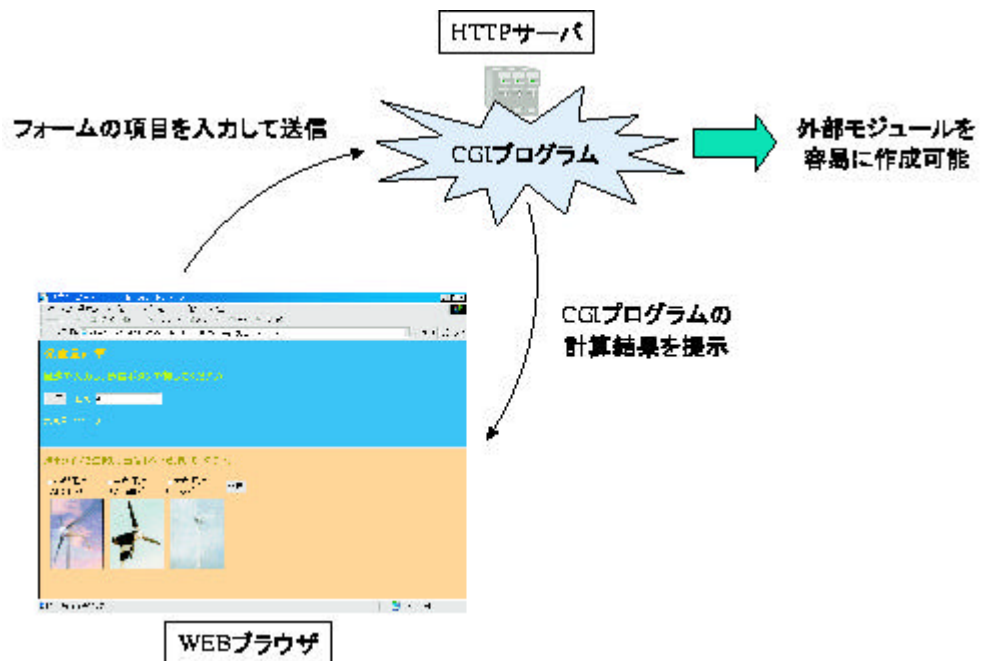


図 5.9: CGI プログラムを利用して外部モジュールを作成する例

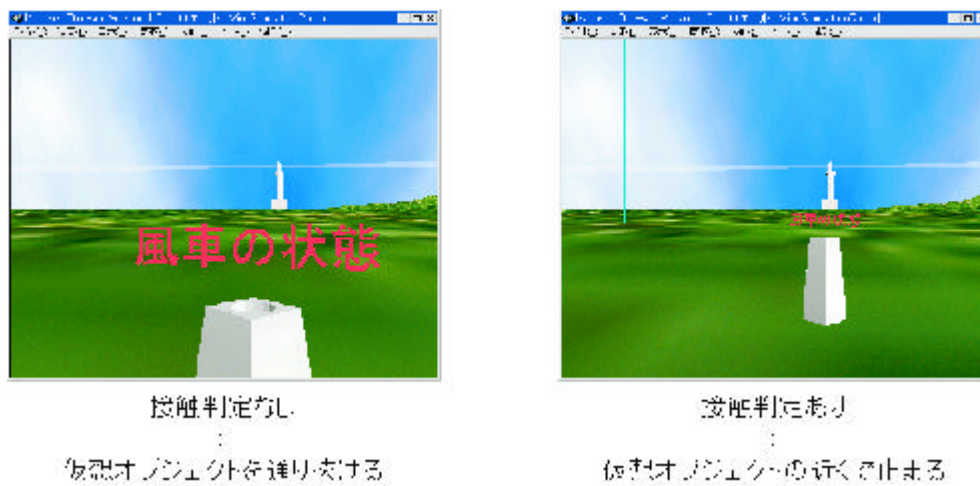


図 5.10: 衝突処理の有無による仮想世界の相違

因果関係の管理 移動と衝突判定、風力発電のコンテンツ、情報提示ウィンドウの3項目を確認した。

移動と衝突判定 因果関係の確認項目と確認結果について表 5.1 に示す。例えば図 5.10 に示した結果と同様に、すべての項目において正しい因果関係で処理していることを確認できた。

表 5.1: 移動と衝突判定の因果関係の確認

確認項目	結果
接触判定シミュレータのみでは移動できない	
移動判定シミュレータのみでは移動のみできる	
接触判定と移動判定シミュレータを利用すると接触判定と移動が可能である	

風力発電のコンテンツ 因果関係の確認項目と確認結果について表 5.2 に示す。すべての項目で正しい因果関係を処理していることを確認できた。例えば図 5.11 では、風速をもとに処理した電力を表示し、その電力で灯台の光を回転させている。

表 5.2: 風力発電のコンテンツの因果関係の確認

確認項目	結果
風速データを取得する	
風速に応じて吹き流しの処理が行われる	
風速に応じて風車の処理が行われる	
風車の発電量に応じて灯台の処理が行われる	

情報提示ウィンドウ 情報提示ウィンドウの確認項目と確認結果について表 5.3 に示す。すべての項目で正しい因果関係で処理していることを確認できた。例えば図 5.12 では、風車の種類を変更する情報提示ウィンドウを提示している。

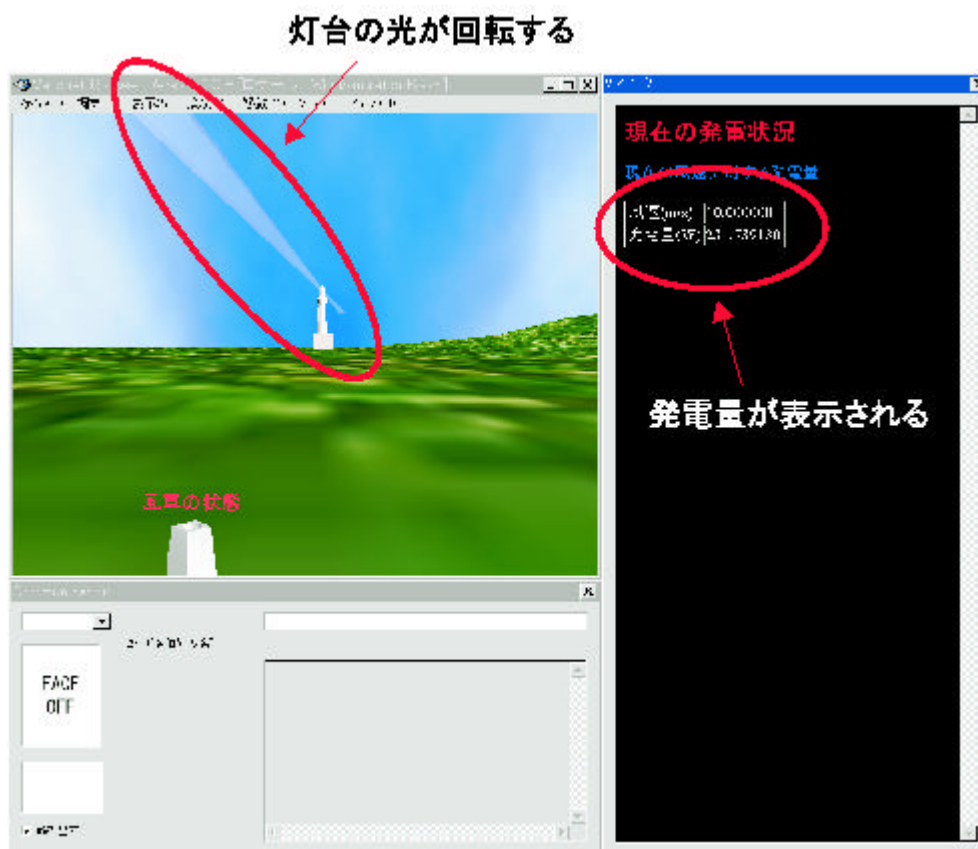


図 5.11: 風力発電と灯台の動作例

表 5.3: 情報提示ウィンドウに関する因果関係の確認

確認項目	結果
オブジェクトに接近すると情報提示ウィンドウに情報が提示される	
オブジェクト毎に異なる情報が提示される	
情報提示ウィンドウから情報を入力し仮想世界に反映できる	
変化した仮想世界の情報が提示される	



図 5.12: 情報提示ウィンドウの動作例

5.1.4 考察

インターネット上のホームページより風速データを取得できることを確認し、ネットワーク上の情報資源を外部モジュールとして活用できることを確認した。また、図 5.2、図 5.3、図 5.4 のように、互いに関連した外部モジュールを用いて、例えば風速を変化させることにより灯台の光が変化するなど、処理が正しく行われていることより、論理的に因果関係の管理が正しく行われていることを確認した。

以上より、仮想世界の作成者が外部モジュールと仮想オブジェクトの関係を記述するだけでネットワーク上の情報資源を DVE に組み込めることが確認できた。

5.2 DVE システムの性能評価

本論文で提案したシステムでは、ロケールサーバが集中的にメッセージ管理を行っている、また容易に仮想世界を作成するために新たな処理を追加したため、システムのパフォーマンス低下が考えられる。本節では、1. 外部モジュールの個数が増加した場合、2. 通信するメッセージ量が増加した場合のロケールサーバの描画数を測定し、本研究で構築した MALIONET が実用的な速度で動作しているかを確認する。

5.2.1 測定項目

システムでのメッセージ処理が集中するロケールサーバのメッセージの処理能力について調べる。以下の 2 つの状況について、ロケールサーバの描画数を測定する。

² 外部モジュールの個数が増加した場合

² ロケールサーバが処理するメッセージの個数が増加した場合

5.2.2 確認方法

図 5.13 に実験に使用する外部モジュールの構成、表 5.4 に外部モジュールの一覧を示す。本実験では、外部モジュール A でデータを変更しメッセージを送信し、そのメッセージが外部モジュール B₁ から外部モジュール B_i ($i = 1, \dots, N$) まで順々に伝わっていく際のロケールサーバのパフォーマンスを測定する。外部モジュール A では処理時間毎にデータを変更しメッセージを送信する。外部モジュール B では、受信したデータを 10ms 後そのまま送信する。

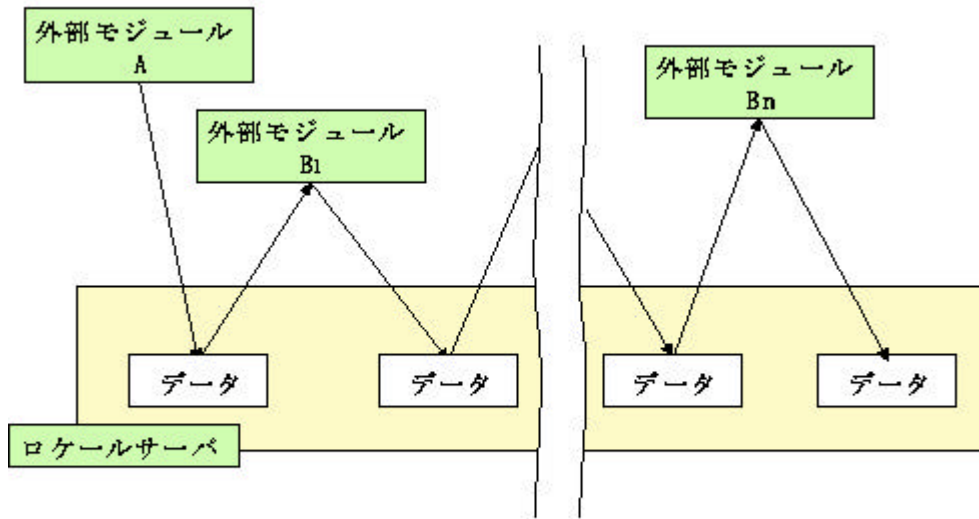


図 5.13: ロケールサーバの性能評価に使用する外部モジュールの構成

表 5.4: ロケールサーバの性能評価に使用する外部モジュール一覧表

外部モジュール	処理時間	待機時間	入力	出力
A	100,200,500, 1000,2000ms	なし	なし	1個(整数値)
B	10ms	1000ms	1個(整数値)	1個(整数値)

具体的には、外部モジュール A の処理時間を 100,200,500,1000,2000ms、外部モジュール B の個数 N を 1,5,10,20 個と変化させ、ロケールサーバの描画数を測定することにより、外部モジュールの個数、ロケールサーバが送受信するメッセージの数がロケールサーバのパフォーマンスに与える影響を測定する。

本実験では、外部モジュールをつながない無負荷の状態でのロケールサーバの更新間隔を 100ms とする。また、測定時間は全てのモジュールを起動し定常状態になってから 1 分間とする。

5.2.3 動作環境

ハードウェア構成 使用する 4 台の PC の性能を表 5.5 に示す。LAN 等その他の構成は 4.1.1 項に準ずる。

表 5.5: パフォーマンス測定に使用する PC の性能

PC	仕様
PC1	CPU 500MHz、メモリ 256MB
PC2	CPU 750MHz、メモリ 256MB
PC3	CPU 750MHz、メモリ 256MB
PC4	CPU 500MHz、メモリ 256MB

ソフトウェア構成 前節 5.2.2 で述べた外部モジュールを使用する。ワールドサーバとワールド情報データベース、Spline サーバ、ロケールサーバは PC1 で動作させる。外部モジュール A は PC2 で動作させ、外部モジュール B は PC3 と 4 に同じ数を動作させる。

5.2.4 結果

測定結果を表 5.6 に示す。処理したメッセージ量とロケールサーバの描画数の関係を図 5.14 に示す。

ロケールサーバの描画数は、メッセージ数が 2500 個付近まではメッセージ数と外部モジュール数の増加に従い、緩やかに低下している。しかし、2500 個付近を超えると急激に低下し、またその低下量は外部モジュールの個数の増加に伴い大きくなっていく。これは、ロケールサーバの処理能力を超えてしまったためだと考えられる。

表 5.6: ロケールサーバのパフォーマンス測定結果

外部モジュール A の 送信間隔 (ms)	外部モジュール B の 個数 (個)	メッセージ量 (個)	ロケールサーバの 描画数 (Frames/second)
100	1	1198	8.92
200	1	900	8.93
500	1	361	9.00
1000	1	180	9.01
2000	1	90	9.19
100	5	2967	7.08
200	5	2658	7.14
500	5	1320	8.24
1000	5	660	8.94
2000	5	330	9.09
100	10	3533	2.45
200	10	3234	2.46
500	10	2385	7.14
1000	10	1260	8.33
2000	10	630	8.37
100	20	3526	1.23
200	20	3233	1.25
500	20	3037	1.30
1000	20	2457	6.83
2000	20	1230	7.81

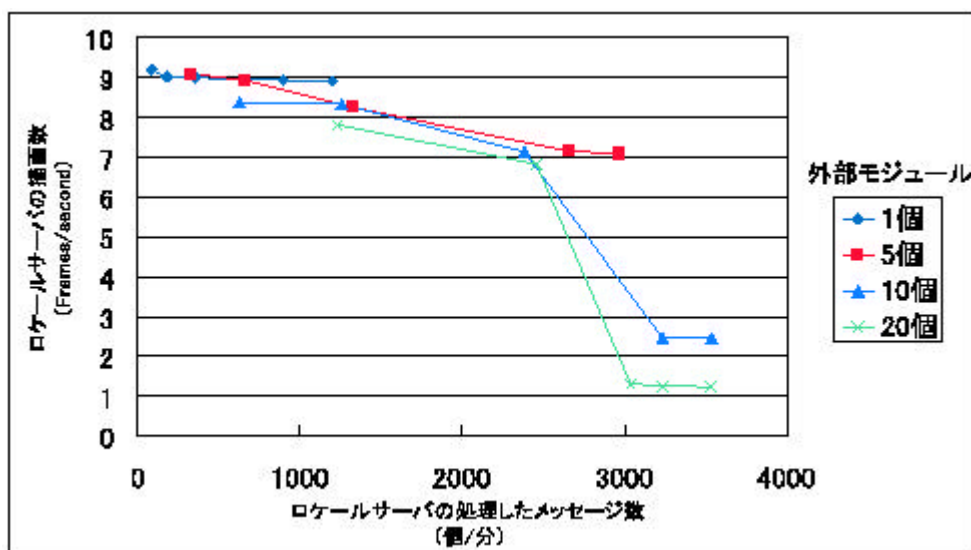


図 5.14: メッセージ数とロケールサーバの描画数の関係

MALIONET では、メッセージを集中管理しているためある閾値を超えると急激に性能が落ち、メッセージを頻繁に送信する外部モジュールを多数組み込むには適していないことが分かった。しかし、閾値以下 (図中 2500 以下) では、性能が多少低下するものの正常に動作していることが確認できた。さらに、本実験では全ての外部モジュールが関連していて、かつ常にデータが変化しているためメッセージ数が増加したが、実際に仮想世界を作成するには本実験よりメッセージ数が減少することが期待できるため、本実験の結果以上の外部モジュールを組み込むことができると考える。

5.3 まとめ

本章では 5.1 節でシステムの使用例を示し、インターネット上の情報資源を活用できること、また作成者が外部モジュールと仮想オブジェクトの関係を記述するだけで、システムが自動的に因果関係の管理を行うことを確認した。また、5.2 節では、システムの性能評価を行い、本システムの限界を示して仮想世界を作成する際、組み込める外部モジュールの個数やロケールサーバで処理可能なメッセージ数等の指標を示した。

また、昨年度開発の NETCOM と比較すると、NETCOM では新しくコンテンツの処理を行う場合には、そのコンテンツを追加するたびにワールド情報データベースに処理内容をプログラミングし、システムを構築しなおす必要があったが、本システムでは専用ライブラリを用いて数行プログラミングするだけで処理が可能である。実際

に、10行程度の専用ライブラリの関数を記述するだけで組み込むことができ、新しくコンテンツの処理を追加する際に作成者の労力が削減されたことを確認した。

今後の課題として、以下のものを挙げる。

処理速度の向上 システムのメッセージ送信頻度や外部モジュール数による限界値の関係を解明し、よりメッセージ送信頻度の高い外部モジュールや多数の外部モジュールを組み込めるようなDVEシステムの構築を行う。

状況に適應する管理手法の発案 本研究で提案したメッセージ管理手法では保留時間を仮に1Frame分の時間としたが、第3章で述べたようにリアルタイム性と現実性の両方を考慮した場合の最適な時間は状況によって変化するので、保留時間をその状況に合わせて変化させる手法を発案する。

複数ロケールへの拡張 本研究では考慮した範囲が1つのロケールであったが、複数のロケールを考慮した場合発生する仮想世界全体でのタイムスタンプの統一などの問題を解決し、複数のロケールに対応したシステムに拡張する。

5.4 今後の展望

本研究では、作成者が容易に様々なコンテンツに対応できるDVEシステムを開発した。

今後、分散型仮想環境が発展を続けていく上では、コンピュータの性能向上や新しい手法の考案、さらにネットワーク環境の整備など技術的課題を解決する必要がある。なお、本システムの一層の改善には、実際にこのシステムを利用し、作成者に具体的な問題点を寄せてもらい、共にアイデアを出し合って解決していくというユーザ参加型の設計改良が効果あると考えている。

第 6 章 結論

本研究では、柔軟かつ容易に様々なコンテンツを提供する仮想世界を作成できる、分散型仮想環境システムを実現するために、ネットワーク上の情報資源を活用するというフレームワークを提案し、その実装と評価を行った。

第 2 章では、分散型仮想環境技術の概要について述べた後、仮想世界を作成する観点から応用研究について展望した。ここでは、仮想世界のコンテンツを作成するためにはプログラミングにより大きな労力が必要である点を問題点と考え、本研究の目的を、仮想世界を容易に作成でき、かつ仮想世界作成の自由度が高い DVE のフレームワークの確立とした。

第 3 章では、この問題点を解決する方法として、現在ネットワーク上に存在している大量の情報を仮想世界のコンテンツとして活用することによって、仮想世界を容易かつ自由に作成できるネットワーク情報資源活用型分散型仮想環境のフレームワークを提案した。なお、ネットワーク情報資源には、ホームページ等に掲載されている情報やインターネットで公開されているシミュレータを念頭においた。このフレームワークでは、1. 仮想世界の外観を作成する、2. ネットワーク情報資源を用意する、3. 仮想世界の外観とネットワーク情報資源を結合して仮想世界をシミュレーションできるようにする、という手順で仮想世界を作成する。しかしそのためには、従来はプログラミングで記述していた処理の手順等を、システムの側で自動的に管理しなければならない。本研究では、並列分散事象シミュレーションの分野で研究が進められている楽観的手法を参考にして、このような問題を解決できる因果関係の管理手法を考案した。この際、仮想世界のシミュレーションをリアルタイムに処理できることに重点を置いた。

第 4 章では、考案した手法に基づき、システムの構築を行った。このシステムでは、仮想世界の外観やネットワーク情報資源を用意して、これらを結合するための設定ファイルを記述しさえすれば、仮想世界を容易に作成可能なものとした。

第 5 章では、構築したシステムを用いて動作確認を行った。まず、システムの利用例としてエネルギー教育用仮想世界を作成し、1. ネットワーク上の様々な情報資源を仮想世界に組み込むこと、2. 考案した手法により因果関係の整合を保ったまま仮想世界をシミュレーションできることを確認した。次に性質の異なる情報資源を結合し、

通信処理等のシステムのパフォーマンスを測定することによって、このようなシステムを様々な分野で利用する際の仮想環境システムの通信頻度や情報資源の性質等の限界値を指標として示した。

本研究で提案したフレームワークにより、仮想世界の作成者の立場に立ち、容易かつ自由にコンテンツ豊かな仮想世界を作成する機会を提供するシステムを実現できる。今後は、様々な作成者にこのシステムを使用してもらい、その意見を反映して、システム改良にフィードバックすることによって、本システムの性能が一層向上することを期待している。

謝 辞

本研究を進めるにあたり、その研究の方向性から論文の執筆についてまで、常に的確なご指導を与えて下さいました吉川榮和教授に深く感謝いたします。

本研究を直接ご指導頂き、数々の貴重な助言を頂きました下田宏助教授に深く感謝いたします。

意見を出し合い、DVEに関する研究をともに行い研究の要所要所で力を貸して頂いた石井裕剛助手に深く感謝いたします。

共に研究を行い様々な苦勞をおかけした博士課程の山本倫也氏に深く感謝いたします。

本システムを構築する上で、欠かすことのできない力となってくれた、学部4回生の小林隆君に心から感謝いたします。

昨年度、現システムの基部構築に精力を傾けていただいた、岩田大司氏、修士1回生の松崎剛士君に感謝します。

最後の最後まで研究室で共に生活し、苦勞を分かち合った修士2回生の皆様に心から感謝いたします。

貴重な助言を常に与えてくれた、博士課程の大林史明氏、小澤尚久氏に大いに感謝いたします。

心に安らぎを与えてくださった、修士1回生の皆様に感謝いたします。

2年間にわたる研究室での充実した毎日を共に過ごした吉川研究室および宮沢研究室の学生の皆様に深く感謝いたします。

研究を進める上で何かとお世話頂いた谷友美秘書、藤岡美紀秘書および吉川万里子秘書に心から感謝いたします。

最後に、分散型仮想環境構築の基盤ソフトウェアを無償で提供して頂いた、三菱電機(株)情報技術総合研究所の前田慎司氏をはじめ、皆様に感謝いたします。

参考文献

- [1] Martin R. Stytz : Distributed Virtual Environments, IEEE Computer Graphics and Application, Vol. 16, No. 3, pp. 19-31 (1996).
- [2] 広瀬 通孝 : バーチャルリアリティ, 産業図書 (1993).
- [3] 鶴沼 宗利, 野中 士郎, 宇佐美 芳明, 好永 俊昭 : 発電プラントの設計データを用いたバーチャル環境実現のための検討, 電気学会論文誌, Vol. 117-D, No. 2, pp. 231-237 (1997).
- [4] Chris Webb, Randy Garrett: Virtual Biology : Simulating the Cell, Proceedings of the Virtual Worlds and Simulation Conference (VWSIM '99), pp. 27-32 (1999).
- [5] 広瀬 通孝 : 人工現実感の教育応用, 計測と制御, Vol. 31, No. 12, pp. 1228-1223 (1992).
- [6] 鳥脇 純一郎 : バーチャルリアリティの医学応用, 日本バーチャルリアリティ学会誌, Vol. 3, No. 1, pp. 32-41 (1998).
- [7] 石本 則子 : ゲームと VR, 日本バーチャルリアリティ学会誌, Vol. 4, No. 2, pp. 143-146 (1999).
- [8] Richard C. Waters, and John W. Barrus : The rise of shared virtual environments, IEEE SPECTRUM, Vol. 34, No. 3, pp. 20-25 (1997).
- [9] 内藤 剛人, 松田 晃一 : エージェント指向仮想社会の構築 - PAW -, 日本バーチャルリアリティ学会論文誌, Vol. 4, No. 2, pp. 399-406 (1999).
- [10] <http://www.blaxxun.com>
- [11] <https://www.japanet.co.jp/vrml/>
- [12] 岩田 大司 : 分散型仮想環境技術による保守訓練システムの連携作業訓練への拡張法の研究, 京都大学大学院エネルギー科学研究科エネルギー社会・環境科学専攻修士論文 (2000).

- [13] 山本 倫也, 岩田 大司, 長松 隆, 下田 宏, 吉川 榮和 : 大規模プラント保守訓練のための分散仮想環境のシミュレーション手法, 日本バーチャルリアリティ学会論文誌, Vol. 5, No.4, pp. 1103-1112 (2000).
- [14] 細谷 克美, 川野辺 彰久, 角田 進, 吉田 孝 : 仮想空間を利用した体験型協調学習システム:HyCLASS, NTT R&D, Vol. 46, No. 6, pp. 99-106, Aug (1997).
- [15] <http://www.3d-ies.com/>
- [16] 長松 隆, 嶋村 純二, 山本 倫也, 大石 創, 宮沢 龍雄 : 分散型仮想環境を用いたエネルギー教育システムの構築, 計測自動制御学会システム情報部門シンポジウム 1999 講演論文集, pp. 335-340 (1999).
- [17] 家崎 洋, 池井 寧 : 分散仮想環境のシミュレーション手法に関する研究, Human Interface N&R, Vol. 12, No. 2, pp. 163-169 (1997).
- [18] <http://vs.sony.co.jp>
- [19] M. Yamamoto, H. Yoshikawa, T. Nagamatsu and D. Iwata : Human Performance, Situation Awareness and Automation: User-Centered Design for the New Millennium, Savannah, U.S.A, October 15-19, pp. 330-335 (2000).
- [20] <http://www.isc.intsp.or.jp/>
- [21] <http://www.sharp.co.jp/sc/excite/cook/text/c-index.htm>
- [22] 郵政省 : 平成 12 年度版 通信白書 (2000).
- [23] Rodger Lea, Yasuaki Honda, Kouichi Matsuda, and Jun Rekimoto : Technical issues in the design of a scaleable shared virtual world, SCSL-TR-95-039 A (1995).
- [24] 松崎 剛士, 大坂 融弘, 山本 倫也, 長松 隆, 下田 宏, 吉川 榮和 : ネットワーク仮想環境における空間管理手法の研究, ヒューマンインタフェース学会研究報告集, Vol. 2, No. 2, pp. 137-142 (2000).
- [25] Richard C. Waters, David B. Anderson, William S. Yerazunis, 小塚 宏, 福岡 久雄 : 分散仮想環境基盤ソフトウェア"Spline", 三菱電機技報, Vol. 71, No. 2, pp. 26-29(1997).

- [26] Richard Fujimoto : Parallel Discrete Event Simulation, Communications of the ACM, Vol. 33, No. 10 (1990).
- [27] K. Mani Chandy and Jayadev Misra : Distributed simulation: A Case Study in Design and Verification of Distributed Programs, IEEE Transactions on Software Engineering, Vol. SE-5, No. 5, pp. 440-452 (1979).
- [28] Jayadev Misra : Distributed Discrete-Event Simulation, ACM Computing surveys, Vol. 18, No. 1, pp. 39-65 (1986).
- [29] Capin, Pandzic, Magnenat-Thalmann, Thalmann : Avatars In Networked Virtual Environments, JOHN WILEY & SONS, LTD (1999).
- [30] J.L.Peterson : ペトリネット入門, 共立出版社 (1984).
- [31] Maria Hybinette, Richard Fujimoto : Optimistic Computations in Virtual Environment, Proceedings of the Virtual Worlds and Simulation Conference (VWSIM '99), pp. 39-44 (1999).
- [32] <http://www.opengl.org>
- [33] 林 晴比古 : 新 Visual C++ 5.0 入門 ビギナー編・シニア編, ソフトバンク (1998).

付録目次

付録 A エネルギー教育用仮想環境に用いた外部モジュール	付録 A-1
付録 B 情報提示ウィンドウ	付録 B-1
B.1 概要	付録 B-1
B.2 入出力処理の概要	付録 B-1

付録 図目次

B.1 2次元画像提示ウィンドウに関わる外部モジュールの構成図	付録B-2
---	-------

付録表目次

- A.1 教育環境に用いた外部モジュールの一覧 その1 付録A-1
- A.2 教育環境に用いた外部モジュールの一覧 その2 付録A-2

付録 A エネルギー教育用仮想環境に用いた外部 モジュール

エネルギー教育用仮想環境に用いた外部モジュールの一覧を以下の表 A.1、A.2 に示す。

表 A.1: 教育環境に用いた外部モジュールの一覧 その1

モジュール名	詳細	
利用者入力 (ブラウザ)	入力	利用者のジョイスティック、マウス、キーボードの入力
	出力	アバタの移動量
	処理内容	利用者がジョイスティック、マウス、キーボードなどを利用した入力を数値に変換して出力する
利用者移動	入力	アバタの移動量、衝突した仮想オブジェクトの名前
	出力	アバタの姿勢
	処理内容	衝突した仮想オブジェクトがあれば移動せず、なければ利用者の入力に応じてアバタの姿勢を計算する
衝突判定	入力	アバタの移動量、利用者の姿勢、仮想オブジェクトの姿勢
	出力	衝突した仮想オブジェクトの名前
	処理内容	衝突判定があれば衝突した仮想オブジェクトの名前、衝突判定がなければ NULL を返す
風力特性 データベース	入力	風車の種類
	出力	風車の特性
	処理内容	風車の種類に応じて、風速と発電量の関係を表す風車の特性データを出力する

表 A.2: 教育環境に用いた外部モジュールの一覧 その2

モジュール名	詳細	
風力発生源	入力	インタフェース外部モジュールから CGI による入力 (風速 \downarrow 方角)
	出力	風速 \downarrow 風向
	処理内容	利用者が入力した風のデータを風速 \downarrow 風向に変換する
風力発生源	入力	計測器からの風速データ
	出力	風速 \downarrow 風向
	処理内容	計測器によって測定された宇治キャンパス本館屋上の風速 \downarrow 風向を出力する
風力発生源	入力	ホームページからの風速データ
	出力	風速 \downarrow 風向
	処理内容	インターネット上に公開されている風速 \downarrow 風向のデータを取り込み、出力する。ここでは東京電力が HP で公開している柏崎刈羽原子力発電所の風速 \downarrow 風向データを出力する
風車 シミュレータ	入力	風車の種類 \downarrow 風車の特性 \downarrow 風速 \downarrow 風向
	出力	風車の姿勢 \downarrow 風車の形状 \downarrow 発電量
	処理内容	風車の種類によってその形状を変化させる。また、風車の特性データ \downarrow 風速 \downarrow 風向をもとに風車の姿勢 \downarrow 発電量を計算する
吹流し シミュレータ	入力	風速 \downarrow 風向
	出力	吹流しの姿勢
	処理内容	風速 \downarrow 風向をもとに吹流しの姿勢を計算する
灯台 シミュレータ	入力	発電量
	出力	光の形状 \downarrow 姿勢
	処理内容	発電量をもとに灯台の光の形状 \downarrow 姿勢を計算する
インタフェース 外部モジュール	入力	CGI による利用者の入力
	出力	利用者に提示する HTML
	処理内容	詳細を付録 B に示す

付録 B 情報提示ウィンドウ

B.1 概要

5.1 節で述べたように、ブラウザの機能拡張として様々な情報を提示できるように、情報提示ウィンドウを実装した。このウィンドウは、利用者と仮想オブジェクトの位置関係に応じて、複数の情報から適切な1つを提示する。そして、インタフェース外部モジュール、CGI(Common Gateway Interface) と HTML を利用してデータを入出力する。インタフェース外部モジュールは、この CGI とロケールサーバを仲介してデータを入出力しながら、提示情報の切替を制御する外部モジュールである。情報提示ウィンドウは、Microsoft 社の ActiveX コントロールによる WEB ブラウザを MALIONET のブラウザに組み込んで実装した。

ウィンドウの提示では、仮想オブジェクト毎に異なる情報を提示できるように、仮想オブジェクトは形状、姿勢などの他に、利用者に提示する情報を保持している。例えば風車オブジェクトでは、風車の種類、発電量を利用者に提示するという情報を保持している。

B.2 入出力処理の概要

EEWING では、アバタが仮想オブジェクトに衝突した場合、利用者には仮想オブジェクトが保持している情報に基づき、インタフェース外部モジュールによって処理が行われ、情報提示ウィンドウに情報を提示する。以下の図 B.1 に情報提示ウィンドウに関わる外部モジュールの構成図を示す。

利用者に情報が提示されるまでの処理は以下のとおりである。

1. アバタが仮想オブジェクトと衝突すると、衝突判定モジュールがそれを検知し、インタフェース外部モジュールに衝突した仮想オブジェクトの名前を送信する。
2. インタフェース外部モジュールは、衝突した仮想オブジェクトが保持している情報をもとに提示用 HTML ファイルを HTTP サーバに格納する。また、ブラウザに CGI の URL を送信する。

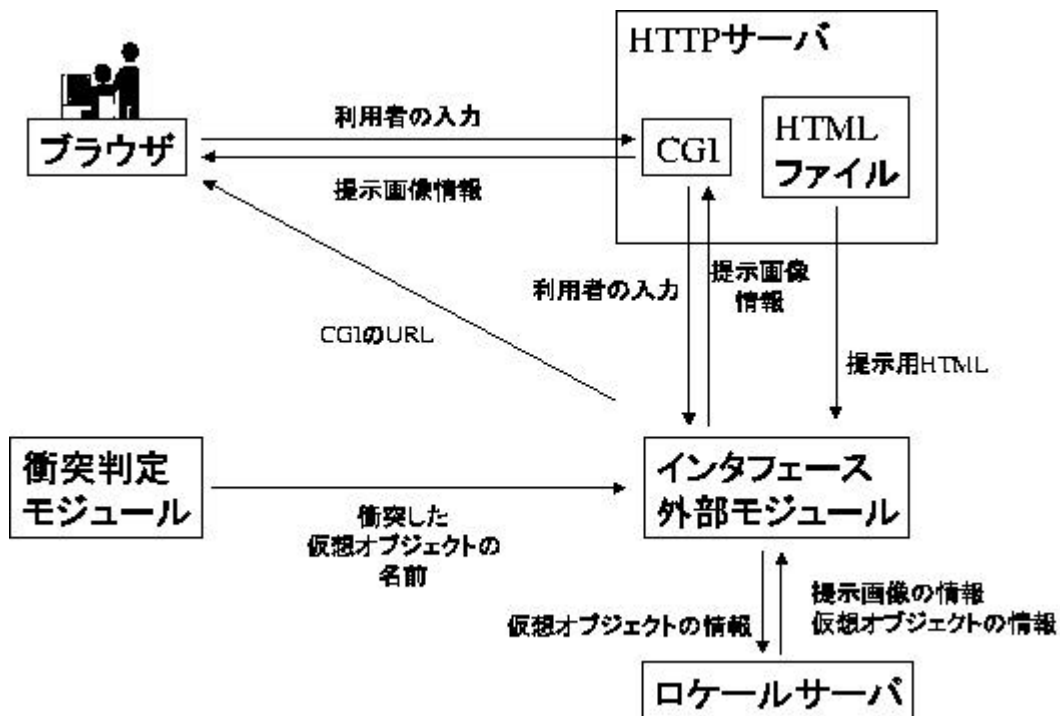


図 B.1: 2次元画像提示ウィンドウに関わる外部モジュールの構成図

3. CGI が提示用 HTML ファイルをもとに提示情報を HTML 形式に変換し、ブラウザでは情報提示ウィンドウに情報が提示される。

また、利用者が仮想世界に情報を入力する場合は、以下の処理が行われる。

1. 利用者は、情報提示ウィンドウで情報を入力する。
2. 情報提示ウィンドウが HTTP サーバ内の CGI を起動し、利用者が入力した情報をインタフェース外部モジュールに送信する。
3. インタフェース外部モジュールは、受信した情報に基づき、仮想オブジェクトのデータを変更し仮想世界を変化させる。
4. 情報提示ウィンドウに提示する情報が変化した場合、前述の処理によりその情報を利用者に提示する。

インタフェース外部モジュールがこのように処理して、利用者に仮想オブジェクト毎に異なるインタフェースを提供することができる。