

オブジェクト指向に基づく一貫相似性モデリング過程の方法

島山 正行[†]

要約 対象世界の再現シミュレーションにおいては、対象世界と再現シミュレーション世界との間の構造や機構、振る舞い等々にわたる相似性の高さが重要である。本論文では、両世界の間を繋ぐ幾段階にもわたるモデリング段階を、一貫したモデリングパラダイムで実現することが高い相似性を実現する最良の方法の一つであるという観点に立ち、そのモデリング方法とモデル、一貫相似性成立の検証方法、そしてそれを実現するための特別な駆動機構の構築により、一貫相似性の高いモデリング過程の実現方法を実現し、検証する。ただし、シミュレーションを行うドメインユーザの要求に応えるため、手続き型言語での実装を現行駆動環境（UNIX）上で実現することを可能にする。そのためにモデリングパラダイムとしてオブジェクト指向を選び、あるオブジェクトモデル（OB オブジェクトモデル）を提案する。それらを使って一貫相似性モデリング過程に基づく再現シミュレーションを実現し、そのモデリング作業過程詳細を追跡（トレース）して、本論文で提案した一貫相似性の基準に照らして評価・検証した。また実装具体例の結果等から、OB オブジェクトモデルの有効性・有用性・実用性を検証し、一貫相似性モデリング過程の実現を実証し、手続き型言語を用いて実装しても相似性の高いシミュレーションが実現する方法を確立できたことを確認した。

A Methodology on Integrally Consistent and Similar Modeling Processes based on the Object-oriented Paradigm

MASAYUKI HATAKEYAMA[†]

ABSTRACT We consider that one of the best methodology to develop the simulation system with a "high similarity" is to establish the modeling/programming processes based on an integrally consistent and similar modeling paradigm throughout from the target world up to the re-constructed simulation world. To realize this purpose, the Object-oriented modeling/programming paradigm has been adopted and extended, and an object-model called the extended OB model has been proposed. We have implemented and realized this simulation methodology, and have verified their reasonability and the practical applicability. To verify the integrated consistency and similarity in this modeling processes, some integrated similarity criterion and the verification procedures have been proposed. We have also proposed and verified the validity of the OB mechanism on the UNIX system for the simulation (domain) users to implement their simulation programs by making use of the procedure type programming languages. By making use of the extended OB model and the OB mechanism, we have established and verified the validity and the utility of the Object-oriented integrally consistent and similar modeling processes, and also the realization of a new kind, practical Object-oriented simulation methodology for the simulation (domain) users.

1. はじめに

科学技術計算においては、対象とする自然世界や工学的な設計物・構築物を計算機の中で再現或いは予測のためのシミュレーションがしばしば行われる。その様な計算のためのソフトウェア開発法や支援環境はビジネス分野と違って余り存在しないのが現状である。しかし、その様な開発法の需要は計算を行うユーザ（ドメインユーザと呼ぶ）がプログラミングやシステ

ム開発法の専門家でない故により強くかつ潜在していると言える。

オブジェクト指向とは構造化手法の一種であるともできる。そこで、この構造化の手法をある種の枠組みという形式で捉えれば、その構造的な形式性が対象世界をモデル化する際の強力な指針となることが予想される。そこで我々は、我々が対象とする自然現象や工学的な構築物の再現シミュレーション、つまりいわゆる科学技術計算の分野（或いは磯田の論文⁷⁾でいう真性実世界）に対してこの様なオブジェクト指向モデリングパラダイムを適用し、例えば航空関係の流れの問題（風洞など）に対して実現してきた。

[†] 茨城大学工学部情報工学科
Department of Computer & Information Sciences, Faculty of Engineering, Ibaraki University

現状においては対象世界のモデル化の最初から、シミュレーションに至るまでの過程全体についての方法についてしっかりと確立された方法があるとは言いがたい点が多い。例えば、モデリングの過程の中の各段階の間でオブジェクト指向記述モデルを如何にするか？、対象世界との相似性をどの様に保ち、そのことをどの様に検証するか？、科学技術計算で主用されている Fortran との整合性が取れるのか？とかいう点である。本論文で扱う対象世界は磯田の言う真性実世界であり、通常ソフトウェア工学の対象とされている擬似実世界ではなく、また、そのユーザは自身でソフトウェアを作成し、自身で改訂しながら使う「ドメインユーザ」である。

そこで、本論文ではモデリング方法に焦点を当て、一貫したモデリング過程を実現する方法を確立することによってドメインユーザに対象世界の概念モデリングに始まり、シミュレーション駆動に至るまでの作業過程内部における作業指針と相似性の検証方法を提供することを試みる。それにより、ソフトウェア設計・実装という観点からは間接的であろうとも、特にドメインユーザにとっては、如何にモデル化をするかに付いての有効な指針となり、ドメインユーザがソフトウェア開発を開発する際にそのモデル化概念の形成に役立ち、従って、有用なモデリング作業基盤を形成するであろう事が予測されるからである。その様な成果が得られれば十分な価値があると考えられる。

従って本論文ではドメインユーザ向けに、何をどう分析・抽出し、何を記述すべきかに付いて主要な着目点として論じ・提案することにする。具体的にはオブジェクト指向記述モデルをモデリング過程の全段階に亘って提案し、その使い方と、相似性の検証を行う方法を提供しようと試みるものである。

我々は以前から科学技術用計算のために OB オブジェクトモデルと呼ぶ具体的なオブジェクト指向モデルを構成し⁴⁾、オブジェクト指向言語 (C++) を用いてオブジェクト指向シミュレーションを一貫相似性モデリング過程に近いと思われる手順で実装・駆動させることをまず実現した^{1), 2), 20)}。これらの実現例は結果的には本論文の一貫相似性と一貫相似性モデリング過程の詳細な枠組みの基準にほぼ適合する実現を残した。「もの、シング」モデルとオブジェクトモデルの一对対応関係を表の形で明示的な記述にしたものを表 1 に示す。しかし、モデリング過程の方法論の構成という理論的な側面から見ると、

- (A) OB オブジェクトモデルの適用範囲が狭い
- (B) 一貫相似性の成立の明示的な検証方法が未確立

表 1 “もの、シング”モデルとオブジェクトモデルの対応関係

“もの、シング”	オブジェクト
属性、データ	属性
“もの”の振る舞いの記述	メソッド
構成部品・要素のまとまり	カプセル化単位
相互作用 (アクセス) 制限	情報隠蔽
“もの”の間の相互作用伝達	メッセージパッシング

表 2 ドメインユーザからの要求記述

(1) 記述言語の任意性 (新規記述言語の回避)
(2) 既存プログラムや蓄積ライブラリの再利用
(3) 現用駆動環境 (UNIX) での互換性の高さ
(4) 現在の高い処理実行効率の維持

- (C) 一貫相似性モデリング過程の方法が OB オブジェクトモデルや OB 機構と一貫した合理性や整合性を持つことの検証が不十分
- (D) OB 機構がオブジェクト指向パラダイムと一貫相似性モデリング過程に対する適格性・正当性・整合性の検討や論証が不十分

であった。そこで本論文ではこれらの点に関する再検討を行うことでオブジェクト指向に基づく一貫モデリング過程成立の検討と応用を試みる。

2. 一貫相似性モデリング過程の構成

2.1 ドメインユーザの要求

現状のシミュレーションシステムでは、主として手続き型のモデリング/プログラミングの考え方と方法、Fortran や C 等の手続き型の言語 (または専用のシミュレーション用言語) を用い、蓄積されてきた莫大なソフトウェア資産を利用して構築し・実行している場合が圧倒的に多い。また、ドメインユーザはシステム開発技術自体には関心が薄い場合が当然多く、新しいモデリングの考え方や不慣れなプログラミング言語、新規な開発技術・技法に必ずしも積極的ではない点も考慮しなければならない。そのようなドメインユーザからの具体的な要求記述 (要請) は、殆ど手続き型を前提とした表 2 の様な四項目に集約される^{4), 5)}。

2.2 一貫相似性モデリング過程

各モデリング段階の共通基本構成を図 1 に、一貫相似性モデリング過程全体構成を図 2 に示す^{1), 2)}。紙幅節約のために図 2 には次節で提案する OB オブジェクトモデルの記述が既に入っている。

2.3 OB オブジェクトモデル

OB オブジェクトモデルの構成を表 3 に示す。表 3 は“もの、シング”、あるいはオブジェクトと呼ばれる最小単位のモデル構成を示す。表 3 は、図 1、図 2

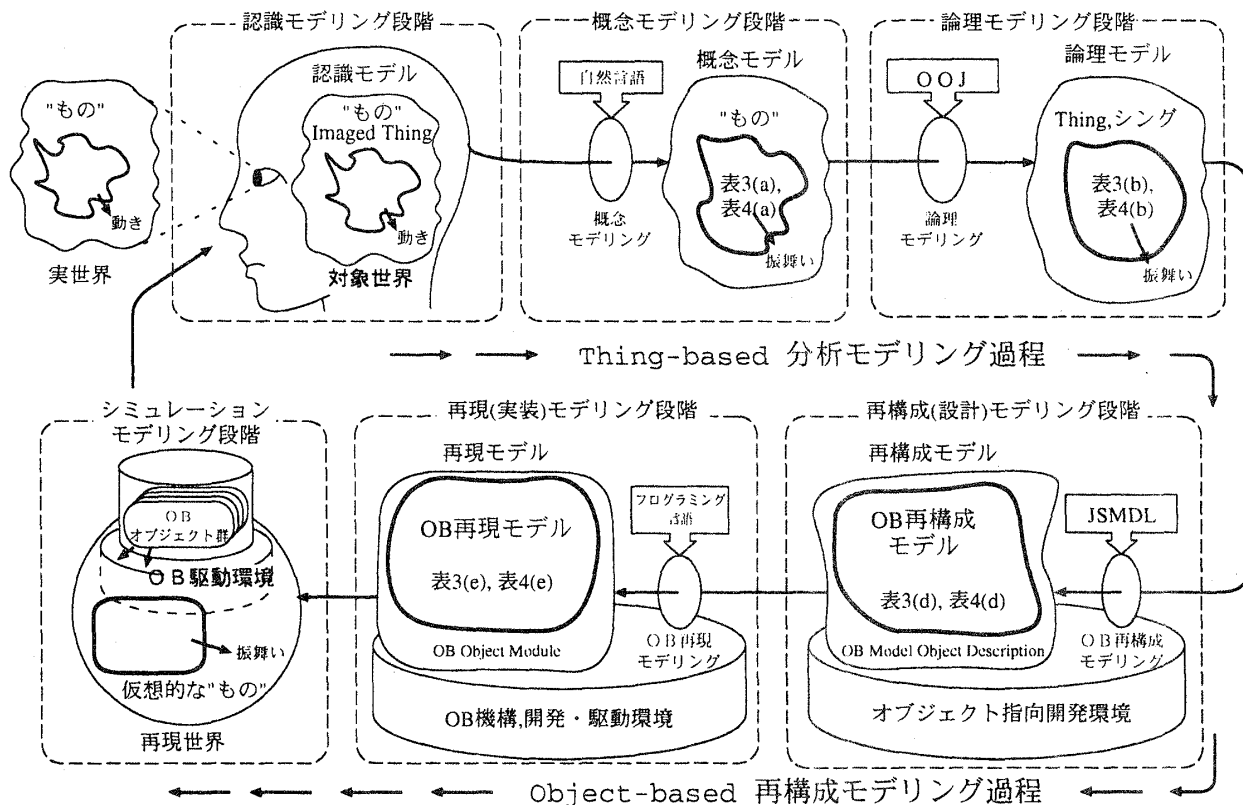


図2 一貫相似性モデリング過程 (OB オブジェクトモデル内蔵の場合の例)

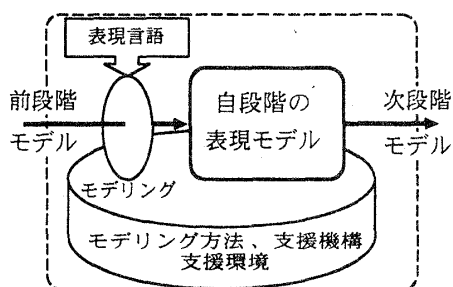


図1 各モデリング段階の基本構成

の各モデリング段階に各々対応させるべき記述モデルであり、既に図2の中に対応させて記入してある。OB オブジェクトモデル^{4), 6)}は、「オブジェクト」モデルとしての構成そのものは一般に見られるものとの共通部分が多く^{10)~12)}。オブジェクト指向の分類から言えばオブジェクトベースに近い^{10), 11)}。モデルの名前付けもそこに由来する。

表中の相互関係という用語は、付録の用語の定義を参照されたい。本 OB オブジェクト指向記述モデルの特徴の一つは相互関係を「もの」、シング」、或いはオブジェクト、の属性と設定した定義にある⁸⁾。分析記述モデリング段階における相互関係の記述に関する

やや詳細な構成要素定義を表4にその他の記述構成要素も含めた形で示す。

3. 一貫相似性の定義とその検証の方法

相似性を判断・検証する際には「似たような」構造(付録の用語定義参照)とか振る舞いを抽出・比較し、それらが「どの程度似ているか」を評価する方法が通常よく考えられる。しかしこの方法自体を分析して行くと、オブジェクト指向においては相似性の成立の可否は各オブジェクト及びその内部の最小構成要素に始まる記述要素間に一対一対応の相互関連(付録の用語定義参照)を持つか否かの関連の同定と、その一対一対応の相互関連の「高さ・良さ、即ち記述落差の小ささ」の評価と、の二つの問題に帰着する事が分かる。そこで、一貫相似性の定義とその検証方法を以下のように提案する。

3.1 一貫相似性の定義

一対一対応関連

モデル記述の各構成要素毎に記述の変換を追って行き、その任意の複数のモデリング段階前後の記述要素間に張って定義記述した相互関連(付録の用語の定義参照)が一対一対応をしていることを指す。

表3 最小単位のOBオブジェクトモデル

モデリング段階	モデル構成
(a) “もの” ベース 概念モデリング段階	
	最小単位 “もの” 概念モデル = (抽象 “もの” 名) + (振る舞い記述群) + (相互関係記述) 相互関係記述 = (相互関連記述) + (相互作用記述)
(b) “シング” ベース 論理モデリング段階	
	最小単位シング論理モデル = (抽象シング名) + (サブシング記述群) + (相互関係記述) サブシング = ((振る舞いの論理的記述 (式等)) + (振る舞いの部分属性・データ)) のセット 相互関係記述 = (相互関連記述) + (相互作用記述)
(c) 分析から再構成への遷移モデリング段階	
	(プロセッサ, カプセル化機構, 情報隠蔽機構, メッセージパッシング機構 (インターフェイス機構とメソッド起動 (オブジェクト活性化) 機構) を 新規にモデル化して追加
(d) OB 再構成 (設計) モデリング段階	
	最小単位 OB 再構成オブジェクト = (OB オブジェクト名) + (サブオブジェクト群) + (相互関係機構) サブオブジェクト = ((部分属性・データ構造) + (メソッド)) 記述のセット 相互関係機構 = (相互関係記述) + (メッセージパッシング機構) 相互関係記述 = (相互関連記述) + (相互作用記述)
(e) OB 再現 (実装) モデリング段階	
	最小単位 OB 実装オブジェクト = (OB オブジェクト名) + (サブオブジェクト群) + (相互関係リンクスロット) + (ポートスロット) + (プロセス) + (OB オブジェクト駆動機構) サブオブジェクト = ((データスロット) + (メソッドスロット)) 記述のセット 相互関係リンク = (集約, 関連, 必要なら委譲や継承) リンク + (相互作用) リンク

相似性

モデリングに基づく変換前後で対応する記述要素の組が一対一対応の関連を持つこと。

一貫相似性

モデル記述の最初 (概念モデル) から最後 (実装モデル) までの多くのモデリング段階において, その中のどの任意の複数のモデリング段階のモデル記述を採り上げて比較・評価しても, 常に必ず相似性を見出せる特性を指す。

一貫相似性成立の判定法

任意の複数の (従って全ての) モデリング段階 (ステップ) のモデル記述において, その対応すべき記述形式上の構造を比較し, その全ての対応すべき構成要素の個々に対して, 各々一対一対応の関連を同定 (Identify) できること。

一貫相似性の高さ (精度)

一対一対応関連が同定されたときの両構成要素各々の記述の対応の良さ (記述落差の少なさ) が両モデル記述の (一貫) 相似性の高さ, つまりモデル相似精度 (付録の用語定義参照) である。ユーザ判断事項。

3.2 一貫相似性の検証方法と手順

本節では前節で提案した一貫相似性検証の手順を, 明示化された形式的な枠組みとして述べる。

OB オブジェクトモデルの表 3,4 及び一貫相似性モデリング過程の図 2 という枠組みに従って対象世界をドメインユーザがモデル化すれば, オブジェクト指向の強い構造記述性を持つ記述が必然的に為される。故にその記述された構成要素個々の複数 (全) モデリング段階間での縦の一対一対応関連が容易に指摘 (同定) し, 記述の比較評価を行う。

表 3 における構成要素個々に対する全モデリング段階にわたる一対一対応の同定がなされればその抽象度・詳細度におけるモデル記述の一貫相似性の成立が検証され保証できる。

4. モデリング作業過程の追跡と検証

4.1 概念モデリング段階 (表 3 の (a))

このモデリング段階では対象世界の概念単位としての “もの” モデルを初めて「明示的に」言語で表現・

表4 OO対象分析記述モデル

*

1. "もの"名
2. 属性
2.1 単独属性
属性名, 属性種, 属性値, 参照作用名
2.2 相互関係属性
2.2.1 相互関連属性(構造属性)
(1) 汎化・特化: 上位(汎化) "もの"名, 下位(特化) "もの"名リスト
(2) 集約: 上位 "もの"名, 下位 "もの"名リスト
(3) 集合: 所属 "もの"名リスト
(4) 一般関連: 相互関連属性名, 相互関連相手 "もの"名リスト
2.2.2 相互作用属性(振舞い属性)
相互作用抽象名, 相互作用相手 "もの"名, 相互関係 "もの"名リスト
2.3 制約属性
"もの"と属性には, アクセス制約・条件 振舞いには, 起動制約・条件
3. 振舞い
3.1 内部振舞い
内部振舞い名, 振舞い記述
3.2 相互作用振舞い
相互作用抽象名, 振舞い記述, 相互関係相手 "もの"名

記述する。"もの"モデル記述の枠組みは人間のモデリング思考体系そのものであり、記述言語は通常は自然日本語であるので、"もの"モデルとしてのモデリングや記述方法なども、強くかつ豊富な表現力を用いて対象世界と十分高い相似性を持つ記述を起こすことが可能である。

4.2 論理モデリング段階(表3の(b))

概念モデルは曖昧な部分や記述漏れ・不足を残した記述になることが屢々ある。この曖昧さや記述漏れを少なくするため、形式化・論理化され、数学や物理学、論理学等の各専門分野(ドメイン)の厳密な言語・用語体系や概念体系(方程式を用いた記述等も含む)に写像して、対象世界を厳密に捉えて過不足無く記述するモデリング段階である。この段階の"もの"モデルを他と区別して"シング"モデルと呼ぶことにする。

4.3 遷移モデリング段階(表3(c))

分析モデリング過程から再構成モデリング過程へ移る途中の遷移モデリング段階においてモデル記述の様

式や構成、記述すべき内容とその記述言語に対してモデリング過程で最も大きな変化や落差が生じることが良く知られている。その原因は振る舞いを生じさせる根源が自然からコンピュータシステムに替わることにある。それ故に例えば、"もの、シング"モデル記述から変換されただけのオブジェクト記述を"オブジェクトとして"起動・駆動させるためのオブジェクト駆動機構、等を新規に記述する必要がある。このためには以下の三条件が満たされなければならない。^{4)~6)}

- 1 カプセル化機構(サブオブジェクト, 最小単位及び集約オブジェクト単位の構成境界を明示的に設定)
- 2 情報隠蔽機構(シングモデルと相似な相互作用を実現するアクセス制限機構)
- 3 メッセージパッシング機構(相互作用の伝達機構)
 - a インターフェイス機能(通信機能)
 - b メソッド起動機能(相互作用機能)

である。上記の1の機構は本論文ではUNIXファイルシステム(以降、UFSと略記)の特徴を利用してユーザ自身が実装する。上記の2と3の機構はOB機構として我々が新たに構築・提供したオブジェクト駆動機構である^{3)~6)}。このOB機構はOBオブジェクトモデルとセットで用いるように拡張設計された機構であるのでOBモデルとの非整合性は起こらない。

以上の1, 2, 3の三つの機構を実現することで遷移モデリング段階における相似性は保持される。

4.4 再構成モデリング段階(表3の(d))

この段階はモデリングの観点から見れば再構成モデリング段階、ソフトウェア工学の観点から見れば設計段階に当たる^{10),12)}。表3の(b)から(d)ではモデル名称がシングからオブジェクトへと変わったこと、オブジェクト駆動のためのメッセージパッシング機構の記述が相互作用実現のモデル化されたものとして加わったことであり、相似性保持の障害は無いことが結論できる。このモデリング段階のOBモデルの記述力やオブジェクト指向との整合性は以前から採用されてきていて充分検証済みである^{1),2)}。

もう一点重要な点は、本論文ではドメインユーザの強い要請の実現を目標にしているため、現状の駆動環境(UNIX)上で手続き型言語を用いて前々節の三条件及びその他の必要条件を実現するOB機構(=OBオブジェクト駆動機構)を新規に構築した^{3)~6)}。

4.5 OB機構

以前の論文¹⁾で指摘したように、UNIXシステムはそのUFSにオブジェクトの構造部分を、UNIXプロセスにその駆動部分に対応(map)させればUNIX自体が一種のオブジェクトモデルを持っていると見なす

表 5 UNIX システムとオブジェクトシステムの対応関係

UNIX システム	オブジェクトシステム
ディレクトリ	集約階層オブジェクト
ディレクトリ	カプセル化機構
ディレクトリ名	オブジェクト名
下位ディレクトリ名	集約オブジェクト要素名
上位ディレクトリ (...)	上位集約オブジェクト
通常 UNIX ファイル	属性 (相互関係含む)
シンボリックリンク	相互関連
実行可能ファイル	メソッド
UNIX プロセス	オブジェクトの振る舞い オブジェクト間相互作用
プロセス間通信	メッセージパッシング
実行可能ファイルの 起動	メッセージパッシングに依 るメソッド起動 (振る舞い)
OB 機構	オブジェクトシステム
ファイルへのプロ セスのアクセス権限	情報隠蔽機構
委譲リンク	委譲機構
セルフディレクトリ	自分自身の同定機構

ことが出来る。そこでこの既存のオブジェクトモデルの上に更にいくつかの新規な見なし方を加え、かつ新規な機構を実装をすることで、UNIX システムを OB オブジェクトシステムへと再構成することが出来る。その新規な見なし方を表 5 の上段部分に示し、新規に実装された OB 機構を同じく下段に示す。この表 5 の対応関係の見なし方と実装機構を含めて OB 機構としても良い。詳細な OB 機構の説明は既出の論文^{(4), (6)}を参照されたい。

OB 機構では遷移モデリング段階において見いだされた新規に実装すべき 3 つの機構の内、カプセル化機構はディレクトリ構造をそのまま利用してカプセル化すべき単位各々をユーザ自身が実装して格納することで実現し、情報隠蔽機構は OB 機構で新規実装されており、メッセージパッシングは UNIX のプロセス間通信と実行可能ファイルの起動機能で実現された。この結果、ドメインユーザの強い要望である任意選択言語 (主に現行の手続き型言語) を用いてオブジェクトシステムを構築・駆動させたいという要求が実現され、かつ、遷移モデリング段階におけるオブジェクトシステムの相似性保持も実現された。

4.6 再現 (実装) モデリング段階 (表 3 の (e))

表 3 の (d) と (e) の相似性については、表 3 の (d) に表 5 の変換を施したものがちょうど表 3 の (e) に一対一対応していることは明確であり、従って、その相似性は成立しており、実装モデリング段階においても設計モデリング段階との高い相似性を保ったモデリング (変換) が実現可能であることが示された。

以上本章の結論として、OB オブジェクトモデル (表

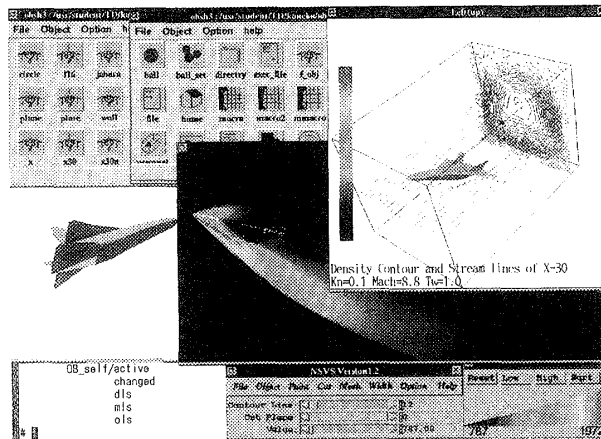


図 3 物理風洞の再現シミュレーション (数値風洞) の駆動例

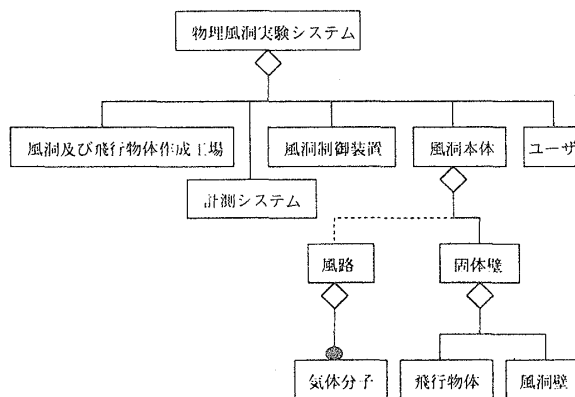


図 4 物理風洞の分析モデリング過程内の論理モデル記述

3.4) の適用と一貫相似性モデリング過程 (図 2) の追跡・評価と考察により、ドメインユーザの充分な記述 (実装) 力を前提にすれば、高い一貫相似性を持つオブジェクトシミュレーションシステムが実現できることを検証できたと言えよう。

5. 実装具体例

我々の研究グループ内ではオブジェクト指向一貫相似性モデリング過程の方法に基づいて既に物理現象 (流れ) を幾つか OB オブジェクトモデルで記述し、実シミュレーションシステムの具体的な評価が実施されてきた^{(1)~(6)}。ここでは、実装具体例の 1 つ⁽⁵⁾を挙げる。なお実装言語は手続き型である C 言語とオブジェクト指向言語 C++ の両方を併用し、混合実装とした。

対象世界である風洞 (物理風洞) とは空気のある一定の条件下で人工的に流す装置である。その内部に飛行物体などを置き、空気の乱れの様子や、飛行物体の表面の圧力分布を計測したりする。物理風洞に近いイ

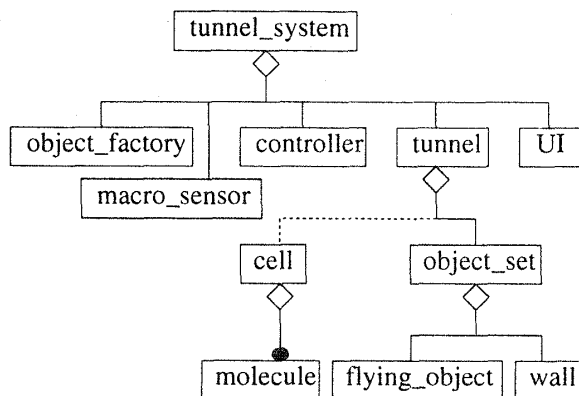


図5 DSMC 数値風洞の再構成モデル記述

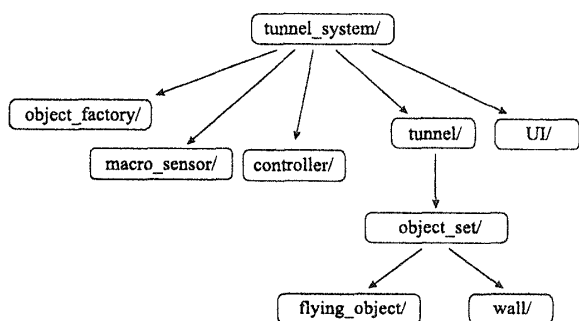


図6 DSMC 数値風洞の実装モデル記述

メージ例としては、図3が挙げられる。

以上のような物理風洞をオブジェクト指向に基づき分析して記述した図（OMT表記）が図4である。図5は図4を変換しモンテカルロ法（DSMC法）を用いて記述した再構成モデル記述の最上位の集約階層オブジェクト図（OMT表記）である。分析（論理）モデリング過程の対応する“もの、シング”構造図である図4と比較すると、図5のobject_setを固体壁とし、cellを離散化された風路等と名称を変えれば、全く同じ構造の、つまり相似性の高い、論理モデル図が出来上がる。

図6は図5の再構成モデル記述を表5に従い、UNIX上へ実装したモデル記述である。図3にDSMC数値風洞の再現シミュレーションの駆動例を示す。

6. 評価と考察

6.1 一貫相似性モデリング過程の方法の検証

実用に利用した結果から、全モデリング段階に対して適用される図2と表3、4の組み合わせはドメインユーザとコンピュータシステムの専門家との両方に充分理解でき、緩やかな制約しかないOBモデルを用い、全モデリング段階において、協調して議論しつつ進める上での良好な共通概念を与える共通基盤として

利用出来、オブジェクト（シミュレーション）システムの開発の実作業上非常に有用である事実が判明した。

6.2 一貫相似性の検証方法の考察

（一貫）相似性の検証とその相似精度の評価の二点の具体的な定義と評価を行うのは各ドメインユーザ自身が責任分担すべき問題である。従って一貫相似性モデリング過程の方法としては、第3章で述べた検証手順と図2と表3.4で代表される検証の枠組みの組み合わせだけで一貫相似性の検証手順と枠組みを充分に提供しているか否かを考察すればよい。

この点については既に第4章において、第3章で提案した方法に従って全作業過程での相似性（即ち一貫相似性）同定の模擬作業を行い、その成立を検証した。そして、第4章の作業過程と同じ作業を第5章の典型的実装例において実施し、その途中過程及び、シミュレーション駆動での直接比較評価から、相似性の高いシミュレーションの実現を確認している。この結果は第3章の一貫相似性の検証方法に関する提案が正当かつ合理的な定義と判定基準であることを裏付けていると判断して良い。

以上から本論文第3章で提案した一貫相似性の評価と検証手順とその方法の枠組みは、整合性の高く評価・検証に適した合理的な方法であることが分かった。

7. 結論、今後の課題

対象世界に対する相似性の高い再現シミュレーションを実現するには一貫相似性モデリング過程の方法を実現することが最良であるとの考えを提示し、オブジェクト指向モデルとしてOBオブジェクトモデルを考案・提案し、一貫相似性モデリング過程の方法と一貫相似性の成立の検証方法を提案・適用・実装した。以上から以下の結論が得られた。

- (1) 一貫相似性モデリング過程の方法の成立
- (2) 一貫相似性の検証方法の確立
- (3) 手続き型言語でオブジェクトシステムを実現
- (4) オブジェクト指向システムの実用開発技法

今後の課題としては一貫相似性モデリング過程の方法の適用分野の拡張と実装例の蓄積、ICS-OBモデルとOB機構を用いてプログラム実装までを支援するための開発支援環境の開発等が挙げられる。

参考文献

- 1) 島山 正行, 金子 勇, 渡辺 正雄, 物理世界のオブジェクト指向分析・設計・実装・シミュレーション駆動の実現, オブジェクト指向'96 シンポジウム

- ム, 朝倉書店刊, 147-154 (1996).
- 2) Hatakeyama, M., Watanabe, M., Suzuki, T., Object-oriented Fluid Flow Simulation System, International Journal of Computers and Fluids, 27-5-6, pp.581-597 (1998).
 - 3) 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, 情報処理学会研究報告, 95-49, pp.33-44 (1994).
 - 4) 畠山 正行, 金子 勇, オブジェクトシミュレーションシステムへのUNIXシステムの再構成, 情報処理学会論文誌, 39-7, pp.2060-2073 (1998).
 - 5) 金子 勇, 畠山 正行, オブジェクト指向シミュレーションの実現とその実行環境, 日本シミュレーション学会誌, 17-4, pp.295-309 (1998).
 - 6) 金子 勇, 畠山 正行, プロトタイプベースオブジェクトファイルシステム, 情報処理学会論文誌, 39-9, pp.2671-2683 (1998).
 - 7) 磯田定宏, 実世界モデル化有害論 オブジェクト指向モデル化技法の解明, 電子情報通信学会論文誌, Vol.J83-D-I, No.9, pp.946-959, Sept. 2000).
 - 8) 畠山 正行, 加藤木和夫, 石井義之, オブジェクト指向記述日本語 OODJ とその記述環境, 情報処理学会論文誌, Vol.41, No.9, pp.2567-2581, (2000).
 - 9) 米沢 明憲, 柴山 悦哉, モデルと表現, 第1章 第2章, 岩波書店 (1992).
 - 10) 所真理雄ほか(編), オブジェクト指向コンピューティング, 第1章, 岩波書店 (1993).
 - 11) Wegner, P., はやわかりオブジェクト指向, 共立出版 (1992). 尾内理紀夫訳 (Concepts and Paradigms of Object-oriented Programming, key note Lecture in OOPSLA '89).
 - 12) Rumbaugh, J. et. al., Object-Oriented Modeling and Design, Prentice Hall (1991).
 - 13) 加藤木 和夫, 畠山 正行, オブジェクト指向日本語一貫プログラミング環境, 情報処理学会論文誌, 40-7, pp.3016-3030 (1999).
 - 14) Ungar, D., Smith, R. B., Self: The Power of Simplicity, Proceedings of the OOPSLA '87, 22-12 (1987).
 - 15) 日本機械学会, 日本機械学会第6回計算力学講演会講演論文集, pp.35-60 (1993).
 - 16) Computational Mechanics '95, Proceedings of the ICES'95, 1, pp.14-68 (1995).
 - 17) 日本計算工学会誌「計算工学」, 1-2, pp.5-30 (1996).
 - 18) Dahl, O. J., Myrhaug, B., Nygaard, K.(eds.), Simula 67 Common Base Language, Norwegian Computing Center (1968).
 - 19) Zeigler, B. P., Object-Oriented Simulation with Hierarchical, Modular Models, ACADEMIC PRESS, INC. (1990).
 - 20) 畠山 正行, 金子 勇, オブジェクト指向に基づく数値風洞シミュレーション 高精度なモデリン

グと柔らかな計算システムの実現例 (小特集「ソフトコンピューティング」記事), シミュレーション学会誌, 16-3, pp.181-188 (1997). 及び表紙カラー画面.

8. 用語の定義

オブジェクト (シミュレーション) システム: オブジェクトを, モデル定義・分析・実装・駆動の基本単位としたシステムを指す広義の用語.

オブジェクト駆動機構: 本論文では, UNIX 上の実装記述オブジェクトを "オブジェクト (システム) として" 起動・駆動させるための機構. 駆動 (実行) 環境とも呼ぶ. 文献^{4)~6)}参照.

広義のオブジェクト指向

[1] クラスベース (狭義のオブジェクト指向. クラス, クラス階層, 継承, メッセージパッシングを必須要素とし, 最もよく利用されている^{11),12)}), プロトタイプベース, オブジェクトベースの三種のオブジェクト指向の総称.

[2] 概念モデリング段階に始まり再現 (シミュレーション) モデリング段階に至るまでの全てのモデリング段階に適用できるオブジェクト指向.

相互関連: 複数要素間の静的な (時間の経過があっても変化しない固定的な) 関わり.

相互作用: 複数要素間の動的な (時間の経過に従って変化する) 関わり.

相互関係: 相互関連と相互作用を二つのサブセットとする統合化概念. 属性の一種.

精度: 記述精度と相似精度の二種類が使われている. オブジェクト記述の質の良さ (精度) を, どの程度の「詳細さ」と「正確さ・適正さ」を持つかの両評価要素の積で表現する, とした. 数値的な精度 (桁数, 次数) とは異なる.

構造: 複数の要素とそれらの間の相互関連記述の組.

強い構造記述性: 対象世界の構造の側面に着目したモデル化記述をし, それが対象世界のモデル記述全体で特に強く支配的である, という特性.