情報処理学会論文誌　数理モデル化と応用　Vol.5　No.3　63–78 (Sep. 2012)

UMLとの比較に基づくオブジェクト指向分析設計記述言語OONJの評価

池田 陽祐[1]　三塚 恵嗣[2]　加藤木 和夫[3]
大木 幹生[4]　上田 賀一[5]　畠山 正行[5,a)]

受付日 2012年2月14日，再受付日 2012年4月8日，
採録日 2012年5月17日

概要：記述言語 OONJ は大学院生向けの科学技術計算やシミュレーション分野での教育目的の利用のために開発されてきた．一方で OONJ 自体が記述言語として適切に設計されているかについてソフトウェア工学的な観点からの評価が必要になった．そこで本論文では代表的なモデリング言語である UML との比較と評価を行った．比較には UML のクラス図，アクティビティ図を用いた．また，構成要素の概念的比較を行った．その結果 OONJ が静的・動的の両側面について十分な対応関係と記述内容を持つことが明らかになった．また OONJ は想定ユーザと記述想定分野を絞ったゆえに，汎用言語である UML に対してはサブセットであることが分かった．結論として，OONJ の想定ユーザと想定分野に対しては，UML のダイアグラムと同等の記述力と記述特性を持つことが分かった．

キーワード：UML，オブジェクト指向，記述言語，分析，設計

The Estimation of the Object-oriented, Analysis and Design Descriptive Language OONJ Based on the Comparisons with UML

Yousuke Ikeda[1]　Keishi Mitsuka[2]　Kazuo Katougi[3]
Mikio Ohki[4]　Yoshikazu Ueda[5]　Masayuki Hatakeyama[5,a)]

Received: February 14, 2012, Revised: April 8, 2012,
Accepted: May 17, 2012

Abstract: The descriptive language OONJ has been developed for the education purposes of the graduate students in the domain of science or engineering and in the domain of simulations. The estimation has been needed from the stand point of the software engineering whether OONJ has properly been designed as the descriptive language. To attain the aim, we have compared with the representative modeling language UML. The class diagram, and the activity diagram has been used for comparisons. As the results, it has been made sure that OONJ has the sufficient correspondent relations and the descriptive contents from both the static and dynamic aspects. Since the users and the fields to apply OONJ have been restricted, it has become clear that OONJ is the subset of UML. It has been concluded that OONJ has realized the equivalent descriptive power to UML for the target users and the target fields.

Keywords: UML, object oriented, descriptive language, analysis, design

1. はじめに

実世界のある一部を対象世界の問題領域として分析・モ

[1] 茨城大学大学院理工学研究科博士後期課程情報・システム科学専攻
Graduate School of Information and System Sciences, Ibaraki University, Hitachi, Ibaraki 316–8511, Japan
[2] 株式会社日立情報システムズ
Hitachi Information Systems, Ltd., Shinagawa, Tokyo 141–8672, Japan
[3] 茨城県立産業技術短期大学
Ibaraki Prefectural Industrial Technology Junior College, Mito, Ibaraki 311–1131, Japan
[4] 群馬工業高等専門学校教育研究支援センター
Technical Support Center for Education and Research, Gunma National College of Technology, Maebashi, Gunma 371–8530, Japan
[5] 茨城大学工学部情報工学科
Department of Computer and Information Sciences, Ibaraki University, Hitachi, Ibaraki 316–8511, Japan
[a)] htkyma@mx.ibaraki.ac.jp

デリングし，最終的には科学技術計算やシミュレーションを実現するに至る技術は，環境の負荷軽減の問題解決や大規模制御システムの安全性の検証等に応用されることで，現代においてはますますその重要性を増してきている．そのような技術は大規模な科学技術計算や工学シミュレーションを扱う技術 [1] に発展している．しかしその一方では，将来的にそのような技術の開発に携わる高度な技術と知識を持つ人材が不足しており，大学院教育が期待されている [2], [3].

このような流れに沿って私たちは主に大学院生を対象とした教育を目的とし，彼ら自身の教育や研究，そして将来を見据えた開発や設計等を行うプロを育成するための仕組みを開発してきた．その 1 つが対象世界の問題領域を分析・モデリングするための分析設計記述言語の開発である．具体的にはオブジェクト指向（Object Oriented（以下 **OO** と略））に基づいてモデリング [4], [5] と分析設計を行い，そのモデルの記述用に構造化の仕組みである OOSF（Object Oriented Structured Frame）[6] を開発し，それを核とした OO 分析設計記述言語 OONJ（Object Oriented Natural Japanese）[7], [8] を開発してきた．

開発の狙いは OONJ の想定ユーザ自身が把握している対象世界の概念体系やモデリングの方法に十分に適合させた分析設計記述を実現することである．

しかし一方，OONJ をソフトウェア工学[*1]の範疇の記述言語と位置付け，記述言語としての記述力や記述特性等を客観的に確認し，評価する必要がある．それは OONJ が記述言語として学術的な見地からの客観的な評価が十分ではないからである．その評価は相対的な比較評価が基本とし，その基準言語としては UML [9] が最適である．よく知られているように UML は計算機で用いられる世界標準の OMG で規格化された言語 [10] でもある．これとの比較評価によって OONJ がどれだけの記述力[*2]や記述特性を持っているかを評価する．

また OONJ が微分方程式を支配方程式とする物理現象等の数理モデルや概念モデルを記述するための記述言語であるのと対照的に，企業の業務処理ソフトウェア開発に使われることが多い UML であることを念頭に比較する．本研究では，記述対象の静的と動的の両側面において，UML の多くのダイアグラムの中からクラス図とアクティビティ図を取り上げ，その記述力や記述特性を比較し評価すべきであると考える．

以降，2 章では対象世界と離散モデリングや想定ユーザについて述べる．3 章では対象世界の記述技法を，4 章では構造の記述技法を説明する．中心となる 5 章では主に UML との比較を詳細に行い OONJ を評価する．6 章で結論と今後の課題を述べる．

2. 対象世界とその離散モデリングおよび想定ユーザ

2.1 一次元衝撃波流れの定式化とその離散モデル

まず対象世界の具体例として「一次元衝撃波流れ」[11] を取り上げる．理由は超音速流体力学の専門分野では広く知られている古典的な問題であることと，シミュレーション計算式が簡単であるということにある．その現象の理解を得るために実世界の衝撃波管（Shock Tube）を設定する．衝撃波管のイメージを**図 1** (A) に示す．ただし簡単のため定立衝撃波が伝播する付近だけをモデリングの対象とする．この流れは数学的には二階の非線形偏微分方程式の初期値境界値問題として定式化される．数理計算的な立場からの離散モデリングを行った結果の概略 [11] を**表 1** に示す．

表 1 の離散単位セルは，想定ユーザも日常的に使う図 1 (B) の “セルモデル” と同じである．このセルモデルは差分方程式で表現された数理計算向けのモデルである．セルモデルは従来の計算機能に着目した方法（たとえば Solver [12] を用いる方法）であれば適切なデータ配列にとられるが，本モデルでは表 1 の (4) のようにセルという名前の離散単位[*3]として定義され，表 1 の (3) のように離散単位セルの属性として衝撃波伝播速度を表す u が内部に定義される．u は表 1 の (2) にあるように Euler 方程式の差分方程式を導入してその時間と空間の変化を「離散単位の

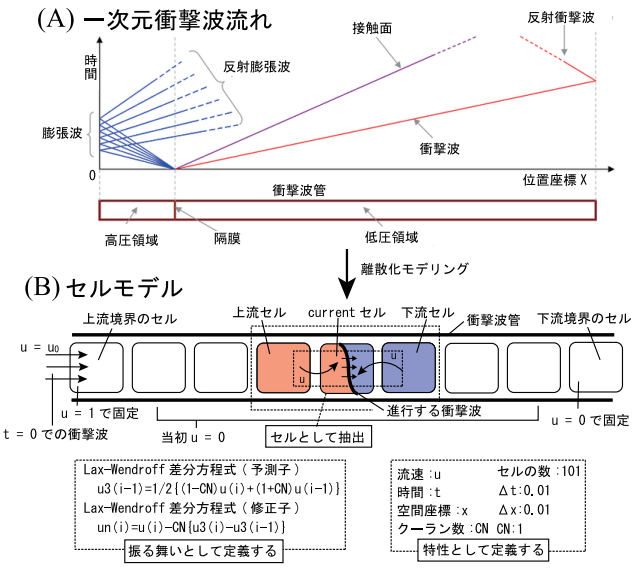


図 1　対象世界の概略とセルモデルの構成

Fig. 1　Outline of the target world and the cell-model.

[*1] この用語は大規模なソフトウェア開発を行う体系の学術的な側面を指す用語として用いる．

[*2] 記述力とは，書きやすいとか的確に表現できるとかにはかかわらず，とにかく「書けるか否か」である．日本語で書けない分野はほぼないと答えられるであろう．その場合，記述力は強いと形容する．

[*3] オブジェクト指向ではインスタンスあるいはクラスにアサインされるべき単位で，離散的であることを強く意識したネーミングである．後に 5 章で UML との対比として議論する．

表 1　一次元衝撃波流れの概念モデルとしてのセルモデル
Table 1　Cell model as the concept model for one-dimensional shock wave flow.

(A) モデリングの準備としての物理量の無次元化
(1) 対象世界の属性（物理量を表す変数相当）はすべて一定の基準物理量値で無次元化する．時刻 t は t0=L/u0 で，速度 u は流入する流れの音速 u0 で，空間座標 x は平均の分子間自由飛行距離 L で無次元化する．

(B) 一次元空間を離散化したセルモデルと，離散単位セルの定義
(2) 衝撃波管内の空間はセルという名称の番号付けされた微小空間に分割し，各々を離散単位セルと定義する．
(3) 各セルは属性として衝撃波速度 u を持つ．ただし Solver 型等と異なり u のデータ配列はとらない．
(4) 衝撃波速度 u は，Euler 方程式を差分方程式化した Lax-Wendroff の予測子・修正子法を用いて計算する．衝撃波速度 u は計算ではなくセルの振舞いと位置付けられる（差分方程式の導出等の詳細は省略する）．
(5) 対象世界で共通な属性として，空間セル幅 Δx=0.01，時間幅 Δt=0.01，クーラン数 Cn=Δt/Δx=1 を持つ．
(6) 衝撃波の中間に位置するセルは t=0.0 において u=0.0 を与える．
(7) 境界に設定されたセルは，流入口の x=0.0 においてはすべての時刻 t において u=u0 の一定速度を持たせ，流出口ではすべての時刻において u=0.0 であり，左側から衝撃波が伝播してきた時点で計算を終了する．

(C) セル間の相互作用のモデリング
(8) セル間の相互作用は離散単位間の相互作用の情報を伝達するための情報モデルであるメッセージ・パッシング（message passing 以降 mp）の形式にモデリングされ，伝達される．
(9) 隣接セル間で行われる mp の伝達内容は自身の衝撃波速度 u であり，影響を及ぼすセルへ伝達する．

振舞い」として計算する．初期値と境界値は表 1 の (6) と (7) のとおりである．セル間の相互作用は (9) のように相互作用の情報伝達方式である mp（表 1 の (8)，(9) 項，および後の 4.1 節参照）により行う．

ただし表 1 では日本語で書かれた概念モデルの概略はできているが，骨格だけであり，さらに十分な肉付けをした分析「記述」に組み上げる方法や言語が不可欠である．通常のソフトウェア開発では要求分析のステップで表 1 のモデルを得るが，OONJ では想定ユーザ自身がその分野の専門家である（2.3 節参照）ため既知のモデルである．

2.2　離散モデリングの基礎概念

離散単位のモデリング結果の構造記述の方法として**図 2** のように離散単位とその構造化 [6], [13], [14] だけでとらえるという簡潔な原理を用いる．本論文の構造化は図 2 にあるように離散単位間に対して関わり，すなわち相互関係を付与する，との簡潔な定義を採用している（詳細は 4.1 節で述べる）．この原理をすべての離散単位に適用して記述技法を構築する．

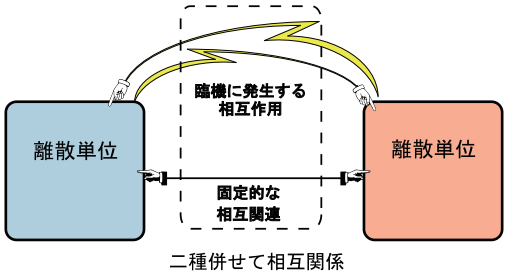


図 2　離散モデリングと構造化の基礎概念
Fig. 2　Fundamental concept of discrete modeling and its structurations.

2.3　記述言語 OONJ の想定：対象分野，想定ユーザ，設計の狙い等

OONJ と UML における対象分野や想定ユーザ等の典型的な状況や取り巻く環境を対比的に**表 2** に示し，表の想定項目の番号を引用しながらその詳細を述べる．

【想定対象分野】：表 2 の想定項目 (1), (2) より，理工学分野の研究や技術開発のプロ養成の教育を目的とする．複雑な関わりを持つ計算式やデータ構造を扱うのが特徴である．

【想定ユーザ】：彼らは想定項目 (1)，(3)，(5) のように，大学院で研究開発のプロになるための基礎訓練を受けており，一定以上のレベル [3] のプログラム作成技術を必要としている．そのような目的のために，最終的には 5,000 行程度のプログラム（OONJ 記述に換算すれば約 2,000 行程度）を作成できる程度の技術の習得を OONJ で達成することを目標としている[*4]．

(1) 彼らは (3)，(4)，(5)，(12) のように自身の研究開発の必要上，自身でモデリングからプログラムまでを一貫して行い計算も行う．その計算結果から対象世界の知見や設計データ等を得ることを主な目的としている．
(2) 彼らは自身の専門分野については項目 (12) により一応の水準の知識があり，対象世界の分析も自力で行える．自身の専門分野のモデル等はすでに持っている．
(3) 同じく想定項目 (5)〜(8) より，多くは手続き型のプログラミング言語 1 つを必要な範囲で知っており，「他との協同を前提にすることなく」，「自身だけで」，「ゼロから一貫して自主的に」，「他人用ではなく自分用」に「小規模」のプログラムを作成する．
(4) 同じく想定項目 (9)〜(12) より，ソフトウェア開発の専門家ではないのでソフトウェア開発の多くの知識やノウハウ，高い OO プログラミング技術は期待できない．
(5) プログラム作成の負担は最小限に，問題点解決にかけ

[*4] 想定ユーザの裾野や周辺には，研究室配属の 4 年生，研究員や助教クラスの人の一部も入る．

表 2　OONJ と UML（ソフトウェア開発）を取り巻く典型的な想定状況や環境の比較
Table 2　Comparisons of typical assumptions or circumstances that surround OONJ and UML.

想定項目	OONJ の想定	UML（ソフトウェア開発）の想定
(1) 主な想定ユーザ	**大学院生（プロ訓練中）**	**ソフトウェア開発技術者（プロ）**
(2) 対象世界	物理現象等の自然世界，工学上の問題領域	企業・事業の業務（経理，人事）
(3) 記述の「直接」目的	対象世界の分析（離散構造化モデリング）結果の記述．その後プログラミング段階へ移行	業務の対象世界自体を理解する目的で描画・記述する．
(4) 記述の「最終」目的	科学技術計算，シミュレーション，設計	業務支援処理，大規模制御システム
(5) ニーズ発生源	研究・開発の必要上	顧客の発注
(6) マンパワー資源	個人単独	複数（多数）の開発技術者
(7) 組織	すべて個人で	組織的協力
(8) 開発者とユーザ	ユーザ自身が開発者（記述者という）	開発者と顧客（ユーザ）は別人物
(9) 納期，価格，契約	なし	商品なのですべて実行される．
(10) 図やドキュメント	レポートや学会発表用資料	大量の文書
(11) 開発環境	テキストエディタ，グラフィックスツール	多様な統合開発環境
(12) 専門分野	工学，理学，モノ製造技術	ソフトウェア開発技術
(13) 記述規模	2,000 行程度（プログラム換算 5,000 行程度）	数万行以上数百万行に及ぶことも
(14) 学習・習得のコスト	大学院での講義と演習での 1 コマ程度	本格的な講習と実務経験が必須

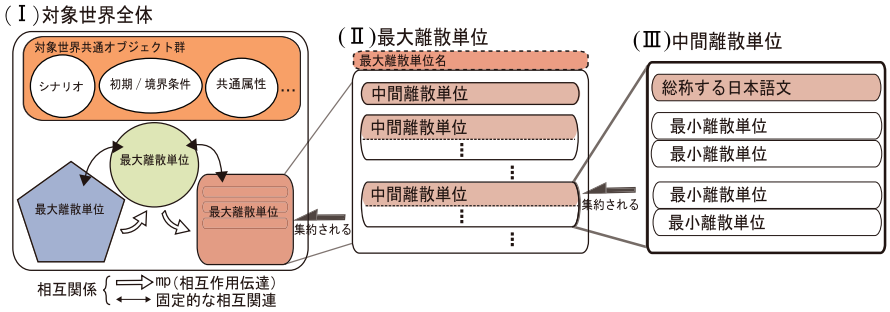


図 3　OOSF の仕組みと構造
Fig. 3　Mechanism and structures of the OOSF.

る時間は最大限にする．

【設計の狙い】：表 2 の目指す機能や記述力，記述特性を持つ記述言語を実現する．

3.　対象世界の記述技法[*5]

3.1　OONJ の基本構造を規定する OOSF

各離散単位とその内部構成を規定する OOSF [6] の仕組みを**図 3** に示す[*6]．図 3 から分かるように，離散単位としてはオブジェクトやクラスに相当する図 3 の (Ⅱ) 最大離散単位が，そして図 3 の (Ⅲ) 中間離散単位，最小離散単位の 3 種類が定義される．図 3 の (Ⅰ) は対象世界全体を表現している．対象世界内部には複数の最大離散単位（図中には 3 つ）が存在し，相互に固定的な構造（相互関連）を結び，あるいは互いに相互作用の情報を伝達しながら自身の特性を更新して動くことで対象世界全体の動きを形成してゆくことを示している．

その各最大離散単位の内部構成の図 3 の (Ⅱ) には，複数の中間離散単位が集約（has-a 関係）されている．本論文ではつねに集約を “has-a” の意味で用いる．さらに中間離散単位の図 3 の (Ⅲ) では，同様に複数の最小離散単位が集約されている．最小離散単位には日本語文を 1 文だけ格納する．日本語文は OOSF における最小の離散単位の実体を表す．つまり，離散モデリングによる OOSF の離散単位には 3 階層の集約階層構造が作り込まれている．

この 3 階層の構成は想定ユーザの利用の便利さのために設定された．拡張機能としては，複数の最大離散単位を集約（has-a 関係）した複雑な離散単位や，最小離散単位の内容を改めて詳細にモデリングして（“展開” という相互関連を結んで）最大離散単位に新規に定義することもできる．また階層構造という観点では，プログラミング言語におけるクラス階層に相当する汎化階層も “汎化” という相互関連を結ぶことで，継承関係（is-a 関係）を持たせて定義できる [16]．本論文は基本構成を対象とするので拡張機能は割愛する．

*5　2.3 節 (2) 項より想定ユーザは対象世界のモデルを常時持っている．彼らが OONJ に求めるのは「記述のためのテクニック（技法）」である．
*6　厳密さはないが読者により分かりやすく，やさしい資料として，工学部 3 年生向けの講義と演習で用いた OONJ の図解資料が当研究室の web page [15] に掲載されている．

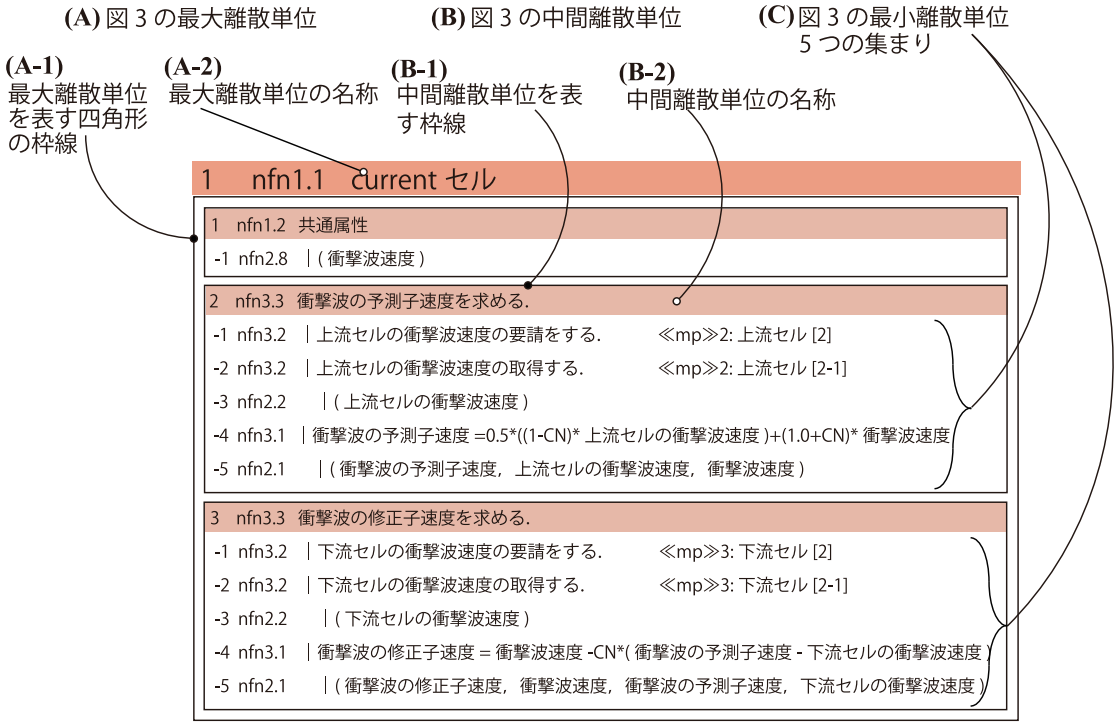


図 4　最大離散単位のセル（current セル）の記述

Fig. 4　Description example of the maximum scale discrete unit “Current Cell”.

3.2　各階層ごとの離散単位の記述技法

OONJ では対象世界を離散化し，その離散単位ごとに記述する．図 1 から抽出される最大離散単位はセルであり，その一次元衝撃波流れの current セルの記述例を**図 4** に示す．図 4 **(A)** は図 1 のセルを具体的に記述したものである．それは，図 3 の (Ⅱ) の最大離散単位であり，**(A-1)** のように四角形の枠線を用いて記述する．最大離散単位の名称は **(A-2)** で示すように枠外の上部に記述する．

次に最大離散単位であるセルの内部構成には表 1 のセルモデルから振舞いや特性を抽出し，図 3 の (Ⅲ) の中間離散単位として記述する．図 4 **(B)** に差分方程式の計算を 1 つの振舞いととらえた記述例を示す．中間離散単位は **(B-1)** のように細長い四角形の枠線で囲み，枠線内の最上部に中間離散単位全体を総称する日本語文（総称文）である **(B-2)** を 1 つと，総称文に集約される図 4 **(C)** の日本語文が複数集約される．この個々の日本語文は最小離散単位として扱われる．用いる日本語文が最小離散単位であるためには，1 つのモノや概念，事物を一意に指すような単文，特に数式・計算式・論理式等であることが推奨される．日本語文や計算式には日本語の文法や数学以外の新たな制約はまったくなく，想定ユーザ自身が決めた適切な形式で書けばよい．以上のように OONJ では，OOSF に 3 つの集約階層を持たせ，相互に内包関係にあることを枠線で表現する．

3.3　離散単位を一意に識別する記述技法

3.1 節で述べた 3 階層の離散単位の数は多数にのぼり，それらすべてを離散単位の記述上で一意に識別する方法が必要である．そこで OONJ では最小離散単位までを識別できるように最大離散単位ごとに，中間離散単位ごとに，そして最小離散単位ごとに付した「一連番号の組」と「種類を示す記号（facet number，fn と略記する [6]．OONJ では特に “nfn” を用いる)」をセットにした識別子を用いて区別する方式をとっている．具体的には最大離散単位番号，中間離散単位番号，最小離散単位番号，facet 番号の組を対応させればよい．これは日本図書分類法（NDC）と類似の方法である．

実際の記述例として**図 5** を示す．図 5 **(A)** が最大離散単位の識別子である．この記述では番号が「1」，種類を示す記号が「モノ・概念を示す fn1.1」，名前が「current セル」であることを示している．中間離散単位の例を図 5 **(B)** に，最小離散単位の例を図 5 **(C)** に示す．たとえば図 5 **(C)** の日本語文を一意に特定するには「1:current セル [2-4]」と記述すればよい．図 5 **(D)** にも別の例が見出せる．

3.4　対象世界共通オブジェクト群等の特別な離散単位

対象世界には実在しないが，必要となる特別な離散単位が数種類ある．それらを OONJ では対象世界共通オブジェクトと呼ぶ．たとえば微分方程式を解く際の初期条件や境界条件であり，世界をイメージとして頭の中で動かす助けをするシナリオであり，図 3 の (I) に位置づけられている．よく使われる最大離散単位の例を**表 3** に，記述例を**図 6** に示す．

記述例の図 6 **(B)** ではすべての初期条件が 1 つの最大離

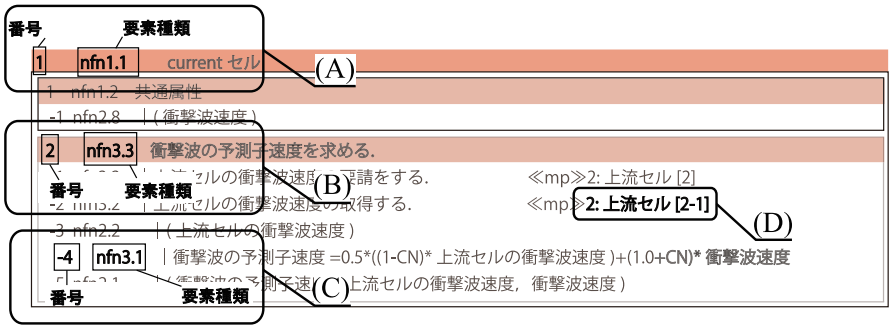


図 5　離散単位を識別するための記述技法の例

Fig. 5　A description technique to identify uniquely the every discrete unit.

表 3　対象世界共通オブジェクト群

Table 3　World common objects in the target world.

代表的な対象世界共通オブジェクトの記述内容
＜初期条件＞： 時刻 t=0.0 の各属性の実値を与える.
＜境界条件＞： 空間的な境界における実値を与える.
＜対象世界共通変数＞： 対象世界に共通な変数，定数等.
＜シナリオ＞： 記述された世界を起動し，各離散単位の起動手順を示す.
＜時間と空間定義オブジェクト＞： 時刻や時間スケール，空間座標系
＜スクリプチャオブジェクト＞： 対象世界全体を定義し，世界を起動する main 相当.

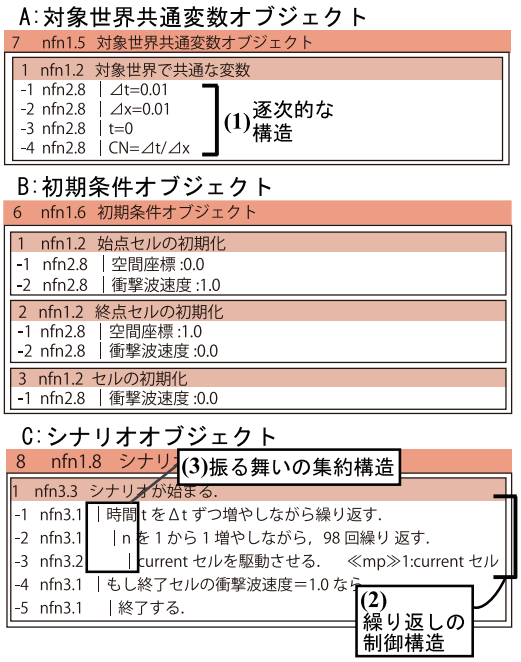


図 6　初期条件，対象世界共通変数オブジェクト，シナリオオブジェクトの記述例

Fig. 6　Description examples of target world global attribute object, initializing object, scenario object.

散単位内部に集中的に定義されているが，これらは分割・分散して配置されてもよい．これらは記述法として一意に定義されていない．それは分野ごとに異なる可能性が高くて定義できないことと，表 2 の (1)〜(4) から想定ユーザならば自身の分野の形式で書くことが十分期待できるからである．

4.　構造の記述技法

4.1　相互関係を用いた構造の記述技法

OOSF の構造は離散単位間に相互関係を付与する形式を採用しており，OONJ も同様である．この「相互関係」は，静的な関係である「相互関連」と動的な関係である「相互作用」の 2 種類に分けて設計した [6].

【a】相互関連とは主として集約や汎化等に代表される固定的な関わりである．

【b】相互作用とは一般には相手に臨機的に影響を与えることであるが，OONJ では表 1 の (C) 項 (8) にあるように，相互作用情報伝達（message passing（略して **mp**））を行う情報伝達モデルを用いる．

相互作用と相互関連の記述構成はともに同じで，相互関係の「向き」と「種類」,「相手先の一意な特定」の 3 項目を用いて記述する．相互作用（mp）の例を**図 7** に示す．相互関連も同様なので本文では省略する．図 7 **(B)** の記述は，左から相互関係の向きを示す 2 つのギュメ「≪」,「≫」を使い，「≫　≫」は発信を，「≪　≪」は同じく受信を，「≪ mp ≫」はメッセージの送受信を意味する．2 つのギュメに挟まれた「mp」は相互関係の種類で，名詞を用いて「mp」や「集約」等のように記述する．最後の部分は「相手先」を示している．相手先の特定には 3.2 節で示した「2: 上流セル [2-1]」のように記述する．

以上のように OONJ では振舞いが元々持っている意味内容を検討し，相互関係として改めて抽出して付与することで，対象世界が持つ構造を明示することができる．

4.2　日本語文間および日本語文内の構造の記述技法

最小離散単位である日本語文間の構造化には，日本語文

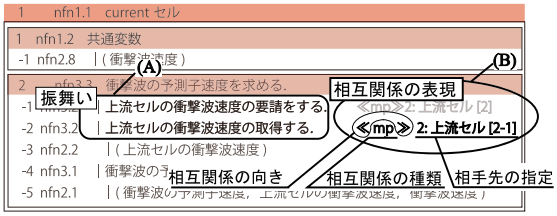


図 7　相互関係の記述技法

Fig. 7　Expression technique to describe the mutual relations.

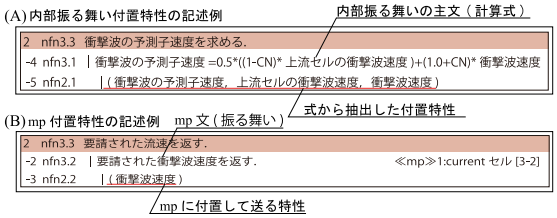


図 8　OONJ における付置特性の記述例

Fig. 8　Description examples of attached attribute for minimum discrete unit.

の前に全角 1 個分の字下げと縦棒線（“|”）の記号を適切に付与して記述する．たとえば，“集約” の場合には図 6-C の **(3)** のように記述する．同位の縦棒線で構成される文章の場合には，図 6-A の **(1)** で示している逐次的な構造であったり，図 6-C の **(2)** で示している繰返し構造等の日本語文の構造であったりしても記述可能である．以上のように日本語の箇条書き，制御構造の一部を流用することで日本語文間を構造化している．

次に日本語文内部には属性やローカル変数とすべき特性が含まれている場合があり，これらを独立させて扱いたい．そこで日本語文内部の特性を抽出し，その日本語文の直下に “付置する形で” 記述する．具体例を**図 8** に示す．この日本語文に付置する特性（以降，付置特性と呼ぶ．この用語は我々の造語である）は 2 種類に分けて述べる．

【1】内部振舞いの付置特性の記述

内部振舞いとは離散単位の外部との相互作用情報伝達（mp）がない振舞いである．付置特性が図 8 **(A)** のように複数であればコンマで区切って並べ，全体を括弧で括る．

【2】相互作用情報伝達（mp）の付置特性の記述

送る側の mp 文（mp を行う振舞い文の意）から抽出されて相手に送られる情報やデータ，モノ等を記述するための付置特性である．記述例を図 8 **(B)** に示す．

このように日本語文から抽出した特性は “付置” という相互関連を用いて構造化する．

4.3　セルモデルにおける差分方程式の記述技法

前節で述べたように，最小離散単位は日本語文であるが，想定ユーザの実体からいえば OONJ 記述完成時には図 4〜

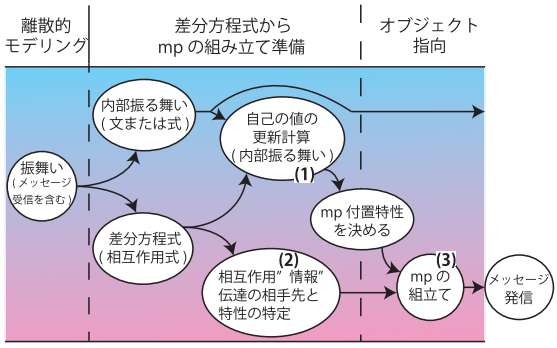


図 9　差分方程式のセルモデルから変換作業の手順

Fig. 9　Transformation procedures from the difference equations in the cell model.

図 8 にあるように数式・計算式・論理式が多い．この傾向は記述規模が大きくなればいっそう顕著になる．そこで本節では差分方程式計算を行う OONJ 記述を作成する記述技法を紹介する [17].

図 1 のセルモデルにおいて示した Lax-Wendroff の予測子・修正子法のような陽的な形式の差分方程式は一般に，複数セルの特性値（たとえば衝撃波速度 u）の一次結合で表され，その右辺は他セルの特性値から計算・更新される．他セルの特性値は相互作用に相当し，これを使って自身の新しい時刻の値を計算する．

次に計算・更新後の自身の値 u を関連セルに送る．これが他セルに対する相互作用つまり mp である．これを記述するには，差分式を分析して，速度 u を mp で送ってきた関連セルを抽出してそのセルに対して mp を送る．そのとき mp を (a) どこに送るか，(b) 何を付置特性で送るかを抽出し mp を組み立てる．結果はすでに図 4 に示した．

この過程を一般的な作業手順にまとめて**図 9** に示す [17]．図 4 から分かるように，図 9 の (1) で内部振舞いの計算式を使って計算するとともに，差分方程式から図 9 の (2) で相互作用情報伝達（mp）に必要な特性を抽出し，(3) でメッセージを組み立てる．

使った差分方程式の例ではごく簡単なケースなので目視で分かるが，複雑な差分計算ではこの抽象化された作業手順である図 9 をさらに詳細化してセルの振舞い（差分方程式計算）の記述や定義すべき特性を形式的に扱うことで正確に抽出することができる．差分方程式計算等を行う想定ユーザに対してはこのような記述技法の手順の提供が必要である．

5.　UML との比較とその評価・考察

5.1　UML と OONJ を取り巻く状況や環境の比較

UML [10] は表 2 にあるようにソフトウェア開発の視点で高い専門性を持つモデリング言語であり，主として業務支援（企業の事務処理システム等）や大規模制御システム等を複数人の開発技術のエキスパートが協力して行う組

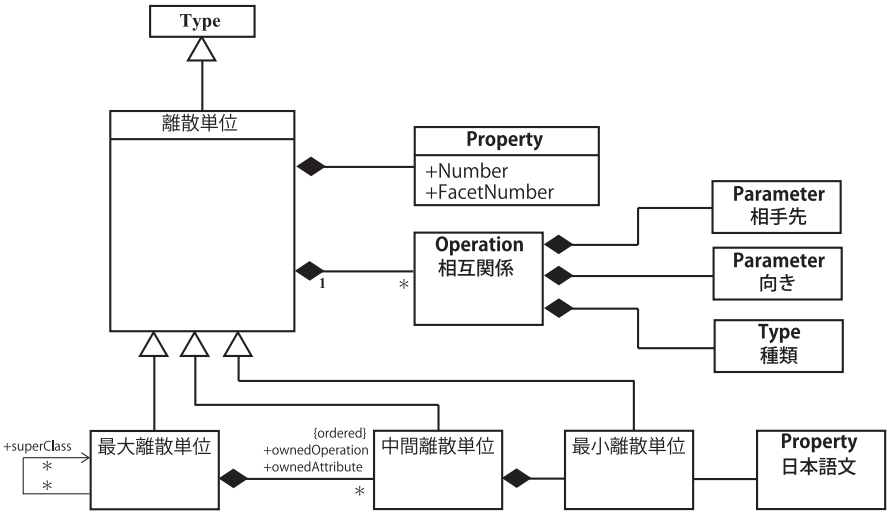


図 10　UML メタモデルに基づいた離散単位の記述構造定義

Fig. 10　The definition of the discrete unit description structures based on UML meta model.

織的な体制下でのソフトウェア開発の際に用いられる [9]. UML では表 2 の (8) のように複数の顧客（ユーザ）を別途に想定する．UML を利用する開発チームはクラス図やユースケース図等の多数のダイアグラムを使って開発し，システム全体を多様な側面から網羅的に表現する．そしてすべての情報を document の形式で作成して共有する．

一方 OONJ では，多種類ある UML のダイアグラムを使う必要性が薄い．なぜならば OONJ では想定ユーザ（開発者とは呼ばない）自身は，すでに対象世界の知識やモデルを十分持っており，顧客（ユーザ）と同一人物であるゆえに必要なモデリングと記述はすべて自力で書ける・書くからである．したがって開発チームの他のメンバや顧客（ユーザ）のためのダイアグラムを書く必要がない．

以上から，UML と OONJ はその状況や環境が大きく異なる場合が多いことが分かる．しかし本論文ではそれらも考慮しつつ，両言語の記述対象の静的な側面や動的な側面（処理や振舞い）[*7]に焦点を当てて比較と評価を行う．

5.2　UML メタモデルと OONJ の離散単位との比較

本節では UML メタモデル [18] と UML の Specification [19] を用いて UML の Structure と OONJ の離散単位である「静的な側面」の構成要素を比較する．比較の基準はクラスの基本定義である文献 [17] の p.95 の figure10.3 [*8]（以下，これを Classes diagram と呼ぶ）である．UML メタモデルを使用した OONJ の離散単位の定義を**図 10** に示す．

図 10 では 3.1 節で示した OONJ の最大離散単位が複数の中間離散単位を集約し，そして最小離散単位を集約していることを示している．また，3.3 節で示した離散単位の種類と番号による識別子や，4 章で示した相互関係が離散単位ごとに内包されていることも示している．

Class には OONJ の最大離散単位が相当し，Class の Property と Operation に相当する内部構成は中間離散単位である．しかし OONJ ではさらに中間離散単位の詳細な実体を日本語文を用いて記述しており，その点が異なる．また 3.3 節で示した離散単位の種類を示す FacetNumber，離散単位の一連番号を示す Number を離散単位の Property として定義している．これは 3.2 節で示した対象世界の離散単位を一意に識別するための要素であり，UML では用いられていない．4 章で示した相互関係は離散単位に Operation として集約されており，Parameter として相手先，向き，そして Type として相互関係の種類を集約関係を用いて定義している．UML では各ダイアグラムごとに記述対象となる構造が異なるため Classes diagram では定義されていない．

その一方で Classes diagram で定義されている Property の Type，TypedElement は OONJ の最大離散単位には存在しない．それは OONJ では対象世界を記述することが主目的であり，Type，TypedElement はプログラムへ変換する作業過程で定義・設定されるからである．OONJ 記述では分析に終始し，プログラム開発へ進まない場合もある．

以上の対応関係を**表 4** に示す．最大離散単位は主にクラスに相当しており，オブジェクトやコンポーネント等にも相当する可能性がある．中間離散単位は主に特性（Feature）に相当する概念であるが，内容的には Operation や Property に相当することが分かった．最小離散単位との対応関係は存在しなかったが，これはクラス図ではメソッド内部の情報を記述しないからである．

[*7] モデリングにはオブジェクトモデル，機能モデル，状態モデルの 3 つの側面のモデリングがあり，これらの観点から比較を行っていく．ただし状態モデルは，各オブジェクトの状態を結果として得るのが科学技術計算の目的であるため作成する必要はない．

[*8] UML Infrastructure Specification [18] の Core::Basic で定義されている Classes Diagram である．

表 5　UML の Behavior と OONJ の振舞いの対応関係
Table 5　Correspondent behaviors between UML and OONJ.

Behavior の基本概念	対応する OONJ の構成要素
Action	
OpaqueAction	中間離散単位（振舞い）
InvocationAction	最小離散単位（相互作用伝達振舞い）
SendSignalAction	- *1
CallAction	最小離散単位（相互作用伝達振舞い）
CallBehaviorAction	最小離散単位（相互作用伝達振舞い）と 中間離散単位（特殊な振舞い）
CallOperationAction	最小離散単位（相互作用伝達振舞い）
MultiplicityElement	- *2
Pin	相互関係記述
OutputPin	右向きのギュメ（≫　≫）
InputPin	左向きのギュメ（≪　≪）
ValuePin	左向きのギュメ（≪　≪）と最小離散単位（付置属性）
Activity	
ActivityEdge	相互関係（線表記なし）
ControlFlow	相互関係（線表記なし）
ObjectFlow	相互関係（線表記なし）
ActivityFinalNode	-
ActivityGroup	-
Action	中間離散単位
ActivityNode	
ControlNode	-
InitialNode	- *2
JoinNode	- *2
MergeNode	- *2
DecisionNode	- *2
ObjectNode	最大離散単位
ActivityParameterNode	- *1
Common Behavior	
Behavior	中間離散単位，最小離散単位（振舞い）
OpaqueBehavior	最小離散単位（振舞い）
FunctionBehavior	最小離散単位（振舞い）
BehavioralFeature	離散単位の指定子
BehavioredClassifier	ファセット記号
OpaqueExpression	最小離散単位

- 該当要素なし
*1：特殊な最小離散単位を使用することで同様の記述は可能である．
*2：日本語文で記述される場合がある．

表 4　OONJ の離散単位と UML の特性要素における対応関係
Table 4　Correspondence between the descritized unit of OONJ and the Elements of UML.

OONJ の離散単位	UML の特性要素
最大離散単位	Class （オブジェクトやコンポーネントの可能性もある．）
中間離散単位	特性（feature），属性（property）， 操作（operation）
最小離散単位	なし

5.3　UML の Behavior と OONJ の振舞いの比較

動的な側面の構成要素の比較には，UML の Specification [19] の 11 章の Actions から 13 章の Common Behaviors（p.225～p.472）を対象とした．そこで規定されている Basic Action，Basic Activity，FundamentalActivities，Basic Behavior 等の基本概念，Diagram の規定と OONJ の構成要素との対応関係を**表 5** に示す．

まず，UML の Behavior の基本概念である Action との対応関係を示す．振舞いの基盤となる OpaqueAction，InvocationAction の 2 つは振舞いを示す中間離散単位に相当し，InvocationAction の子要素である CallAction，Call-

BehaviorAction，CallOperationAction は相互作用情報伝達を示す最小離散単位と特殊な中間離散単位に相当する．MultiplicityElement に相当する要素は存在しないが，最小離散単位を用いた日本語文で記述される．これは概念的に相当する場合である．SendSignalAction に直接相当する要素は存在しなかった．ただし，最小離散単位を組み合わせることで表現することは可能である．Pin は入出力を表すために用いられるため OONJ での相互関係表記に相当する．

Activity では，Action，ActivityEdge，ObjectNode と OONJ の構成要素は対応関係が存在した．ControlNode である InitialNode，JoinNode，MergeNode，DecisionNode に相当する構成要素は存在しないが，シナリオ等の特殊な最大離散単位や振舞いを表現する日本語記述とは概念的に相当する場合がある．その他の基本概念とは対応関係が存在しなかった．これは，OONJ はオブジェクト（離散単位）を中心とした記述を目的としており，振舞いの順序や状況を強調する Activity とは記述内容が異なるためである．

CommonBehavior においては BehaviorFeature，BehaviorClassifier が OONJ の離散単位の種類を示す指定子に相当し，Behavior は中間離散単位と最小離散単位に相当した．このように振舞いの詳細を定義する基盤である CommonBehavior と OONJ の種類を特定する指定子が対応することは，動的な側面の特性が対応することを示している．

以上から UML と OONJ 間には対応関係が存在していない振舞いも存在するが，表 5 で示すように UML の Behavior の基本要素である Action，CommonBehavior に分類される基本概念とは対応関係が存在している．さらに Action の MultiplicityElement や Activity の ControlNode については，概念的のみではあるが対応関係が存在する．そのためすべての OONJ の構成要素と対応させているわけではないが概念としては，いずれかの箇所で扱っており同等の記述が可能である．構成要素として対応しないのは，OONJ の記述目的が異なることが原因であるため妥当な差異であるといえる．つまり，動的な側面において OONJ の構成要素は UML の Behavior すべてに対しての対応関係が存在するわけではないが，両者の構成要素と基本概念の対応関係を見出すことができた．

5.4　UML のクラス図と OONJ 記述の比較

図 4，図 6 等から構成される OONJ 記述とそれらを UML のクラス図 [19] で表現した**図 11** を用いて比較を行う．UML のクラス図は，開発目標のシステムの内部構造（静的な側面）を記述するためのダイアグラムである．一方で OONJ 記述は静的な側面だけでなく動的な側面も記述される．最大離散単位とはクラスが相当し，中間離散単位とは属性と操作が相当することが分かる．また，クラスが最大離散単位に相当することで，クラスの関係は最大離散単位間における相互関連として表現することが可能である．このように，OONJ 記述中にはクラス図のクラスやメソッド，そしてクラスの関係に相当する離散単位が記述されており，静的な側面においては OONJ から見て十分な対応関係が存在するといえる．

5.5　UML のアクティビティ図と OONJ 記述の比較

OONJ の動的な側面の要素として mp と最小離散単位の振舞いがあり，それらと対比するにはアクティビティ図やシーケンス図がある．アクティビティ図はシステム内で行われるプロセスを表現したものであり，フローチャートを起源としている [20]．シーケンス図はオブジェクト間のメッセージ交換を表現することでシステム内部のメッセージの順序を簡潔に表現することが可能である．

我々が対象とする科学技術計算やシミュレーションにおいては，並列・分散化実装の有無にかかわらず，分析設計では計算手順としてオブジェクトが駆動（振舞いを起こす）する順序をとらえることが必要である．また，本研究の対象世界の分析設計では，オブジェクトの駆動順序は直線的（一連の作業順序）となるため，シーケンス図で記述してもそれはアクティビティ図と同等の記述となる．さらに，OONJ の想定ユーザは，表 1 のようなプロセスに着目した記述を初めに行うことが多いと考えられる[*9]．そこで本論文では，動的側面の記述比較としてアクティビティ図を取り上げることとする．

OONJ 記述は前節でも記述したようにオブジェクト（最大離散単位）に着目したモデリングを行っており，プロセスに着目したモデリングを行っているアクティビティ図とはモデリングが異なる．そこでアクティビティ図の記述規則の範疇で，かつ表現をできる限り OONJ 記述に近づけたアクティビティ図を**図 12** に示す．図 12 は一次元衝撃波流れの全体のプロセスを表現しており，OONJ 記述と対応関係が分かるようにスイムレーンを使用してパーティションを最大離散単位ごとに分割した．そのパーティションごとにアクションを記述し，表 1 を参考に適切なプロセスとなるようにコントロールフローを記述している．

前述したがアクティビティ図と OONJ 記述はモデリングの視点が異なるので表現が完全に一致せず，開始ノードや終点ノード等のコントロールノード等 OONJ では存在していない表現も存在する．しかし，図 12 において最大離散単位ごとにパーティションを分割することでアクションに相当する離散単位は振舞いの中間離散位と最小離散単位であることが明示的に分かり，コントロールフローは相互作用情報伝達に相当することが確実に推測される．確か

[*9] これは DSL（Domain Specific Language）のブロックダイアグラム [21] からも想起され，また想定ユーザに対象世界を説明した際にも「フローチャートが欲しい」とのコメントをいただいた経験もあり妥当な推定であろう．

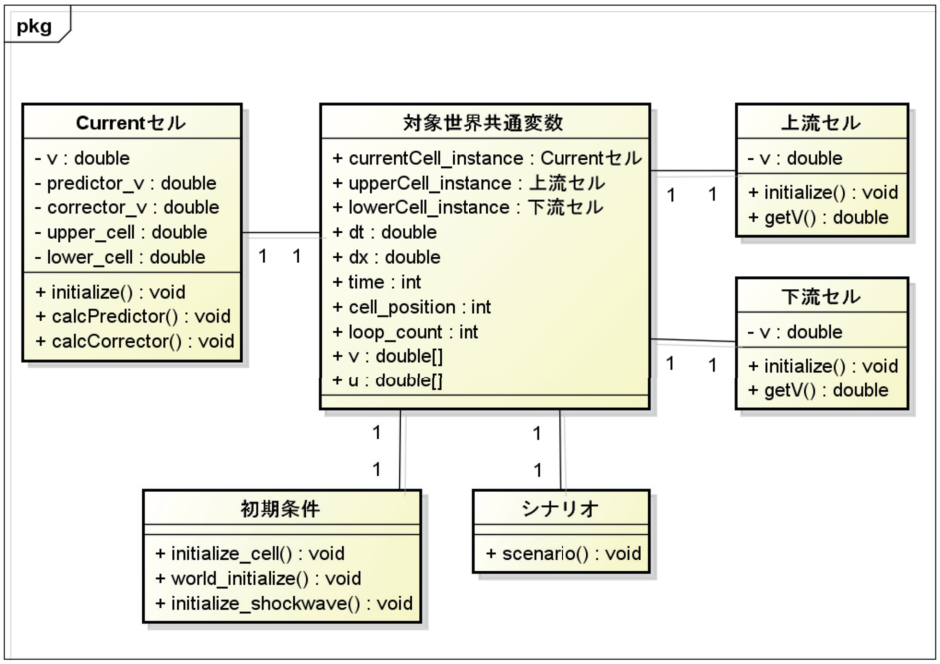


図 11　一次元衝撃波流れの対象世界のクラス図

Fig. 11　Class diagram for the target world of one-dimensional shock wave flow.

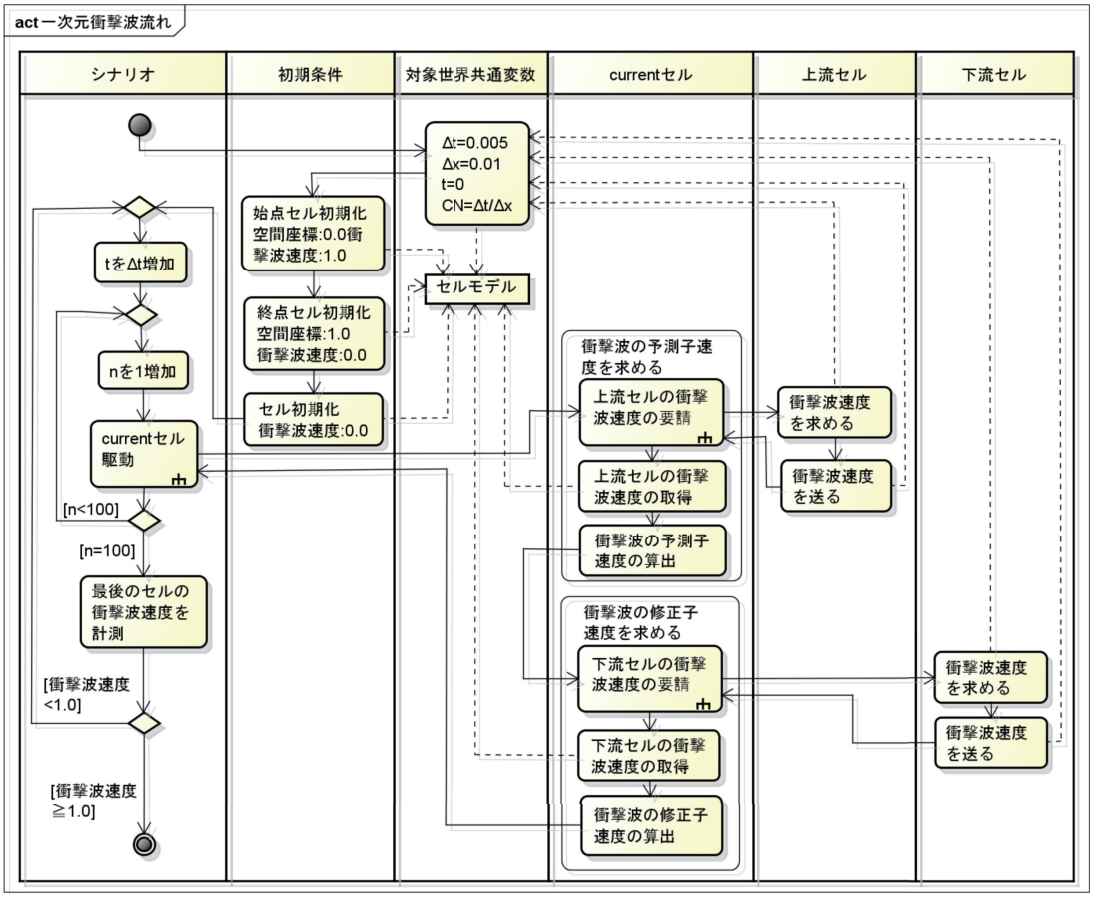


図 12　一次元衝撃波流れの対象世界のアクティビティ図

Fig. 12　Activity diagram for the target world of one-dimensional shock wave flow.

表 6　UML の各ダイアグラムの記述対象と OONJ 間の対応関係
Table 6　Correspondent relations of the descriptive targets between UML diagrams and OONJ.

ビューモデル [20]	ダイアグラムの種類	記述対象やその内容	OONJ
論理ビュー	クラス図	システムのクラス，型，インタフェースの定義等	○
	オブジェクト図	クラスのインスタンス	△
	状態マシン図	オブジェクトの状態，その変化のイベント	×
	シーケンス図	オブジェクト間の相互作用の順序	△
	コミュニケーション図	オブジェクトどうしが相互作用する方法，および，OONJ 記述で必要な接続	△
	タイミング図	オブジェクト間の相互作用のタイミング	△
	相互作用概念図	シーケンス図，コミュニケーション図を軸により広い見方を提供	△
プロセスビュー	アクティビティ図	システム内の順次アクティビティ	○
ユースケースビュー	ユースケース図	システムとユーザの間の相互作用	×
開発ビュー	コンポジット構造図	クラスやコンポーネントの内部構造	×
	コンポーネント図	コンポーネント，および相互作用するために使用するインタフェース	×
	パッケージ図	クラスとコンポーネントのグループの階層的な構造	×
物理ビュー	配置図	システムが現実の世界にどう配置されるか	×

○：OONJ 記述中に含まれる．△：想定ユーザの要求に応じて必要となる場合がある．
×：個人主体で行い，かつ小規模な OONJ 記述中にはない．

にモデリングの視点に依存はするが，コントロールノード以外[*10]の表現についてアクティビティ図は OONJ 記述の動的な側面との対応関係が十分にあるといえよう．

さらにいえば形式的な対応関係だけでなく，内容的な側面についても図 12 は OONJ 記述と同等な記述を持っていることが図 4，図 6 との比較から明白である．結論として両者は動的な側面における対応関係だけでなく，振舞いの順序関係や記述内容についても異なった表現形式ながら同等に近い記述力を持つといえよう．

5.6　UML と OONJ の比較に基づく評価と考察

5.2 節，5.3 節では静的・動的側面の構成要素の概念的比較を行い，5.4 節，5.5 節では例を用いて対象記述の比較を行った．5.2 節，5.3 節で行った概念的比較の結果，UML と OONJ の構成要素の間には，ほぼ対応関係が存在することが分かった．この結果は両言語が比較的近い記述特性を持つことを示唆するもの考えられる．

5.4 節，5.5 節で行った UML のダイアグラムと OONJ 記述との比較結果を**表 6** にまとめて示す．表 6 からただちに分かるように，対応関係があるのは論理ビューとプロセスビューである．これらはシステムの構造とそのプロセスを表現するためのダイアグラムである．OONJ の静的な側面はクラス図に相当し，動的な側面はアクティビティ図に相当する．

詳しく述べれば，OONJ 記述の最大離散単位と中間離散単位（属性），その相互関連の 3 つがちょうどクラス図に相当する．これはクラス階層まで正確に記述するとオブジェクト図とも対応する．また，中間離散単位（振舞い）と最小離散単位，その相互作用情報伝達の 3 つがアクティビティ図に相当する．このような対応関係が存在するのは，OONJ 記述が対象世界の静的・動的な両側面を含めて記述する方式であり，UML のクラス図やアクティビティ図は分離して表現する方式だからである．記述方式は異なるが，重要な記述に関して重なり部分が多い共通点を持っている．

一方，表 6 でも明らかなように，OONJ 記述と UML におけるユーザ–システム間やシステム周辺環境を記述するダイアグラムとの対応関係が薄い [22] ことが指摘できる．これらは表 2 の (4) 項のように UML と OONJ の目的が大きく異なるからである．逆に表 6 の△表記のように両者の対応関係が直接にはなかったり形式的な表記は異なったりしても，その記述対象としている内容から見ると対応可能になる場合があり，必要に応じて補充する記述をすることで対応関係を見出すことが可能になる場合もある．

以上のようにクラス図やアクティビティ図等のシステムの “構造” や “振舞い” を記述するダイアグラムは，3，4 章で述べた構造や振舞いに関する OONJ 記述とその記述内容まで含めて共通部分の多い対応関係を持つことが示された．ゆえに OONJ 記述は静的・動的な両側面において

[*10] 厳密には最小離散単位を構造化することで DecisionNode は実現されている．その他のコントロールノードはユーザが OONJ 記述中に指定をするだけで追加が可能である．

表 7　OONJ と UML の記述実験のコメントや感想
Table 7　Comments and Impressions of description experiment using OONJ and UML.

1. 両言語の対比的な位置付け
1.1　OONJ は UML との比較でいうとよく知っている世界のプログラミングに近い.
1.2　UML はまさに知らない世界を初めて学習して分析あるいはモデリングしている感じがする．UML が多面的に分析できるのは長所でもあるが，多面的に見なければ OONJ が書けないというのは想定される記述の性質を考えると短所と感じた.
2. OONJ や UML に関する学習や知識等
2.1　OONJ は記述内容さえ熟知していれば，UML と比べて必要な知識は少なく簡単に書ける.
2.2　UML は詳細で特殊な用語，記法・記号等が多くあり，十分な知識と経験がないと書けない.
3. 書き易さ，読みやすさ，理解しやすさ
3.1　OONJ は日本語が多いので事前の知識があまりなくても理解しやすく書きやすい．OONJ の構造表現はただの文字列なので UML よりは直観的には理解し難い.
3.2　UML の方がダイアグラムなので知識があればパッと見に理解しやすい．アクティビティ図が持つ全体の動的な側面の理解しやすさは OONJ にはない．記述力は強いが書き方も記法も難しい.
4. その他特別なコメント
4.1　微分方程式の差分近似計算等を支援する特別なダイアグラムやライブラリが欲しい.
4.2　UML のアクティビティ図では再帰的な振舞いや構造が書き難い場合がある.

UML とほぼ同等の記述力と記述特性を持つといえる.

5.7　OONJ と UML ダイアグラムの記述実験を通した比較

記述特性等について比較するために想定ユーザ 4 人を対象として記述実験を行った．4 人とも OONJ 記述と UML のダイアグラムに近い図は書いた経験があった．まず UML のクラス図とアクティビティ図とその周辺知識，一次元衝撃波流れの概念モデル（図 1 と表 1）とについて，想定ユーザに匹敵する程度の予備知識を与えた.

以上の準備の後に，本論文で記述例に使っている一次元衝撃波流れを対象世界とし，OONJ 記述と UML のクラス図とアクティビティ図を全員に書いてもらった．OONJ 記述とクラス図については全員が問題なく作成した．アクティビティ図についてはソフトウェア工学が専門の院生がシナリオと current セルの一部を書けたのみで，他はほとんど書けなかった.

そこで実験後に，著者たちの描いた OONJ 記述とクラス図，アクティビティ図を広げて比較しながら記述実験の院生たちと話し合った．そのうちの言語仕様に関わる比較事項を中心にまとめたのが**表 7** である．想定ユーザにとっては OONJ の方が記述に必要な知識が少なくて書きやすいが，知識があればダイアグラム表記の方が格段に分かりやすいこと等を確認した.

ほとんどが書けなかったアクティビティ図について，彼らは「結果の図を見れば理解できるが，想定ユーザの我々が自力で書くのはまず無理だと思う」と答えた．また内容的には OONJ の記述とクラス図，アクティビティ図の内容的な対応関係が（彼らの知識の範囲では）かなりの細かな部分に至るまで存在し，静的・動的の両側面について両言語ともそれぞれの形式で良好な記述特性を持って概略ほぼ同等のことが書けることを彼らも十分に理解できた.

5.8　OONJ 単独の記述実験データからの考察

本節では 2.3 節で述べた OONJ の記述力が実現できるか否かの評価を行うため想定ユーザに 500 行～1,000 行程度の記述例を作成する記述実験を行った．彼らは大学院の講義（半期，2 単位）を受けた程度の知識を前提とし，自力で勉強した分野の専門知識を使って，一般のエディタ（Word，Excel，一太郎，TeX）等を使って記述した [23].

表 8 からすぐに分かる特徴は，第 1 に対象世界の分野が多様であることであり，かつほぼすべての例についてその分野での専門性の高い内容を持っていることである [23]．これらの事実は，OONJ の実際の記述対象世界がある程度広いことを示唆すると推定できよう．第 2 はその記述量と分析記述時間の大きさである [23]．彼等は平均で 80 時間以上かけて 1,000 行近くを書いている．この事実は OONJ の記述力の強さや各分野に対する記述方法の容易さや柔軟性を持った言語であることを示唆している．また，4 章であげた相互関係の 3 項目の情報構成についてもすべての記述例で書けている．これにより十分な知識と分析力を持つ想定ユーザにとっては，表 8 で取り上げた対象世界を記述するには OONJ は必要かつ十分な記述力を持つと判断した.

以上の議論と記述データから，記述例を作成した院生たちにとっては OONJ は十分適切な記述力があり，柔軟で表現のしやすい言語でもあることを推定できる．したがって 2.3 節で設定した目的は OONJ で実現可能であることが分かった.

5.9　比較的近いその他の言語との比較・評価

想定ユーザや対象とする世界が OONJ と近い言語としてシミュレーション言語 DEQSOL [24] がある．DEQSOL は偏微分方程式の記述から差分法や有限要素法を用いて Fortran のソースコードを自動生成し，大気汚染や熱伝導等の数値的な解析に使用するシステムである．ただし数式の設定等には GUI を使っての高度にかつ専門的な方法で行い，特殊な数式や境界条件表現が必要とされる.

これと比較して，OONJ は各専門分野に特化した数式表現，たとえば差分法や有限要素法の複雑な計算スキームに

表 8　OONJ の記述例データ
Table 8　Data of description examples using OONJ.

対象世界	最大離散単位の数	記述行数	記述量（KB）	分析時間（時間）	記述時間（時間）	経過期間（時間）	記述速度
(1) 英日翻訳システムの分析	31	824	278	73 程度	94	25 程度	8.8
(2) 視覚のメカニズム	23	523	76	20 程度	40 程度	14 以上	13.1
(3) 感情ロボットの仕組み	14	809	266	65 程度	45 程度	28	13.5
(4) ヒトの免疫の仕組み	28	563	89	20 程度	40 程度	14 以上	14.1
(5) 化学プラントの制御	21	712	368	10 以下	50 程度	40 以上	14.2
(6) DNA 計算（加算）	43	1,188	234	10 以下	80 程度	25 以上	14.9
(7) 真核生物の細胞	73	1,051	229	20 程度	70	12 程度	15.0
(8) 二次元粘性流れ	13	473	194	45	18	10	16.7
(9) 基板検査装置	18	928	276	15	59	27	18.4
(10) 遠隔料理システム	15	2,703	1,321	12	59（記録）	18 以上	45.8
(11) 水の大気循環現象 (*)	36	904	147	20	40	6	22.6
平均値（整数で表示）	29	971	316	28	54	20	18

(*) 当研究室の大学院生によって書かれた．

関わるライブラリ等 [24], [25] はまだ持っていない．逆に計算式表現が可能な対象であれば必ず記述可能という汎用性に特長がある．両者は用途や記述対象に共通点は多いが，記述法や記述形式に相違点が多いことも分かった．

UML よりは OONJ に比較的近い対象世界を扱うモデリング言語として SysML [26] がある．SysML は UML の一部に特徴的な拡張をしたモデリング言語であり，大規模な組み込み系ソフトウェア開発で利用されている．しかし現在の SysML の利用形態は開発側でのドキュメント用が主であり，開発方法との関係が希薄であるため比較が困難であった．

その他に開発環境をともなう MATLAB/Simulink [21] や Mathematica [27], Modelica [28], MECANO [29] 等の言語においても比較を試みたが，記述言語ではなくプログラミング言語である等多くの点で異なっており比較が困難であった．結論として，(計算式に関わる事項を除けば) UML が最も比較の基準言語として適切であることが分かった．

6. 結論と今後の課題

本論文では，UML と OONJ の静的・動的な両側面（クラス図とアクティビティ図）からの比較を行った．比較の結果 OONJ が UML のサブセットであること，OONJ の想定ユーザと対象世界においては UML とほぼ同等の記述力と記述特性を持つことを結論できた．つまり，UML から見ればサブセットであるが，OONJ の想定ユーザから見れば必要な機能を持っており，さらに 2 つの記述実験から想定ユーザにとっては簡単で表現のしやすい記述言語であることも結論できた．

今後の課題として，実用化の観点から記述例の蓄積，記述実験，詳細な記述ノウハウやライブラリ等の蓄積がある．また OONJ の特性を活かす知識記述分野への適用を行う．

参考文献

[1] 日本計算工学会（編）：工学シミュレーションの標準手順 JSCESS-HQC002: 2011 A Model Procedure for Engineering Simulation，日本計算工学会 (May 2011).

[2] 早川義一：名古屋大学工学研究科の大学院教育—現状と課題，入手先 ⟨http://www.cshe.nagoya-u.ac.jp/publications/journal/no8/05.pdf⟩（参照 2012-04-09）.

[3] 片桐孝洋：東京大学のスーパーコンピュータを用いた並列プログラミング教育 (4)—工学部・工学系研究科共通科目，入手先 ⟨http://www.cc.u-tokyo.ac.jp/support/press/news/VOL12/No3/201005_kyoiku-katagiri-3.pdf⟩（参照 2012-04-09）.

[4] 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，シミュレーション学会誌，Vol.21, No.4, pp.295–309 (2003).

[5] 有沢　誠，斉藤鉄也：モデルシミュレーション技法，共立出版 (July 1997).

[6] 畠山正行：オブジェクト指向自然日本語構造化フレーム OOSF の設計と表現技法，シミュレーション学会誌，Vol.22, No.4, pp.195–209 (2004).

[7] 畠山正行：オブジェクト指向自然日本語記述言語 OONJ の設計とその記述力の評価，第 58 回 MPS 研究会報告，2006-MPS-58, pp.59–62 (2006).

[8] 松本賢人，畠山正行，安藤宣晶：オブジェクト指向分析記述言語 OONJ の設計原理構築と記述環境開発，第 150 回 SE 研究会報告，2005-SE-150, pp.57–64 (Nov. 2005).

[9] Object Management Group: Introduction To OMG's Unified Modeling Language, Object Management Group (online), available from ⟨http://www.omg.org/gettingstarted/what_is_uml.htm⟩ (accessed 2012-04-09).

[10] Object Management Group: UNIFIED MODELING LANGUAGE, Object Management Group (online), available from ⟨http://www.uml.org/⟩ (accessed 2012-04-09).

[11] 廣瀬直喜，池川昌弘，登坂宣好，久保田弘敏，本間弘樹：圧縮性流体解析（数値流体力学シリーズ 2），第 2 章，東京大学出版会 (1995).

[12] 峯村吉泰：流体・熱流動の数値シミュレーション，森北出版（株）(Oct. 2001).

[13] 青木　淳：オブジェクト指向システム分析設計入門，第 2 章：広義のオブジェクト指向，（株）ソフトリサーチセンター (1993).

[14] バートランド・メイヤー（著），酒匂　寛（訳）：オブジェクト指向入門，原則・コンセプト，第 2 版，株式会社翔泳社 (Jan. 2007).
[15] オブジェクト指向プログラミング講義&演習，第 2 章，OO 構造化記述法のアイデア：OOSF，入手先 ⟨http://gaea.cis.ibaraki.ac.jp/html/OOP_Lecture/OOP11-chap02.pdf⟩（参照 2012-04-09）.
[16] 三塚恵嗣，池田陽祐，畠山正行：オブジェクト指向記述言語 OOJ における汎化階層と実値化の方法，第 167 回 SE 研究会報告，2010-SE-167-20 (Mar. 2010).
[17] 畠山正行，池田陽祐，三塚恵嗣，加藤木和夫：メッセージパッシングモデルに基づく差分方程式の計算方式とその実行例，第 86 回 MPS 研究会報告，2011-MPS-86-21 (Dec. 2010).
[18] Object Management Group: Catalog Of OMG Modeling And Metadata Specifications Infrastructure, Ver2.4.1, Object Management Group (online), available from ⟨http://www.omg.org/technology/documents/modeling_spec_catalog.htm#UML⟩ (accessed 2012-04-09).
[19] Object Management Group: Catalog Of OMG Modeling And Metadata Specifications Superstructure, Ver2.4.1, Object Management Group (online), available from ⟨http://www.omg.org/technology/documents/modeling_spec_catalog.htm#UML⟩ (accessed 2012-04-09).
[20] Miles, R. and Hamilton, K.（著），原　隆文（訳）：入門 UML2.0，オライリー・ジャパン (2007).
[21] Math Works, Simulink, MathWorks (online), available from ⟨http://www.mathworks.co.jp/products/simulink/description1.html⟩ (accessed 2012-04-09).
[22] 磯田定宏：実世界モデル化有害論—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).
[23] 畠山研究室：OONJ の参考記述例，畠山研究室 (online)，入手先 ⟨http://gaea.cis.ibaraki.ac.jp/?OONJ⟩（参照 2012-04-09）.
[24] 佐川暢俊，金野千里，梅谷征雄：数値シミュレーション言語 DEQSOL，情報処理学会論文誌，Vol.30, No.1, pp.36–45 (1989).
[25] 矢川元基，関東康祐：オブジェクト指向計算力学入門，C++による数値解析プログラミング，培風館 (1999).
[26] Object Management Group: SysML (OMG Systems Modeling Language), Object Management Group (online), available from ⟨http://www.object-report.jp/omginfo/technology/sysml/⟩ (accessed 2012-04-09).
[27] ゲイロード，R.J.，西館数芽：Mathematica：自然現象の計算モデル化セルラーオートマタ・シミュレーション，（株）トッパン (1997).
[28] Tiller, M.M.： Modelica による物理モデリング入門，オーム社 (2003).
[29] 株式会社計算力学研究センター：SAMCEF MECANO，株式会社計算力学研究センター (online)，入手先 ⟨http://www.rccm.co.jp/seihin/mecano/index.html⟩（参照 2012-04-09）.

池田 陽祐　（学生会員）

1984 年生．2008 年茨城大学工学部情報工学科卒業．2010 年同大学大学院修士課程修了．同年同大学院博士後期課程入学．オブジェクト指向日本語記述言語等の研究・開発に従事．

三塚 恵嗣

1986 年生．2009 年茨城大学工学部情報工学科卒業．2011 年同大学大学院修士課程修了．同年日立情報システムズ（現，日立システムズ）入社．製造業の物流管理システム開発・保守に従事．

加藤木 和夫　（正会員）

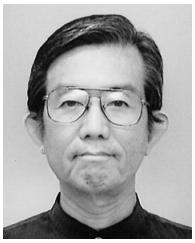
1947 年生．1970 年北海道大学理学部物理学科卒業．1971 年日立エンジニアリング入社．制御用プログラミング言語，ソフトウェア開発環境の研究・開発に従事．2002 年加藤木技術士事務所．2006 年茨城県立産業技術短期大学校．技術士（情報工学）．

大木 幹生

1982 年生．2008 年茨城大学工学部情報工学科卒業．2010 年同大学大学院修士課程修了．同年群馬工業高等専門学校教育研究支援センター．オブジェクト指向日本語記述言語等の研究・開発に従事．

上田 賀一　（正会員）

1961 年生．1989 年名古屋工業大学大学院工学研究科電気情報工学専攻博士後期課程修了．同年同大学工学部電気情報工学科助手．1990 年茨城大学工学部情報工学科講師．2002 年同大学助教授，後に准教授．2012 年同大学教授．現在に至る．ソフトウェア工学，組込みソフトウェアに関する研究に従事．工学博士．電子情報通信学会，ACM，IEEE Computer Society 各会員．



畠山 正行　（正会員）

1947 年生．1976 年東京大学大学院博士課程（航空学専攻）修了．工学博士．東京都立航空高専を経て，1990 年 10 月より，茨城大学工学部情報工学科専任講師，同助教授，後に准教授，現在に至る．この間，超音速凝縮流れ等の非連続気体流れの力学の研究に従事．次いで数値シミュレーションのユーザ向けの計算機環境，オブジェクト指向シミュレーション，オブジェクト指向日本語記述言語等の研究開発に従事．日本シミュレーション学会，日本機械学会，日本航空宇宙学会の各正会員．