ハイパフォーマンス
コンピューティング 51－3
（1994．6．17）

オブジェクトベース・シミュレーションのGUI実行支援環境

畠山 正行　上原 均
茨城大学工学部情報工学科

オブジェクトベースは、我々の提案するソフトウェアによる計算シミュレーションアーキテクチャの概念であり、これによって「もの」のモデル化の単位モジュールがそのままの単位毎に駆動するシミュレーションが実現した。しかしモデル化の観点などからいって、ユーザインタフェースにかんする記述はオブジェクトベース機構の記述とは別に行なわれるべきであり、またそのように実装されているため、オブジェクトベースの実現機構のみではユーザに対して十分なユーザインタフェースが提供できない。

そこで本研究では、実現したオブジェクトベース・シミュレーションに対するユーザインタフェースシステムの構築を目標とした。その目標を達成するために、オブジェクトベースにおけるユーザインタフェースのあるべき姿を議論し、その議論を踏まえた上でユーザインタフェース・システムの構成、構築を行なった。

本研究で述べているユーザインタフェースシステムは、あくまでオブジェクトベースに対するユーザインタフェースシステムとしては最初の段階の一部であり、技術的または機能的な点でやや不足が見られるが、オブジェクトベース機構に対するユーザインタフェースシステムの必要性などを証明する上で充分なものとなったことを確認した。

GUI Execution Support Environment for ObjectBased Simulations

Masayuki Hatakeyama, Hitoshi Uehara
(Ibaraki University)

ObjectBased Mechanism is an architecture for the computer software simulations. We proposed a new simulation architecture for the natural phenomena. But, the user interface for the objectbased mechanism is not enough to handle the objects in the objectbased world. The user can't handle the objects in the objectbased world so freely as those when handling the objects in real world.

Therefore, we developed a UI environment for the objectbased mechanism. In this paper, we will present the UI execution support environment for objectbased mechanism, and also present a GUI execution support environment, which is a sample of UI execution support environment for the objectbased mechanism that we implemented.

As the result, though this sample UI remains some problems which should be improved, the necessity and the validity of the UI system for objectbased mechanism are verified.

1 はじめに

オブジェクトベース・シミュレーションとは、ソフトウェアによる一つの計算機シミュレーション・アーキテクチャのことである [1][2]。それは具体的には、計算機シミュレーションの対象世界に対してオブジェクト指向モデリングパラダイムに従い一貫したモデリング過程に基づく変換を行う。まずそのモデルを「もの」に対応した自然な単位モジュール「オブジェクト」及びその内部に他のオブジェクトとの相互作用の定義を記述しておく。それらを対象世界と相似な方式で起動・駆動し、多数のオブジェクト間で相互作用をしつつ対象世界全体を振舞わせることによって動的なシミュレーションを行おうとするアーキテクチャである。オブジェクトベース・シミュレーションの様子を図1に示す。

この方式のシミュレーションにおいては、ユーザインタフェースの観点から見ると対象世界のモデルは以下の特徴を持つ。

1. オブジェクトの起動・駆動及びオブジェクト間での相互作用表現能力は持ち得るが、オブジェクトベースモデルが作る世界（＝オブジェクトベース世界）の外界に対する表現能力はそのモデル構成上、デフォルトには持っていない。

2. オブジェクト自身あるいはオブジェクトベース世界自身は計算機の OS 及びその周辺に関する記述を必要最小限しか持たない。

3. ハードウェアに関する記述は全く持たない。つまり、デバイスに依存するようなプログラムを記述・内蔵できない。

上記の第1項の特徴については、むしろ持たすべきではないというモデル構築上の要請がある。それは「もの」に対応できる、または「もの」と同等の記述をし、「もの」と同等の振る舞いをさせるべきであるからオブジェクトベース世界外への表現能力は固有には持ち得ないはずとの厳密な要請（条件）から導かれた。

2 オブジェクトベースのユーザインタフェース

前章の三つの特徴から、オブジェクト－人間間のインタフェースに関して改めて如何に構築するかを考えなければならないことが分かる。

2.1 モデリングの観点

前章で述べたモデル構築上の条件を満たしたオブジェクトベース世界が構築されたとすると、これはいわゆる典型的な閉じた世界である。つまり、外界との接触（インタフェース）に関する記述が全くなされておらず、現実世界に存在するユーザがインタラクションを持とうとしても、その方法が存在しない。従って、シミュレーション駆動中ではあるが、モデル化された対象世界の再現シミュレーション駆動の様子を、（グラフィックス表現などで）見られない、操作できない、必要なデータを抽出することもできない、実装法を誤ると対象世界を起動することもできないという状況が必然的に生じることとなる。

このようなシミュレーション駆動世界に対して、再現シミュレーションされている世界の可視化、操作、データ抽出に関する手続きをどう行なえば良いのか。これは、要するにオブジェクトベース世界の構築・駆動アーキテクチャに合わせてユーザとのインタフェース（ユーザインタフェース）をとれば良い。具体的に言うと、

1. オブジェクトベース世界とユーザのインタフェースを取るためには、オブジェクトベース世界内にユーザの代理人（エージェント）を生成させる必要がある。図1にもそれが表現されている。

2. エージェントはオブジェクトベース世界の他のオブジェクトと相互作用を行って、各オブジェクトからインタフェースをとるのに必要な情報などを収集し、また他のオブジェクトへの操作や制御情報を送る役目を果たす。

3. 前述のオブジェクトベース世界の特徴の第3項から、収集され外界に持ち出された情報をグラフィックス表現などの何らかの表現に再構成する機構、及びオブジェクトベース世界

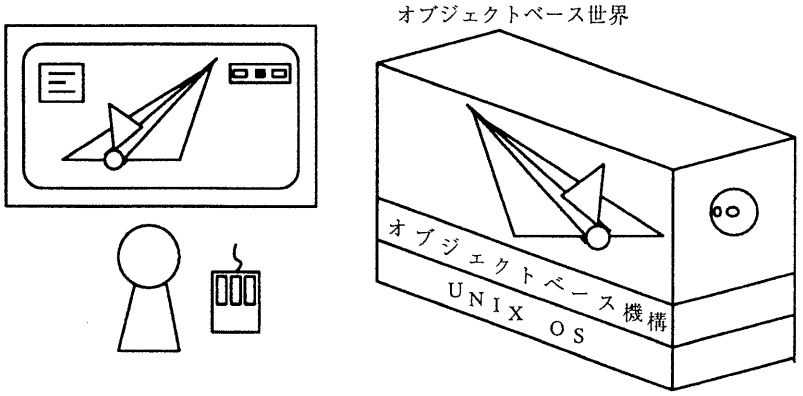

図 1: オブジェクトベース・シミュレーションの様子

へ伝達する情報をオブジェクトベース世界の表現伝達モデルの形式に変換する機構をオブジェクトベース世界の外部に構築する必要がある。

4. これはオブジェクトベース機構にかなりの部分が含まれているが、OS が行うシステム的な実行制御、リソース割り当て、オブジェクトの生成・格納の仕掛け、オブジェクトのカプセル化の仕掛けの駆動をオブジェクトベース世界に代わって実行する機構（駆動支援環境）が必要である。

これらはいずれも従来の手続き型プログラムには記述可能であり、また記述もされているが、オブジェクトベースの実現としてはその内部に記述してはならないものである。このような場合の解決方法は、駆動しているオブジェクト群の相互作用の発現（メッセージ通信）を常時傍受しておき、その情報の内部からサポートを必要とする機能を抽出し、データ収集を行って外部機構でそれを実現・処理し、オブジェクトベース世界の内部オブジェクトに対して（内部にある OS のエージェントを通して）それらを提供するという機構が必要である。それは即ち実行支援環境又はユーザインタフェース環境の必要性に他ならない。

2.2 現実の「もの」とオブジェクトの対応

オブジェクトは、人間がモデル化することによって構成されたものであり、自然シミュレーションに限っていえばモデル化の対象となった「もの」は、おそらく対象世界に存在する。オブジェクト指向では、メッセージ通信という統一的なインタフェースを提供することにより、モデル化されたオブジェクトを、モデル化の範囲内で現実に存在したものと原理的には同様に扱えることを保証している。

これらのことから言って、モデル化されることによる情報の欠落を無視すれば、オブジェクトベース内のオブジェクトと現実のものは本来同等に扱うことができるべきものである。

しかし、対象世界では「もの」の扱い方はメッセージ通信という形式のみに限定されず、通常、人間から見てかなり無意識的に取り扱える場合が多い。現状のインタフェースでは、コミュニケーション手段の限定によってオブジェクトベース世界の表現が本来在るべき姿からかなり離れており、必然的にオブジェクトベース内のオブジェクトと対応すべき現実の「もの」が別のものであるという感覚を半ば無意識的にユーザに与えるものであり、オブジェクト指向モデリングパラダイムの点から見ても、満足できる状態ではない。

これは、人間のオブジェクトベース世界のオブジェクトを認識する上で、そのものの構造だけでなく、インタフェースの構築方法が重要な位置を占めていることを示すものである。つまり、我々が対象世界の事象をシミュレーションする場合、その事象の構造などを忠実にモデル化するだけでなく、その事象とユーザのインタフェースに関しても、あるべき姿を考察し、モデル化の過程に組み込むべきであ

る、と考えている。

これらの考えをさらに進め、我々はオブジェクトベース内部に存在するオブジェクトが対象世界に存在する「もの」と（モデルレベルで）ほぼ同等に取り扱うことのできる状態を「オブジェクトリアリティ」として規定し、その状態を成立させることがオブジェクトを操作する上での理想であると考えている。現在、主にユーザインタフェースの分野で「バーチャルリアリティ」という仮想現実に関する研究が盛んに行われているが、現状ではデータの取り扱いや対象世界との対応性に欠ける点が欠点として挙げられる [3]。この欠点により、現状のバーチャルリアリティのシステムでは、その内部世界を対象世界をモデル化して構築しているにも関わらず、対象世界では（物理的に）起こり得ないような現象が起こり得るシステムとなっている。具体例を挙げるならば、「インテリアシミュレーション」において、ユーザが「壁」を無視して移動できたりする現象が挙げられる。

我々の提案する「オブジェクトリアリティ」では、オブジェクトベース内部にオブジェクト指向に基づいたモデル化によって対象世界の構造と振舞いを再構成し、それをオブジェクトを常時<データ構造＋メソッド>の形式で駆動させ、そのオブジェクトの形を忠実に表現することにより、ユーザに現実感を与えることを最終的な目的としており、従ってそのモデリング方法や実装方法はバーチャルリアリティとは全く異なる。

3 システムの構成

前章で述べたように、オブジェクトベース世界に対するのユーザインタフェースシステムとして新規にユーザインタフェース実行支援環境として構築すべきである。以降では、それらを踏まえてオブジェクトベースのユーザインタフェースシステムの構成などを述べていく。また、これ以降で述べるシステム構成はあくまでオブジェクトリアリティの実現の最初の段階の一部であり、現在、更にモデル的な拡張を行なっていることをお断りしておく。

3.1 システム全体の構成

本研究で構築するユーザインタフェース実行支援環境は、参考文献 [1] で述べられているオブジェクトベース機構と接続されるものとした。まず、オブジェクトベース機構とユーザインタフェース実行支援環境の接続に関しては、オブジェクトベース機構側にユーザのエージェントとなる E-OSQL サーバー [1][2] を用意し、これとユーザインタフェース実行支援環境を接続させた。E-OSQL サーバーは、データベースで用いられる SQL 方式に類似した E-OSQL という方式で現実世界のユーザから命令を受け取ってオブジェクトベース世界内部のオブジェクトに対しメッセージ通信を行なうことで、ユーザとオブジェクト間のインタラクションを可能としている。

また、オブジェクトベース機構とユーザインタフェース実行支援環境は、システムの規模などから見て単独のプラットフォーム上に同時に存在し得るのは、負荷やハードウェアの面から見てかなり難しい。従って、複数のプラットフォームをネットワーク上で連結し、負荷分散や機能分散を計るシステム構成とした。

3.2 リアルなグラフィックス表現

オブジェクトベース世界内部のオブジェクトを、対象世界に存在する「もの」と対応させる形で取り扱うには、高度かつリアルタイム的なグラフィックス表現を可能とするグラフィックスシステムが不可欠である。何故ならば、現実世界に存在するユーザは視覚を通じて「もの」の情報の多くを入手し、様々な行動を起こす上での基盤としているからである。この視覚によって得られる情報は、単純な形状的情報だけではなく、物体の質感や振舞いによる外観的変化などの情報も含まれる。オブジェクトベース上でのシミュレーションとは、オブジェクト指向に基づいた仮想現実においてシミュレーションを行なうことと同義であるから、現実世界のユーザの視覚と同等な、高度なグラフィックス表現がオブジェクトを操作する上で不可欠となる。

また、静的なグラフィックス表現しか行えないようなグラフィックスシステムではメソッドの実行などにより常時変化し得るオブジェクトのリアルタイム的な表現は原理的に不可能である。よって、オブ

ジェクトのグラフィックス表現は、現実に存在する事象と同じように極力現実味のある、リアルタイム的に変化を表現しうるグラフィックスシステムで行われるのが最良であろう。

本来ならば、このグラフィックスシステム自身をオブジェクト指向に基づいたシステムとして構築することが理想的である [4][5]。なぜならば、従来のグラフィックスでは可視化される物体の形状のみが重要視され、本来のオブジェクトの忠実な表現という観点はあまり見られなかったからである。

我々の研究では、オブジェクトを常にデータ構造と振舞いから構成されるものとして取り扱い、グラフィックス表現するシステムを研究の一環として開発、実装している [4][5]。しかし、現状ではグラフィックス表現のリアルタイム性に難があるため、本研究ではオブジェクト指向グラフィックスシステムによる実装を避けた。しかし、現在、リアルタイム的な表現が可能なグラフィックスシステムを開発中であり、将来的にはオブジェクト指向グラフィックスシステムがオブジェクトベースのグラフィックスシステムとして利用可能になる。

3.3 ユーザの意思の入力の支援

現実に存在する「もの」は通常、様々な取り扱いが可能である。しかし、オブジェクトベース内のオブジェクトは最終的にはメッセージ通信によるアクセスしか許していない。この落差はかなり大きく、現実の「もの」と対応するものとしてオブジェクトを取り扱う上でハンデとなり得る。

そのハンデを克服する上で、GUI システムによるオブジェクトの操作に関する支援は重要となる。なぜならば、現実的に構築しうるシステムとして複雑かつ動的に変化しうるオブジェクトの取り扱いを行えるシステムとしては、GUI 方式が最良の方式と思われるからである。

また、オブジェクトベース内のオブジェクトを操作する方法は現状では E-OSQL という方式のみであるが、これは柔軟ではあるが煩雑な方式である。これを直接ユーザが用いて操作する方法では、ユーザの負担が重くオブジェクトリアリティも得られないことは容易に予想される。そのため、操作の自由度を極力落とさずに操作性のみを向上させるために、オブジェクトを操作するための内部的な支援システムを、GUI システム内部に内蔵する必要がある。E-OSQL 方式では、複数のパラメータをある特定の文法に当てはめ、命令文を生成することで操作が行われるので、パラメータの指定以外の行動を支援システムが代行することができれば、現状で考えられる最も容易なユーザインタフェースが構築できるものと思われる。

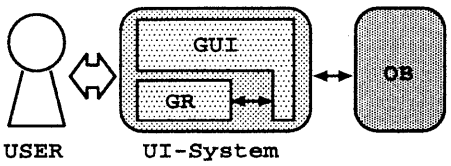

図 2: ユーザインタフェースシステムの全体図

4 実装

今回述べるユーザインタフェースシステムの実装例は、大きく分けて以下のような二つの部分から構成されている。

- 二次元または三次元の画像を表示するグラフィックス部分
- オブジェクトの操作を可能とする GUI 部分

グラフィックス部分は GUI 部分だけでは不十分なデータ解析やオブジェクトベース世界内部の可視化に用いられ、GUI 部分は主にオブジェクトベース世界内のオブジェクトへのメッセージ通信に関して用いられる。また、ユーザインタフェースシステム全体も GUI 部分で制御される。

4.1 システム全体の構築

オブジェクトベースのユーザインタフェースシステムは、グラフィックス部分（GR 部分）と GUI 部分の二つから構成される（図 2参照）。この二つは独立したプログラムとして実装され、UNIX のプロセス間通信を利用して接続されている。これは 3.1 節で述べた機能分散または負荷分散の必要性から採用した構築法であり、UNIX ネットワーク上での利用をデフォルトとして仮定している。

また、ユーザインタフェースシステムは、本質的にオブジェクトベースとは異なったシステムであ

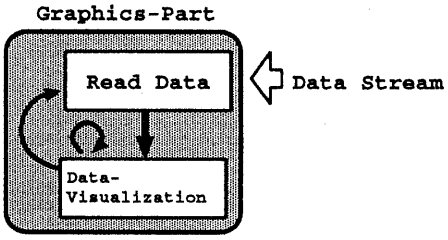

図 3: グラフィックスルーチン

り、それぞれの構築において依存するような記述は避けるべきであるとの観点により、極力依存性を排除した記述で構築されている。よって、相互に不可欠な機構でありながら、必要に応じて自らの変更などを行いやすいシステム構成となっている。

4.2 グラフィックス部分

3.2節で述べた条件を満たしうるグラフィックスシステムとして、今回はアプリケーション・ビジュアライゼーション・システム (Application Visualization System：以下 AVS と略)[6] を利用して、オブジェクトベース専用の可視化システムを構築した。AVS システムは、比較的に容易に高レベルな三次元あるいは二次元表示が可能なシステムである今回の実装ではその表現力の高さに注目し、グラフィックス部分構築の基盤システムとして用いた。

また前述のようにオブジェクトベースのユーザインタフェースとしてグラフィックスシステムを構築する場合、オブジェクトの動的な変更をリアルタイム的にグラフィックス表現を行うことが重要な機能となる。当然、その動的な変更に関する情報をオブジェクトベース内部のそれぞれのオブジェクトから受け取る機能が必要となる。

そこで本研究のグラフィックス部分には AVS のコルーチンを利用して、この機能を実装している。ここで AVS のコルーチンとは、その生成から消滅までの間、常時実行され続けている無限ループを持つ特殊なルーチンである。通常ではアニメーションのデータ生成などに用いられるが、その機能をプロセス間通信の常時読み込みとその変換処理などに用いることによって、オブジェクトベース内部のオブジェクトの動的な変更を常時読み込み、可視化する機能を実現している（図 3参照)。

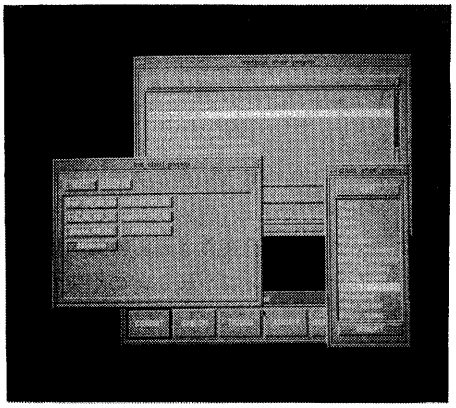

図 4: GUI を用いた操作

4.3 GUI 部分

本システムにおけるユーザインタフェースシステムでは、オブジェクトベース世界との相互作用の機能を実現しているのが GUI 部分であり、GUI 部分では、OSF/Motif[7] を用いて構築を行なった。OSF/Motif は、イベントドリブンという形式で処理が行われる。本研究では、ユーザからの入力をイベントという形で処理することで GUI システムを実現している。

GUI 部分では、ユーザがボタンやメニューを用いて指定したパラメータから、オブジェクトに対する指示を生成、ユーザのエージェントである E-OSQL サーバーに送信することでオブジェクトベース内のオブジェクトの操作を行なっている。GUI 部分ではそれぞれのウィンドウにクラス名やインスタンス名、メソッド名を表示し、それらからユーザがパラメータの指定を行うことで、半自動的に E-OSQL 文を作成、オブジェクトを操作している（図 4参照)。

5 システムの駆動

この章では、前章で述べたシステムの実際の駆動に関して述べる。

実際の駆動画面は図 5のようになっている。図 5 から分かるように、GUI 実行支援環境はマルチウィンドウ・システムをベースとしている。

実際の GUI システム上からのオブジェクトの操作は以下のように行われる。

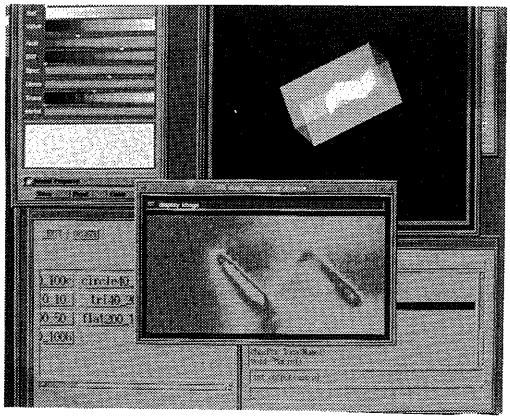

図 5: 実際の画面

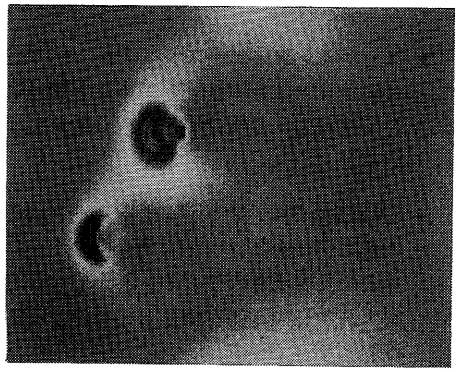

図 6: 二次元表示

1. オブジェクトのクラスを指定する。⇒ 指定されたクラスのインスタンスが表示される。

2. メッセージ通信を行いたいインスタンスを指定する。⇒ インスタンスに送ることのできるメソッド群が表示される。

3. 送るメソッドを指定し、必要ならばメソッドのパラメータを指定する。⇒ オブジェクトの操作が行われ、オブジェクトの操作の結果がインタフェースシステム上で表示される。

オブジェクトへの操作の結果、GUI 実行支援環境側への戻り値がグラフィカル表現すべきであるようなデータ流であった場合、GUI システム内部で自動的に判別され、操作の 3番目の段階でグラフィックスシステムでグラフィックス表現されることとなる。実際にグラフィックス表現されるのは、このシステムの例では風洞などの三次元形状とシミュレーション結果として得られる巨視物理量である (図 6 参照)。また、三次元的形状のグラフィックス表現に関してはシミュレーション中の状態変化をアニメーション的表現で可視化できる (図 7参照)。

上記のように、GUI システムの支援によるオブジェクトの操作のためのメッセージ通信の簡便化と、グラフィックスシステムによるリアルなかつリアルタイム的なグラフィックス表現によって、オブジェクトベース世界内部のオブジェクトが視覚的に「目に見える」存在として容易に操作できるようになった。

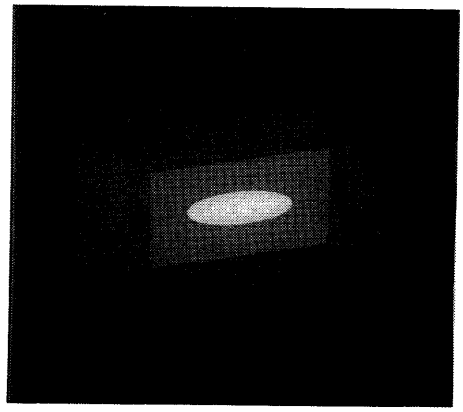

図 7: 三次元形状の表示

6 システムの評価と結論

前述の GUI 実行支援環境の実装例の評価とそれらから導き出された結論を以下に述べる。

- オブジェクトの操作方式である E-OSQL 文を、GUI によって半自動的に生成することにより、簡便にオブジェクトの操作ができるようになった。

- GUI 部分とグラフィックス部分の連動により、シミュレーションの結果のデータ解析やオブジェクトのデータの可視化などが容易かつリアルタイム的に可能になった。

- グラフィックス部分は AVS システムを利用して実現することにより、かなり高度なグラ

フィカル表現がリアルタイム的に表現できる。

- ユーザインタフェースシステムは、GUI部分とグラフィックス部分をプロセス間通信を用いて接続し、機能分散と負荷分散を実現している。
- GUI部分では、オブジェクトベース内の1つのオブジェクトはウィンドウ上の1つのボタンとして表現されており、不十分ながらオブジェクトを視覚的実体として操作することが可能となっている。
- オブジェクトの保持する情報を基にした二次元及び三次元画像を生成できるが、その画像を利用したユーザからの直接入力は現状では実現できていない。また、テキストなどに関する入出力における相補性も獲得できていない。

今回述べたGUI実行支援環境の評価をまとめると、技術的な面において改良すべき点は見られるが、オブジェクトベースに対するGUI実行支援環境として有効であったことは明らかであり、またオブジェクトベース・シミュレーションに対するGUI実行支援環境自体の必然性を実証したものと思われる。これは、本研究で述べたGUI実行支援環境によって、インタフェースに関する記述を殆んど持たないオブジェクトベース内部のオブジェクトを、ユーザが「目に見える」存在として操作可能となったことから明らかと思われる。システム全体としてはユーザインタフェースシステムとして提供すべき機能を最低限有し、システムの構築方針なども基本的に正しかったものと思われる。

7 将来の展望

本研究での前述のシステムモデルについて更に考察を深めた結果、現在では「オブジェクトリアリティ」を実現するためのより進んだモデルを考案した。現在は、そのモデルに関する考察を深め、そのモデルに基づいた実装を行っている。

現状のユーザインタフェースシステムでは、ごく単純にオブジェクトベース内のオブジェクトの操作と可視化が簡便にできるシステムに過ぎないが、オブジェクトリアリティを実現するようなユーザインタフェースシステムの構築が期待されており、現在上記の新しいモデルに基づいて構築中である。

オブジェクトリアリティと類似する研究としては、バーチャルリアリティ[3]が挙げられるが、我々が現在構築中のモデルは実装法がそれとは全く異なり、データハンドリングの問題などからいって、我々のモデルに基づくシステムの方が、特にシミュレーションの分野などでは高いリアリティを持つことになると確信している。

参考文献

[1] 畠山 正行、金子 勇、「オブジェクトベース機構：オブジェクト指向一貫モデリング過程論に基づくシミュレーションの実現」、情報処理学会第17回プログラミング研究会、平成6年6月3日。

[2] 畠山 正行、金子 勇、「オブジェクトベース機構に基づく数値シミュレーション」、情報処理学会第51回ハイパフォーマンスコンピューティング研究会、平成6年6月17日。

[3] M.W. クルーガー著、下野 隆生訳、「人工現実インタラクティブ・メディアの展開」、トッパン(株)、1991年11月25日。

[4] 新谷 学、畠山 正行、「オブジェクト指向の概念を導入したグラフィックス表現法」、第67回情報処理学会グラフィックスとCAD研究会、情報処理研究会報告書 vol.94,No.17,pp.25-32,1994年2月18日。

[5] 畠山 正行、坂石 卓哉、「オブジェクト指向に基づく視覚表現系及び操作系の研究」、第48回情報処理学会全国大会講演論文集 3-235～236、1994年3月25日。

[6] "AVS Developer's Guide", Advanced Visual System Inc.,1992

[7] 兜木 昭男他著、「X-Window OSF/Motif プログラミング」、日刊工業新聞社、1990年10月20日。