

一貫モデリング過程論に基づく物理世界のオブジェクトモデル

畠山 正行

茨城大学情報工学科

〒316-0033 茨城県日立市中成沢町4-12-1

e-mail:masayuki@cis.ibaraki.ac.jp

概要

対象世界の再現シミュレーションにおいては、再現シミュレーション世界との間の構造、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、両世界の間を結ぶ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、それを実現する UNIX 上の機構を提案する。そのため、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 モデル) を提案し、それを使って一貫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た。更には、現状のシミュレーションユーザの手続き型の常用言語や利用環境、既開発プログラムや蓄積ライブラリ等に関する要請を入れて、OB 機構と呼ぶ機構を考案・実装する事で実現した。以上で提案した一貫モデリング過程の概念、OB モデル、OB 機構、それらの組み合わせの整合性、一貫性、相似性実現の高さ、等を再点検・考察し、統合された一貫 OB モデルを提唱した。

Object Oriented Model for Real World based on Integrally Consistent Modeling Processes

Masayuki Hatakeyama

Department of Computer and Information Sciences,

Ibaraki University, 4-12-1, Nakanarusawa, Hitachi-city, 316-0033 Japan.

Abstract

We consider that one of the best methodology to develop the simulation system with the "high similarity" is to realize the modeling processes based on an integrally consistent modeling paradigm throughout from the target world up to the re-constructed simulation world. To realize this purpose, the object-oriented modeling and programming paradigm is selected and an object-model called OB model has been proposed. We have also developed the object-oriented simulation mechanism for the simulation (domain) users who use the proceduretype programming languages, programs and the UNIX execution environment. To attain the aim, we have devised and realized an object-oriented simulation mechanism called OB mechanism on the UNIX file system. We finally propose the integrally consistent OB model that helps the domain users to apply and develop a new kind object (simulation) system based on the integrally consistent modeling paradigm and its procedures.

1 はじめに

モデリングの対象世界の典型例として、自然世界或いは工学の対象となる世界（例えば、流れ、構造物）の再現シミュレーションには、対象となった世界と再現された世界との構造や機構、相互作用・振る舞い等々の側面での両世界の相似性の高さが最も重要である。

対象世界のモデル化は“もの”を単位とし、“もの”と“もの”との相互作用や振る舞いにより変遷して行く。とすれば、その最終段階である再現シミュレーションも相似なモデル化単位である“仮想的なもの”を駆動単位とし、“仮想的なもの”と“仮想的なもの”との相互作用や振る舞いで変遷して行くのが理想的ではないか、と考えて自然であろう。

そう考えると、その両者の中間の全てのモデリング過程内でも、モデル化単位は認識モデルと同等或いは相似な“シングモデル”であり、全モデリング作業過程内で一貫して共通なモデリングパラダイムを使う方がよい結果を生むであろう。この連続したモデリングの考えを我々は、「一貫モデリング過程」[1]と呼んでおり、相似性の高い再現シミュレーション実現の基盤方針としている。

そこでこの様な一貫モデリング過程を実現し、相似性の高い一貫した再現シミュレーション世界を構築するため、我々はまず、モデリングパラダイムとしてオブジェクト指向を採用し、OBモデルと呼ばれる一貫モデリング過程用に工夫されたあるオブジェクトモデルを考案した[2]。

しかし、現在の再現シミュレーションの殆ど全ては手続き型の言語（Fortran, C）を用い、莫大な蓄積ライブラリを利用することで、行われている。そこで、手続き型言語で記述されたシミュレーションシステム（シミュレーションプログラム群を以下ではこう呼ぶ）をUNIX実行環境上でオブジェクトシステムとして駆動させられるように、オブジェクトモデル（OBモデル）を前提とした新規な機構（OB機構と呼ぶ）を考案し実現した[2, 3, 4, 5]。

そこで本論文では、まずOBモデルとオブジェクト指向言語（必要ならオブジェクト実行環境も）を用いて、一貫モデリング過程が実現できることを検討する。その後、シミュレーションユーザの要請である手続き型の言語の利用の要請などを容れた上での一貫モデリング過程に基づいたオブジェクトシ

表 1: “もの”モデルとオブジェクトの対応関係

Table 1 Correspondence between “Thing” model and Object

| モデル化された“もの、Thing” | オブジェクト |
|-------------------|---------|
| データ | 属性・データ |
| “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（アクセスの）可否 | 情報隠蔽 |
| “もの”の間の相互作用や様式 | メッセージ通信 |

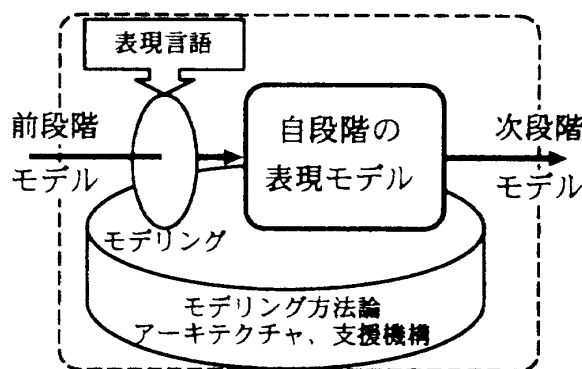


図 1: モデリング段階の基本構成

Fig. 1 Primary constitution of a modeling stage

ステムの実現が可能であるかの検討を行う。これを検証したあと、改めて一貫モデリング過程の概念体系、これを実現するために再検討され検証済みの一貫モデリング過程用のOBモデル、それを實現するOB機構、の3要素を統合した方法を拡張モデル（一貫OBモデル）という形で改めて提案する。

2 一貫モデリング過程

モデリング作業の基本構成を図1に、一貫モデリング過程全体を図2に示す。

2.1 一貫モデリング過程実現の要請

現状の再現シミュレーション世界の記述は手続き型言語を用いたものが主流であり、オブジェクト指向に基づくシミュレーションは未だ数少ない。手続き型言語やプログラムを用いた記述方式での一貫モデリング過程を実現しようと狙う理由は、シミュレーション（ドメイン）ユーザからの強い要請である[3]。

1. 現駆動環境（UNIX, 95/NT）での互換性

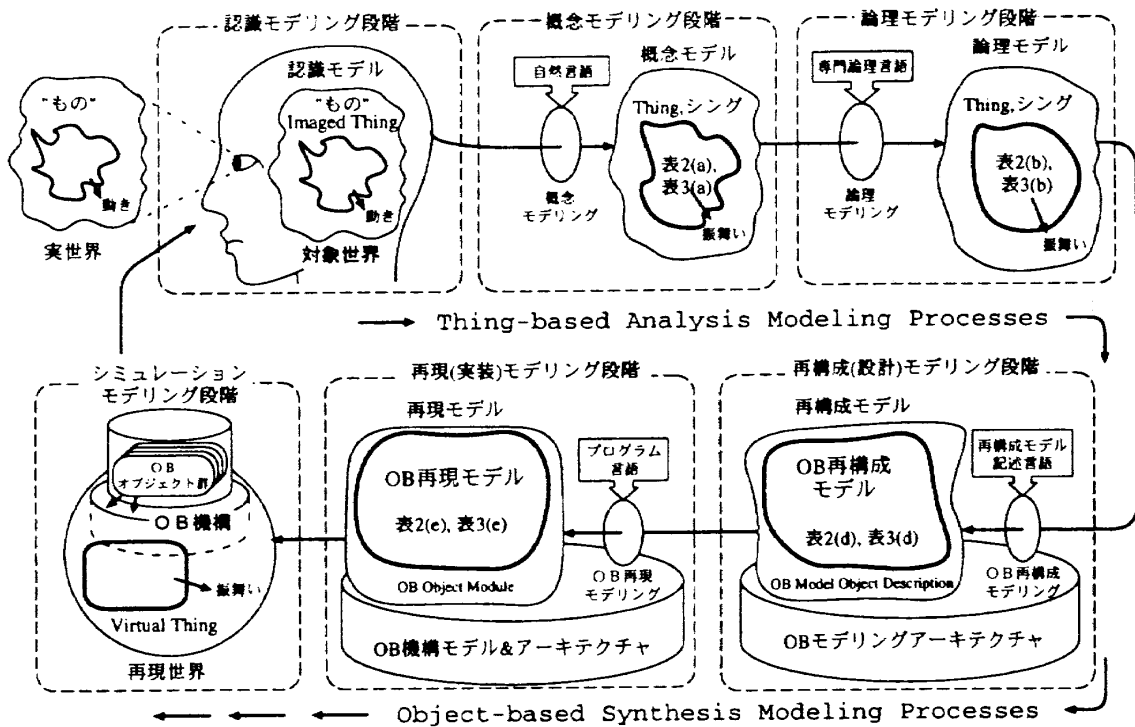


図 2: 一貫モデリング過程 (OB モデルを含む)

Fig. 2 Integrally consistent modeling processes (including OB model)

2. 記述言語の任意性
3. 既存プログラムや蓄積ライブラリの再利用
4. 現在の高い処理実行効率の維持.

図 1 においてモデリング作業は一般に前段階 (或いは更に細かく前ステップ) モデルを自段階に採り入れて分析・解釈し, その結果から自段階の表現モデルへと変換する. モデリング作業の要素はモデリング方法と表現言語 (表現系) の二つの要素に大別できる. 主用すべき言語は OODJ と我々が呼んでいる日本語系の一貫日本語記述言語である [6].

2.2 OB モデル

OB モデルについては既に発表もしており [2], その構成そのものは一般的に見られるものである [7, 8]. オブジェクト指向の分類 [8] から言えばオブジェクトベースに近い. モデルの名前付けもそこに所以する. しかし, 通常のオブジェクトベースとの最も大きな違いは生成されたオブジェクトに対する規定しかならないモデルである点である. 即ち, クラスベース [7] でいうインスタンス生成後, プロトタイプベース [8] でいうオブジェクトのコピー生成後, に対するモデル定義・規定・制約が定められている

だけである.

このモデル規定は, 別の側面から言えば, 実装に依ってはクラスベース (狭義のオブジェクト指向), プロトタイプベースの両方式のオブジェクト指向に基づくオブジェクトも実装や起動・駆動に障害がない限りにおいて混合して利用して構わない事を意味する [3, 6].

2.3 分析モデリング過程

認識モデリング段階は, 実世界をまず最初に人間の脳が認識 (認知) する段階を指す (図 2 参照). “もの, シング” とは従って, 人間の脳が, 認識し, 概念として外界に出てきたもの全てを指す. 認識モデルは予め, 各人間が必ず持っているものと仮定する. この段階のモデル (認識モデル, Imaged Thing) が以降の一貫モデリング過程における「対象世界」になる.

概念モデリング段階では概念としてのシングモデル (図 2) の枠組みや概念体系の規範を念頭において, 対象世界 (認識モデル) を初めて「明示的に」モデル化して言語に表現・記述して行く. シングモデルの枠組みや概念規範等は人間のモデリング体系

表 2: 最小単位 OB モデル

Table 2 Minimum unit OB model

| モデリング | モデル構成 |
|--------------------------|--|
| (a) “もの” ベース 概念モデリング | 最小単位概念シングモデル = (抽象シング名) + (振る舞い記述群) + (内外相互関係記述) 内外相互関係記述 = (相互関連 (静的) 記述) + (相互作用 (動的) 記述) |
| (b) シングベース 論理モデリング | 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) + (内外相互関係記述) サブシング = ((振る舞いの論理的記述 (式など)) + (振る舞いの部分属性・データ)) 内外相互関係記述 = (相互関連 (静的) 記述) + (相互作用 (動的) 記述) |
| (c) 遷移 分析から再構成へ | (プロセッサ, インタフェース機構, カプセル化機能, 情報隠蔽機能, メッセージパッシング機構, 起動・活性化機能) を新規に追加 Implicit (暗示的) 特性の構成から Explicit (明示的) 特性の構成への遷移モデリング |
| (d) OB 再構成 (設計) モデリング | 最小単位 OB オブジェクト = (OB オブジェクト名) + (サブオブジェクト群) + (内外相互関係機構) 内外相互関係機構 = (内外相互関係記述) + (OB インタフェース機構&活性化機構) 内外相互関係記述 = (相互関連 (静的) 記述) + (相互作用 (動的) 記述) サブオブジェクト = ((部分属性・データ構造) + (メソッド)) |
| (e) OB 再現 (実装) モデリング | 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 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)) 相互関係リンク = (集約, 関連, 委譲) リンク + (相互作用) リンク |

表 3: 集約階層構造 OB モデル

Table 3 Aggregated, hierarchically structured OB model

| モデリング | モデル構成 |
|---------|--|
| (a) 概念 | 集約階層構造シングモデル = (抽象シング名) + (振る舞い記述群) + (下階層のシング群) + (内外相互関係記述) |
| (b) 論理 | 集約階層構造シング論理モデル = (抽象シング名) + (サブシング記述群) + (下階層のシング群) + (内外相互関係記述) |
| (c) 遷移 | |
| (d) 再構成 | 集約階層構造 OB オブジェクト = (OB オブジェクト名) + (サブオブジェクト群) + (内外相互関係機構) + (下階層の OB オブジェクト群) |
| (e) 実装 | 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) + (下階層の OB 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) |

そのものである。その表現言語系は通常は自然言語 (日本語) である。

概念モデルは曖昧な部分を論理モデリング段階においては、論理的な専門領域用語を用いた表現体系や概念体系 (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るモデリング段階である。形式化・論理化され、数学や物理学、論理学等の形式記述がされることも多い。

2.4 再構成モデリング過程

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システム設計 (基本設計及び詳細設計) 段階に当たる [7, 9, 10]。従って、再構成以下のモデリング段階に関しては、通常のオブジェクト指向ソフトウェア開発方法論が、多少の手直しを施すことによってそのまま利用できる。再構成モデリング方法論には一般にオブジェクト指向モデリング方法論と呼ばれるものを充てることが出来る [7, 9]。

図 3 の最小単位 OB オブジェクトは、その単位

で“シング”モデルの最小単位と同等である。従って、その内側 (内部構造) はオブジェクトではなく部分部品を構成要素とし、何等かの構造を形成している。

3 議論と考察

3.1 OB モデルの評価

OB モデルの意義は表 2 および表 3 において広義のオブジェクト指向の一貫モデリング過程用のモデルとして使える事が分かったことと、このモデルを使うことで一貫モデリング過程が実際に成立することを検証した (出来た) ことにある。その意味では充分有効なモデルであった。またこのモデルのもう 1 つの意義は、ドメインユーザがこのモデルを任意の言語でそのまま利用・実装すれば一貫モデリング過程に基づく再現シミュレーションを実際に可能に出来るような実践的なモデルである点である。メソッド毎にプログラミング言語を変える混合実装等、も許容している。即ちある程度は UNIX 上でオブジェクト指向システム構築における言語の任意

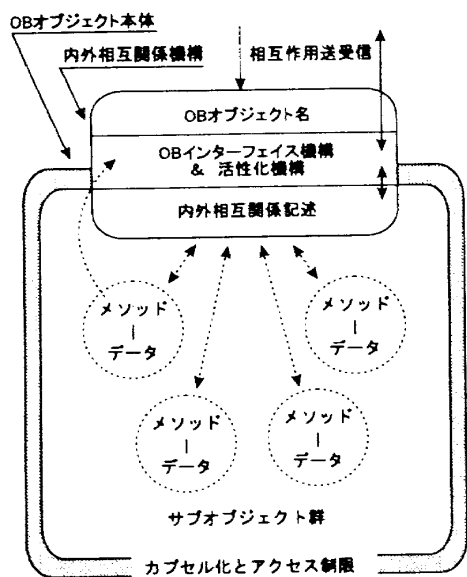


図 3: 最小単位 OB モデルの構成

Fig. 3 Minimum unit OB model constitution

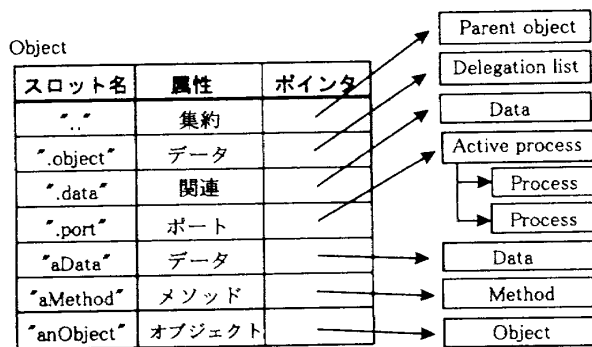


図 4: 実装モデリング段階の OB オブジェクトの構造モデル

Fig. 4 Structure model of OB object in implementation modeling stage

選択が実現出来るようになった、と言えよう。

3.2 一貫モデリング過程実現の考察

表 2, 表 3 を見ると、まずモデルおよびモデリング方法については、OB 機構を前提にした OB モデルの構成概念がそのモデリング段階の相似な構成枠組みに応じて、OB モデル単位での抽出と近隣との相互関係の修復記述をし、論理的な整合性をとっていている事がわかる。遷移モデリングで計算機世界での再現に必要な要素を明示化して採り入れている。OB 再構成モデリング、OB 実装モデリングの各段階ではモデル記述の枠組みに従って、変換・実装すれば相似な再現性が崩れる要因は見つから

ない。

これらから OB モデルの表 2, 表 3 が実装できれば、概ね一貫モデリング作業が実現され、一貫モデリングに基づく再現が実現しているといってもよい。

3.3 ライフサイクルモデルとの比較検討

この様な一貫モデリング過程の考えと類似の既存の考えは存在している。オブジェクト指向ソフトウェア工学のソフトウェアライフサイクルを形成する開発プロセスがそれである [7, 9, 10]。一貫モデリング過程はモデリングの順序関係やスタイルを規定していない。既存のオブジェクト指向ソフトウェア工学の成果を可能な限り採り入れることで、より高精度な再現シミュレーションが実現できよう。

3.4 一貫 OB モデルの提唱

両 OB 機構 (UFS-OB 機構 [4], PBOFS-OB 機構 [5])、OB モデル、一貫モデリング過程の概念方法論と枠組み、UNIX 上の (既存) 駆動機構、を総合したものを「一貫 OB モデル」と呼んで提唱したい。提案の具体的内容は、表 2, 表 3 の OB モデル、OB 機構、一貫 OB モデリング過程のモデリング概念体系、から成る。

4 結論

従来提案・実装していた一貫モデリング過程と OB 機構の上に、OB モデルを拡張して統合化し、一貫 OB モデルとして提案した。このモデルを用いて一貫したモデリングパラダイムに基づくモデル化と再現オブジェクトシステムの構築を実現出来た事を実証した。同時に構築オブジェクトシステムが相似性の高いシステムになることも示した。

これらの三要素統合一体化モデルとしての一貫 OB モデルにより、本論文当初に目標と掲げたユーザの要求記述を満たす再現シミュレーションを実現するオブジェクトシステムを提唱できた。

参考文献

[1] 畠山正行: 高度なシミュレーションのためのオブジェクトベース一貫モデリング過程論とその駆動支援環境, 情報処理学会, 第 1 回数理モ

- デル化と問題解決研究会報告, Vol. 95, No. 44, pp. 33-40 (1995).
- [2] 畠山正行: 物理世界記述用オブジェクトモデルとその駆動機構, 情報処理学会, 第13回数理モデル化と問題解決研究会報告, Vol. 97, No. 41, pp. 1-6 (1995).
- [3] 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, Vol. 39, No. 7 (1998).
- [4] 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, Vol. 94, No. 49, pp. 33-44 (1994).
- [5] 金子勇, 畠山正行: Prototype-based Object File System, 日本ソフトウェア科学会オブジェクト指向コンピューティング研究会 (1997). <http://www.is.titech.ac.jp/ooc/wooc97/Proceedings/>.
- [6] 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第118回ソフトウェア工学研究会報告, Vol. 98, No. 20, pp. 15-20 (1998).
- [7] Rumbaugh, J. et al.: *Object Oriented Modeling and Design*, Prentice Hall (1991).
- [8] 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).
- [9] 青木淳: オブジェクト指向システム分析設計入門, (株)ソフトリサーチセンター (1993).
- [10] 本位田真一, 青山幹雄, 深澤良彰, 中谷多哉子: オブジェクト指向システム分析・設計, 共立出版 (1995).