

## 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実現 とその検証方法の提案

—科学技術計算/解析/設計/シミュレーション等の応用ドメインの専門家ユーザが使えるオブジェクト指向技術—

畠山 正行†

**要約** 対象世界の忠実な再現シミュレーションを実現するには、対象世界との高い相似性を保障するプログラムの開発法が必須である。そのため両世界間の複数のモデリング段階に一貫して同じオブジェクト指向 (以降 OO と略) を採用し、一貫相似性の新規な定義とその形式的検証方法、OO で一貫して使えるオブジェクトモデルを新規に導入し、OO 一貫相似性モデリング過程の実現の枠組みを設計・提案した。その上で想定ユーザである科学技術・工学分野の専門家 (ドメインユーザ, DU) のために具体的なモデリング方法や検証方法を提案した。この方法に対する実用規模の具体例 (水の大気循環) の分析・設計・実装を比較評価の結果、一貫相似性の検証方法が有効に働き、一貫相似性の高いモデリングが実現できたことを確認した。今後この方法を実施する DU のモデリング作業の支援環境の開発や実用化に向けた研究を行う予定である。

### A Proposal of Object-oriented, Integrally Consistent and Similar Modeling Processes and its Verification Scheme

MASAYUKI HATAKEYAMA†

**Abstract** To realize the world reemerging simulation with “high similarity”, we have proposed an Integrally Consistent and Similar Processes throughout from the real world up to the implemented simulation world. To realize this purpose, the Object-oriented (hereafter, OO) paradigm has been adopted and also an OO modeling reference model has been constituted. To verify the similarity between the real world and the simulation world, the concept “the similarity among the correspondent elements” has been proposed and its formal verification scheme has been developed. The results are as follows; (1)The simulation method with high similarity has been established throughout the OO modeling processes. (2)The verification scheme has been verified using some typical example of the atmospheric great circulations of water. Finally, we have concluded the validity and the usefulness of our proposed OO modeling processes for the simulation users.

#### 1. はじめに

本研究の狙いはサブタイトルにあるような計算機を利用して専門的な仕事を遂行している応用分野を対象とし、計算機内に実世界との高い相似性を持った再現シミュレーションを実現する方法を提案することにある。さらにはその相似性の高さを客観的に裏付ける形式的な検証方法を提案することにある。

著者は以前、オブジェクト指向 (以降、OO と略) に基づく分析モデリングの方法について提案した<sup>1)</sup>。モデリングとは対象世界についての着目・抽出・名前付けの一連の作業を指すものとする。本論文ではこれを

出発点として設計段階や実装段階のモデリングについて拡張し、分析から再現シミュレーションまでを一貫したモデリングとして扱い、必要条件を形式的に検証できる方法を提案する。

#### 2. DU と一貫相似性モデリング過程

##### DU とは

本論文のターゲットユーザは、サブタイトルにもある様に計算機を利用する専門分野のドメインユーザ (Domain User, DU と略) である。DU の特徴を表 1 に纏めた。

##### 再現シミュレーションの構成

DU が要求する “対象世界を忠実に再現したシミュレーション” の基本の枠組みとして示すように四つの段階に分けてモデリングする方針とした。まず対象

† 茨城大学工学部情報工学科

Department of Computer & Information Sciences, Faculty of Engineering, Ibaraki University

表 1 想定ドメインユーザー (DU) の特徴  
Table 1 Features of Domain Users: DU

<p>1. 応用ドメインの専門家であるユーザ 情報分野以外の科学/工学/技術/生産等のドメイン (専門分野) の分析/設計/計算に関する研究/開発/技術の専門家。流体や構造解析, 画像分析, 化学合成等多様な分野で解析, 設計, シミュレーション等を行う。ほとんどの研究者や開発技術者が含まれる silent majority である。</p> <p>2. いわゆるソフトウェア開発専門家ではない。 業務支援 (図書館, 経理, 銀行等の勘定系) のシステムや計算機特有 (通信, ウイルス対策, データベース) ソフトなどの擬似実世界<sup>3)</sup>ではなく, 前項のような真性実世界<sup>3)</sup>を対象。</p> <p>3. 専門分野に関わる計算/解析結果に関心と価値 プログラム開発については非専門家, 実力/考え/通常的手法/特性, が千差万別。 主たる関心は計算 (処理) 結果に在り, プログラムの書き方や構造等には関心は薄い (無い)。</p> <p>4. 中小規模の複雑なプログラムの個人自主開発 中小規模 (数千~二万行) の, 試行錯誤や改訂の多い自家用プログラムを必要に迫られ個人か小人数で一貫自主開発。そのアルゴリズムは複雑なことが多く, 別言語への書き換え等は嫌がる。</p> <p>5. DU 自身の常用言語以外を避ける。 常用 PL(注)(多くは Fortran) と常用自然言語 (母国語, NJ) は十分に駆使でき, その知識は前提にできる。未知/異型の新規 PL は避ける。</p> <p>6. 計算機に関する時間的/心的負担や労力は最小 CASE ツール, 新しい開発環境や技法等は使いたがらず, 面倒臭がって避ける傾向が強い。</p> <p>7. 自身の専門に関しては多様で十分な実力を持つ 自身の専門の対象世界に関することならば, 必要十分な知識があり, 対象世界の詳細要素を識別でき, 如何なるモデリングでも縦横に行うことが出来, 表現や記述も的確/正確に出来る。</p> <p>8. OO は概念としては一応は理解し使えたと仮定 OO を概念としては理解し, 専門用語を交えた自然言語 (母国語, NJ(注)) でならば, OO に基づく説明や記述も充分にできる, と仮定。</p>
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

(注)PL: Programming Language, NJ: Natural Japanese

世界を“分析モデリング”する段階。この段階で対象世界から必要な要素を OO モデリングする。次はそれを計算機の世界へ持ち込むために変換という一種の OO モデリング, すなわち“設計モデリング”を行う段階, 三つ目はプログラムへの“変換プログラミング

(実装モデリング)”段階である。

### OO 一貫モデリング過程

これが完成すればプログラム駆動 (実行とも言う) でシミュレーションが始まる。これを対象世界と比較対照して相似性の実現を評価する。図 1 の様に対象世界に始まり, プログラム駆動段階まで続く複数の段階が連なる場合を OO 一貫モデリング過程と呼ぶことにする。

### OO 一貫相似性モデリング過程

OO モデリング前後の二つの対応する要素が相似性を持てば, “対応要素間相似性”と呼び, 対応要素間相似性を持つ対応要素が対象世界から始まり, 分析モデリング段階, 分析モデリング段階を経て駆動モデリング段階まで連なって連続的に相似性を持つときには, これら五つの対応要素は一貫相似性を持つ, と呼ぶ。それらの例を図 2 に示す。

図 2 における対象世界の雲と分析モデリングしたオブジェクト雲との対応関係がすべて相似性を持つとき, A1~A4 の四つの対応要素間の関係が一貫相似性を持つと言う。海に関する変換モデリング B1~B4 も同様である。

同じく対象世界では蒸発して雲になるという相互作用と分析モデリングした後の mp (message passing との対応要素間関係 C1, C2, C3, C4 も同じく一貫相似性を持つ。相互作用や集約などの相互関係が一貫相似性を持つと構造の一貫相似性が成立したことになる。

最後に, 以上の様な対応要素間での一貫相似性が対象世界からモデリングしたすべての要素に対して成立した場合, このモデリング過程は OO 一貫相似性モデリング過程と呼ぶことにする。これが成立したときには, 対象世界と再現シミュレーション全体との相似性が成立したことになる。

つまり各 OO モデリング段階間でのすべての対応要素に対して個々に相似性を持たせることが対象世界全体に対する相似性の高い再現シミュレーションの実現に繋がる, というのが核となる概念である。

## 3. OO モデリングの参照モデルの提案

### 各モデリング段階の参照モデル

- (1) 要素種類表 (表 2)
- (2) 要素内部構成表<sup>1)</sup>
- (3) モデリング作業手順 (ダイアグラムと規則)<sup>1)</sup>
- (4) 作業結果としての典型的な記述例<sup>2)</sup>

### OO 一貫モデリング過程全体の参照モデル

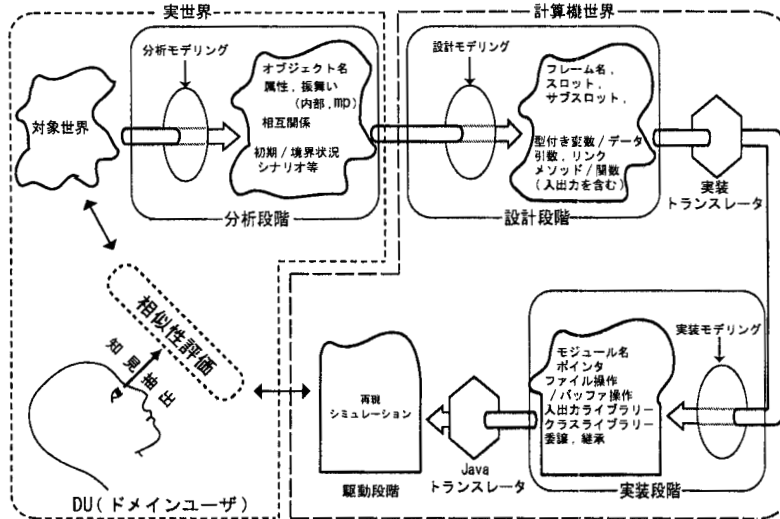


図 1 OO 一貫相似性モデリング過程の方法  
 Fig.1 An Object-based, Integrally Consistent and Similar Modeling processes

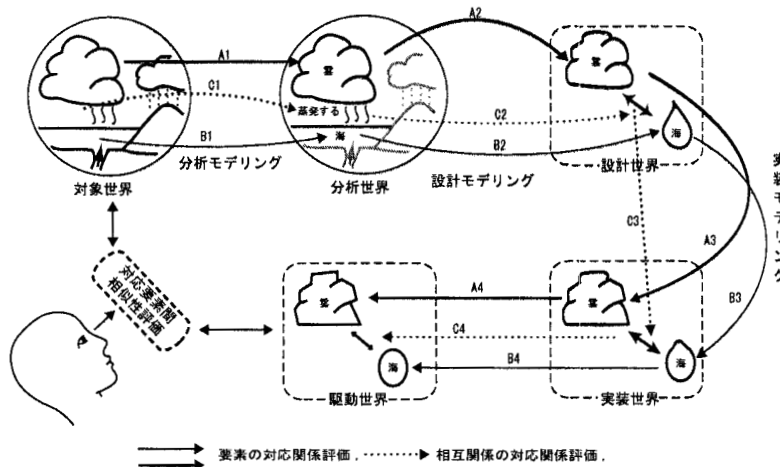


図 2 OO 一貫モデリング過程における一貫相似性の検証の方法  
 Fig.2 A Verification Scheme of Integrated Similarity for an Object-oriented, Integrally Consistent Modeling processes

- (1) 一貫相似性モデリング過程の図 (図 1, 図 2)      いことは自明である。  
 (2) 一貫相似性の高いモデリング例 (記述例)<sup>2)</sup>

#### 4. OO モデリング例と評価/検証

図 3 に分析, 設計, 実装の各モデリング後の対応要素の対応する部分を記述例として引用した。対象世界は水の気象循環現象で, 図 2 にも一部の例としてあげた。三つの段階の記述から見て対応要素の相似性は高

#### 5. まとめ

- (1) 対象世界と高い一貫相似性を持つ再現シミュレーションが DU にも比較的簡単に実現した。
- (2) 対象世界との一貫相似性の明確かつ客観的・形式的な検証方法が得られた。しかも DU 本人ばかりか他人も同じ条件で検証可能である。

- (3) 任意の複数のモデリング段階間の対応要素が明示されるので、比較が十分に出来る。その結果、再現精度が向上する。

問題点：OO に基づいたモデリング・設計・実装に慣れた DU が少ないことである。従って DU に慣れ/経験/熟練が必要である。

総合評価：確かに慣れ等の問題から DU の負担は必ずしも低くないが、得られるプログラムの一貫相似性の高さや信頼性を考慮すれば、これらの負担は十分見合う成果が得られる、と考えられる。

### 参考文献

- 1) 畠山正行, オブジェクト指向分析モデリングの明示形式化・詳細化・手順化, 日本シミュレーション学会誌, Vol.21, No.4, pp.295-309, Dec., (2003).
- 2) <http://gaea.cis.ibaraki.ac.jp/>
- 3) 磯田定宏, 実世界モデリング有害論—オブジェクト指向モデリング技法の解明, 電子情報通信学会論文誌, Vol.J83-D-I, No.9, pp.946-959, (2000).

表 2 対象世界の要素種類  
Table 2 Element kinds of target world

分析モデリング要素	
fn1 オブジェクト名	モノ名, 概念名, 振舞い名
fn2 属性	振舞い参照属性, mp 付置属性 汎化/特化属性 / (被) 集約属性 一般相互関連属性 / 制約属性 可観測/情報隠蔽属性 共有属性, 複合構造化属性
fn3 振舞い	内部振舞い 相互作用伝達 (mp(注)) 相互作用
fn4 相互関係	相互作用伝達 類型化/実値化, 汎化/特化 汎化/特化階層要素 実値要素, 型要素 抽象要素, 高抽象要素 集約/被集約, グループ化 再帰展開 時間的/空間的相互関係 共有 (default は専有/固有)
fn5 数式/定数定義	
fn6 無次元数定義	
fn7 対象世界全体状況記述	初期条件記述, 境界条件記述
fn8 対象世界全定義	スクリプチャ, 振る舞いシナリオ

(注 2) message passing, mp と略

### 分析モデリングの記述 (OONJ)<sup>2)</sup>

fn1.1 海	
1	fn2.8 面積 S、比熱 C、表面温度 T <sub>s</sub> 、表面質量 M <sub>s</sub> 放射熱量 Q <sub>r</sub> 、吸収熱量 Q <sub>a</sub> 、保有熱量 Q
2	fn3.3 熱吸収する
-1	fn3.2   熱放射を受ける <mp< 1:太陽 [2-7]
-2	fn2.5   吸収熱量
-3	fn3.1   熱吸収する
-4	vfn3.1    Q <sub>j+1</sub> = Q <sub>j</sub> + Q <sub>a</sub> * S
-5	fn3.1   温度が上昇する >> [3]
3	fn3.3 温度が上昇する
-1	fn3.1   表面温度を算出する
-2	vfn3.1    T <sub>s,j+1</sub> = T <sub>s,j</sub> + Q / (M <sub>s</sub> * C)
-3	fn3.1   蒸発する
4	fn3.3 蒸発する
-1	fn3.1   蒸発量を算出する
-2	vfn3.1    e = 1.155 * (T <sub>s</sub> - T <sub>z</sub> )
-3	fn3.2   水蒸気が蒸発する
	>> mp >> 13:海上空雲 (高度 1)[2-1]
-4	fn2.5   蒸発量
-5	fn3.1   熱放射する

### 設計モデリングの記述 (OODJ)<sup>2)</sup>

dfn1.1 海	
1	dfn2.8 面積 S、比熱 C、表面温度 T <sub>s</sub> 、表面質量 M <sub>s</sub> 放射熱量 Q <sub>r</sub> 、吸収熱量 Q <sub>a</sub> 実数型 私有
2	dfn3.3 熱吸収する (実数型 Q <sub>a</sub> ) void 共有
-1	dfn3.1   Q = Q + Q <sub>a</sub> * S
-2	dfn3.1   温度が上昇する >> [3]
3	dfn3.3 温度が上昇する void 私有
-1	dfn3.1   T <sub>s</sub> = T <sub>s</sub> + Q / (M <sub>s</sub> * C)
-2	dfn3.1   蒸発を行う >> [4]
4	dfn3.3 蒸発する void 私有
-1	dfn2.5   温度 (高度 0) T <sub>z</sub> 、蒸発量 e 実数型
-2	dfn3.2   温度を取得する
	>> メソッド >> 5:海上空大気 (高度 0)[7]
-3	dfn2.2   T <sub>z</sub>
-4	dfn3.1   T <sub>z</sub> = 結果
-5	dfn3.1   e = 1.155 * (T <sub>s</sub> - T <sub>z</sub> )
-6	dfn3.2   水を受ける
	>> メソッド >> 8:海上空雲 (高度 1)[3]
-7	dfn3.2   熱放射する >> [5]

### 実装モデリングの記述 (OPDJ)<sup>2)</sup>

```

% モジュール 海% 変数 S : 実数型 /* 面積 */
% 変数 C : 実数型 /* 比熱 */
% 変数 Ts : 実数型 /* 表面温度 */
% 変数 Qr : 実数型 /* 放射熱量 */
% 変数 Qa : 実数型 /* 吸収熱量 */
% 変数 Q : 実数型 /* 保有熱量 */
% 変数 Ms : 実数型 /* 表面質量 */
% 変数 ep : 実数型 /* 放熱率 */
% 変数 sigma : 実数型 /* シュテファン・ボルツマン定数 */

% 処理 熱吸収する : void (Qa:実数型) /* Aborb __ Heat */
Q = Q + (Qa * S);
call 海 . 温度が上昇する ( );

% 処理 温度が上昇する : void ( ) /* Risen __ Temp */
Ts = Ts + Q / (Ms * C);
call 海 . 蒸発する ( );

% 処理 蒸発する : void ( ) /* Evaporate */
% 変数 Tz : 実数型 /* 温度_高度 0 */
% 変数 e : 実数型 /* 蒸発量 */
Tz = 海上空大気_高度 0, 温度を取得する ( );
e = 1.155 * (Ts - Tz);
call 海上空雲_高度 1, 水を受ける ( e );
call 海, 熱放射する ( );

```

図 3 分析, 設計, 実装段階の記述例の相似性比較

Fig.3 Comparisons of Modeling Descriptions among Analysis, Design and, Implementation stages