

離散構造化モデルを用いた一貫したプログラム開発 実践教育へのOOJの効果的な適用

池田 陽祐^{†1} 岡田 信一郎^{†2} 上田 賀一^{†2} 畠山 正行^{†2}

概要: 本論文では、以前から多くの離散的な数値計算で用いられる方法について簡潔な離散構造化モデルを構築し、そのモデルを用いたプログラム開発環境を構築した。開発環境は分析・設計・実装・プログラムの四つの段階に分け、前三つの段階には記述言語系 OOJ とそれを利用するエディタを、四つの段階間にはトランスレータを開発した。そのために離散構造化モデルは、設計段階以降においても特化したオブジェクト指向モデルと重ねて適用する。本開発環境の特徴は、分析からプログラム取得までのプログラム開発の手法を大学の講義と演習の半期一科目分で十分に習得できるので、一貫したプログラム開発教育を実現できるという点である。この開発環境の効果を確認するために、数人の大学院生の記述実験と情報工学科三年生の授業科目に適用した。その結果、学生たちはモデルの趣旨や方法は概略良く理解し、設計以降の段階の作業は順調であったが、分析モデリングの方に時間が掛かった。記述実験の院生の方は多少の経験があったため、全く問題なく記述を行ってプログラムを得た。以上から本プログラム開発環境は大学院教育に十分効果的に適用可能であることが推定された。

キーワード: 離散構造化モデル, オブジェクト指向, プログラム開発教育

An effective apply of OOJ to integrated program development education based on the discreted and structured model

YOUSUKE IKEDA^{†1} SHINICHIRO OKADA^{†2} YOSHIKAZU UEDA^{†2} MASAYUKI HATAKEYAMA^{†2}

Abstract: A great deal of numerical calculation programs have been developed using the discreted methodologies. We have constructed a simple discreted and structured model and developed a program development environment. This environment is constructed with four inner stages, that is the analysis, the design, the implement, and the program stages. For the former three stages, the description language system OOJ and the special editor has been developed. The discreted and structured model is applied as the specilized object-oriented model after the design stage. The feature of this development environment is the simplicity. Then, the students can master in 9 times of lectures and 6 times of exercises in the University. Therefore, we have realized an integrated program development education environment from the analysis stage up to program stage. To confirm the advantage, we have achieved the description experiments by several graduate students and the third grade faculty students. The faculty students have shown a good understanding for the model and the methodology of the development environment. They have taken a good piece of time for the analysis modeling stage. The description experiments by the graduate students gotten the program easily and in a short period. In conclusion, this program development environment can be supposed to effectively be applied to the graduate school program development education.

Keywords: discreted and structured model, object-oriented, program development education

^{†1} 現在, 茨城大学大学院理工学研究科情報工学専攻
Presently with Graduate School of Information and System
Science, Ibaraki University

^{†2} 現在, 茨城大学工学部情報工学科

Presently with Department of Computer and Information
Sciences, Ibaraki University

1. はじめに

現代においては社会のインフラとして至る所に電子機器や計算機が使用され、必要欠くべからざる製品として重要性を増している。

そこで大学院教育に期待が掛かる。それは「プログラムの開発もできる専門家」への期待である。しかし情報工学に近い専攻の場合とはかく、それら以外の専攻ではその専門分野自体も大きく発展しており、プログラム開発教育に割ける時間が逆に限られる場合もある。一方で上記のようなニーズに沿った方向で大学院に期待されるのは単にプログラミング教育だけではなく、各専門分野の問題領域に適切なモデルを設定して分析モデリングし、プログラムを設計し実装するという一連の過程を適切に踏んだ数千行程度のプログラムである。しかも近年では対象世界のモデルとプログラム間で十分な精度の近似性や相似性を検証できなければならない [1]。

このような状況に対し、記述言語系 OOJ(Object Oriented Japanese) では想定ユーザを大学院生とし対象世界を科学技術計算の分野であると想定して、プログラム開発教育用のシステムを構築してきた。それは、当該分野の人たちが常用している離散的に近似された計算のモデルである離散構造化モデルを再定義・拡張して、分析からプログラムに至る複数段階の記述とプログラム作成を支援する言語系とその開発環境である。

本論文では上記のような想定で開発した記述言語系を使ってプログラム開発教育を実践してきた結果を述べて、OOJ の特徴や機能を評価・考察する。特に、OOJ がプログラム開発の専門家ではない科学技術計算分野の大学院生の教育用として短期間に成果を望める記述言語系の開発環境構築を実現できるか、という狙いを重点に考察する。

以降、第 2 章においては、OOJ の概念設計と教育目的を示す。第 3 章では OOJ の三つの記述言語と開発環境を示す。第 4 章は OOJ を用いた講義の目的と構成、また受講生の特徴を述べる。第 5 章では、講義のレポートとアンケートを評価し考察する。第 6 章で関連研究との比較を行い、第 7 章で結論を示す。

2. 離散構造化モデルと OOJ 設計の狙い

2.1 離散構造化モデルとオブジェクト指向

図 1 に従来から多く使われてきた離散的なあるいは離散化した問題領域の数値計算を行うための方法を離散化モデルという形で示した。図 1 では対象を離散化を行って離散単位として抽出する。もともと離散的に存在している単位は勿論そのまま離散単位である。離散単位の代表例は実世界での実体を持つ“モノ、物”や、概念である。

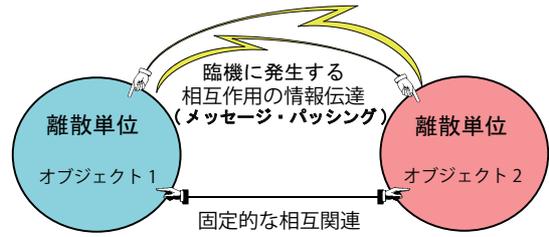


図 1 離散構造化モデル：離散化とその構造化の方法

Fig. 1 Fundamental concept of discrete modeling and its structures

離散単位間の静的及び動的な関係の表現、すなわち構造化 [2], [3], [4], [5] の方法は図 1 に示すように、離散単位間に固定的な相互関連または臨機的に発生する相互作用を付与することである*1、という方法を採用する。本論文ではこの離散単位と相互関係の二種類だけで対象世界の全てを捉えて記述するという簡潔なモデル定義を用いる。このモデルを仮に**離散構造化モデル**と呼ぶことにする。

次に実世界において定義されたこのモデルを、計算機の世界に移そうと考えたと、オブジェクト指向の概念との共通点が多いことが分かる。すなわち、離散構造化モデルはオブジェクト指向モデルを特化したモデルであることが出来よう。なぜならば、計算機の世界においてオブジェクト指向モデルは様々な拡張定義を施しても利用されるが、離散構造化モデルは図 1 で表現される定義で一意的に利用されるからである*2。

2.2 OOJ の概念設計

第 1 章で述べたように専門分野毎のモデリングの教育やプログラミングの演習を同一の講義内で行えれば、より効果的な教育が可能であろう。また、実際の作業ではモデリングからプログラミングまでを行うことは常である。これは科学技術計算の専門家向けに MATLAB[6], [7] や Modelica[8] など、モデリングを主体としたプログラム開発環境が存在することからも分かる。

また、近年ではソフトウェア開発におけるモデリング教育を PBL(Projcet Based Learning) 等を通じて行う手法が盛んに研究されている。特に赤山ら [9] は組み込みソフトウェア開発を対象にモデル駆動開発 (Model Driven Development: MDD) を適用したモデリング教育プログラムの開発を行っているが、組み込みソフトウェア向けの開発プロセスを時間をかけて実践することに意義を持つため目的が異なる。

以上と第 1 章で述べた社会的な要請と大学院生の現状を検討して、OOJ[10], [11] と呼ぶ記述言語系とその開発環境を使った解決を試みてきた。OOJ の考え方の概要を図 2 に示す。

*1 3.1 節参照。相互関連と相互作用を合わせて相互関係と呼ぶ。

*2 本節の考え方は、一般的にはまだ仮説であるので、本論文での適用結果での考察を必要とする。

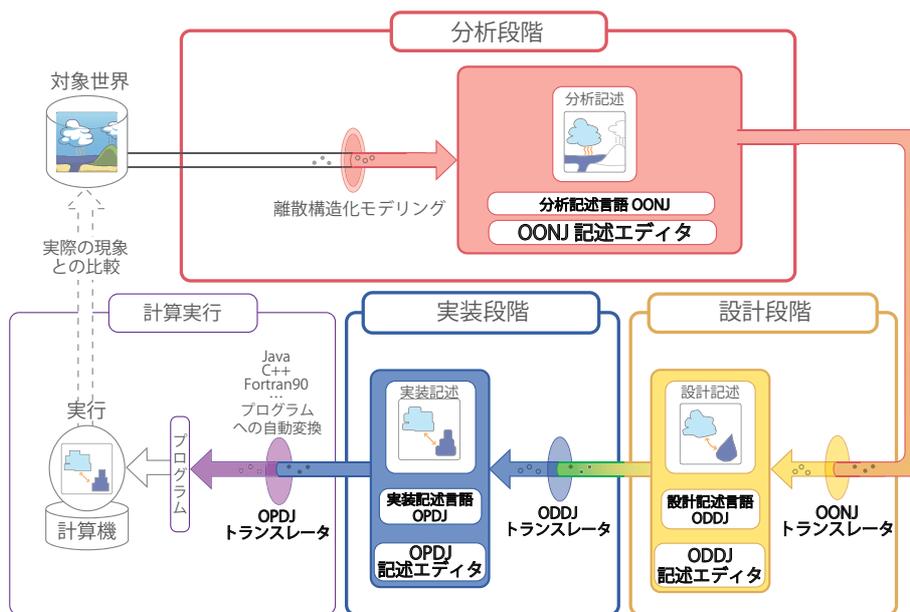


図 2 記述言語系 OOJ をベースとしたプログラム開発の過程

Fig. 2 Program development processes based on the description language series OOJ

まず、想定ユーザは図上段の分析を行う。その時には、手慣れた離散構造化モデルを用いて問題領域をモデリングし、その分析結果を記述する。ここまでの段階は実世界に対する作業である。そして、下段の設計段階からプログラムまでの作業段階に移る。

以上の様に OOJ の作業段階は分析、設計、実装の三段階に分けられている。プログラム開発のプロが行う場合は開発効率重視なので、必ずしも段階分けされていない。しかし OOJ は教育用であるので、分析、設計、実装の三段階に分けてあり、それぞれの各段階でどのような作業を行うかを詳細に規定し、2.2 節で述べた様にどちらかと言えばプログラム開発が得意でない想定ユーザでもプログラム開発の仕組みや手順を把握し、頭の中を整理しながらプログラム開発の実力を段階的に向上することを目的としている。OOJ の詳細は第 3 章で述べる。

2.3 想定ユーザと対象世界

OOJ を利用するユーザは理工系の大学院生としている。彼らは理工学分野の研究や技術開発のプロになるための訓練を受けており、一定以上のレベル [12] のプログラム作成技術を必要としている。その様な目的のために、最終的には数千行程度 (10000 行未満) の小規模のプログラムを作成出来る程度の技術の習得を達成することを目標としている*3。彼らは計算内容的に見て複雑で精密な計算処理プログラムを必要としている人たちであり、いわゆる科学技術計算ユーザという分類に入る。

(1) 彼らは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。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。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”等は既に持っている。
- (3) 多くは手続き型のプログラミング言語一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無く」、「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」のプログラム開発の経験を有する。
- (4) プログラム開発の専門家ではないので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出来ない。プログラム作成の負担は最小限にすることを望む。

2.4 OOJ の開発の全体計画と教育目的

以上の想定や課題を解決するために OOJ を用いて以下の方法を試みた。まず従来から使われてきた離散的な数値計算で使われてきた離散構造化モデルとオブジェクト指向モデルの共通性に着目し、

- 【1】 離散構造化モデルのオブジェクト指向モデルの関わりを明らかにし (2.1 節)、対象世界を正確に捉える離散構造化モデルを構築する。
- 【2】 その離散構造化モデルを正確詳細に表現するための記述言語 OONJ [10] を構築する (3.1 節)。
- 【3】 その OONJ 記述を近似性高くプログラムに反映するために複数の段階に分けて変換可能な記述言語系を作る。設計段階の ODDJ [11] と実装段階

*3 想定ユーザの裾野や周辺には、研究室配属の 4 年生、さらには実務に就いている研究者や技術者 [13] の一部も入る。

の OPDJ[11] がそれである。

[4] その言語系だけを使って分析からプログラムまでの作業を一貫して行い、

[5] 最後に実行可能なプログラムを自動変換で得る。

というのが全体計画である。この計画に基づく記述言語とその開発環境は既に実働を開始している。

すなわち、OOJ を用いた教育目的は「想定ユーザに対するのプログラム*4開発過程の理解と実力の向上」である。

本論文では、この教育目的の達成度の検証を行うために、OOJ を用いた講義を行い、その受講生の中から想定ユーザに近い受講生に対して、教育効果を事例として測定・評価し、更により効果的な適用方法について考察した。検証項目としては、以下の二項目を設定した。

検証項目 1 想定ユーザに近い受講生が OOJ を用いて離散構造化モデルを記述することが可能であるか？

検証項目 2 想定ユーザに近い受講生が離散構造化モデルを Java プログラムへ変換することが可能であるか？

3. 記述言語系 OOJ とその開発環境

3.1 分析段階の概略

前章で示した OOJ の三段階の中で、まず先頭の分析段階である OONJ の概略を述べる [10]。受講生によって記述された OONJ の記述例を図 3 を示す。

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-(A) の最大離散単位が、そして図 3-(B) 中間離散単位、そして中間離散単位の内部を構成する図 3-(C) の最小離散単位の三階層で定義される。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無く、想定ユーザ自身が決めた適切な形式で書けば良い。

また、図 3-(D) に該当するのは、各離散単位を区別するために「一連番号の組」と「種類を示す記号 (facet number, fn と略記する [3]。OONJ では特に “nfn” を用いる。)」をセットにした識別子である。

OOJ で構造を記述するには、離散単位間に相互関係を付与する形式を採用している。「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」の二種類に分けて設計した [3]。記述例を図 3-(E) に示す。

最小離散単位に収容される日本語文間の構造化には、日本語文の前に全角 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。例えば、“集約” の場合には図 3-(F) のように記述する。

3.2 設計段階と実装段階の概略

次に図 2 の右下における設計段階の概略を述べる。ユーザは分析記述に対して計算機に共通的な情報を付与し、同等内容でかつ計算機で実行可能な記述に向けて修正していく。具体的な作業としては記述した属性と振る舞いに対して、データ型の付与や日本語を数式に変換する。これらの作業が可能のように記述言語 ODDJ[11] を設計している。

最後にプログラムへ確実に自動変換するための実装段階 (図 2 の下半分の中央) では、設計記述に含まれる論理的な矛盾を取り除いたりアルゴリズムの再確認や規定された文法に基づいた数式の整理を行う。実装段階の作業が完成すればプログラムへ変換しコンパイル、実行を行い結果を得る。また、ユーザが結果を満足できない場合は、実装段階での作業、そして変換後のプログラムをコンパイル・実行を一つのサイクルとして繰り返す。実装段階で用いられるのは記述言語 OPDJ[11] である。

3.3 OOJ の開発環境

OOJ の開発環境とは図 2 に示してある各記述エディタと各段階の最後に置かれた三つのトランスレータである。開発環境の概略を図 3 を参照しながら述べる。

まず OOJ 記述エディタの画面を操作している様子を図 3 に示す。画面左上の図 3-(G) のメニューのモードを見ると分かる様に OONJ, ODDJ, OPDJ の三つのモードがあり、三つの記述言語の編集が一つの OOJ 記述エディタで出来る様になっている。画面左下の図 3-(H) には、各離散単位の情報が表示される。したがって、ユーザは OOJ の三つのサブ言語というイメージで利用することが出来る。

同じ画面形式が実装段階まで続くということは、設計段階以降でも離散構造化モデルのイメージでユーザは編集作業が出来ることになる。これにより、想定ユーザは開発環境の使い方を理解するだけで、離散構造化モデルからオブジェクト指向プログラムへの変換まで行えるよう設計した。

4. OOJ を用いたプログラム開発教育の計画

4.1 想定ユーザと受講生の差異

そこで「オブジェクト指向プログラミング」の講義に OOJ を採用する機会があり、受講生である大学学部三年生 (以降、受講生) を対象として OOJ の教育目的の達成度を調査した。

ただし、受講生は OOJ の想定ユーザとは以下の点で異なる。まず、想定ユーザは理工学分野の研究や技術開発が専門分野であるが、受講生はプログラミングと専門科目の入り口に居る。また想定ユーザは対象世界に関する知識は十分にあり、自身が行いたい科学技術計算の問題を正確に捉え、モデリング方法も理解している。その一方で、受講生はプログラミングの知識は想定ユーザよりもある。

*4 厳密には、科学技術計算で用いられるプログラムであり、ソフトウェア工学等が対象としている大規模でエンタープライズ向けのプログラムではない [14]。

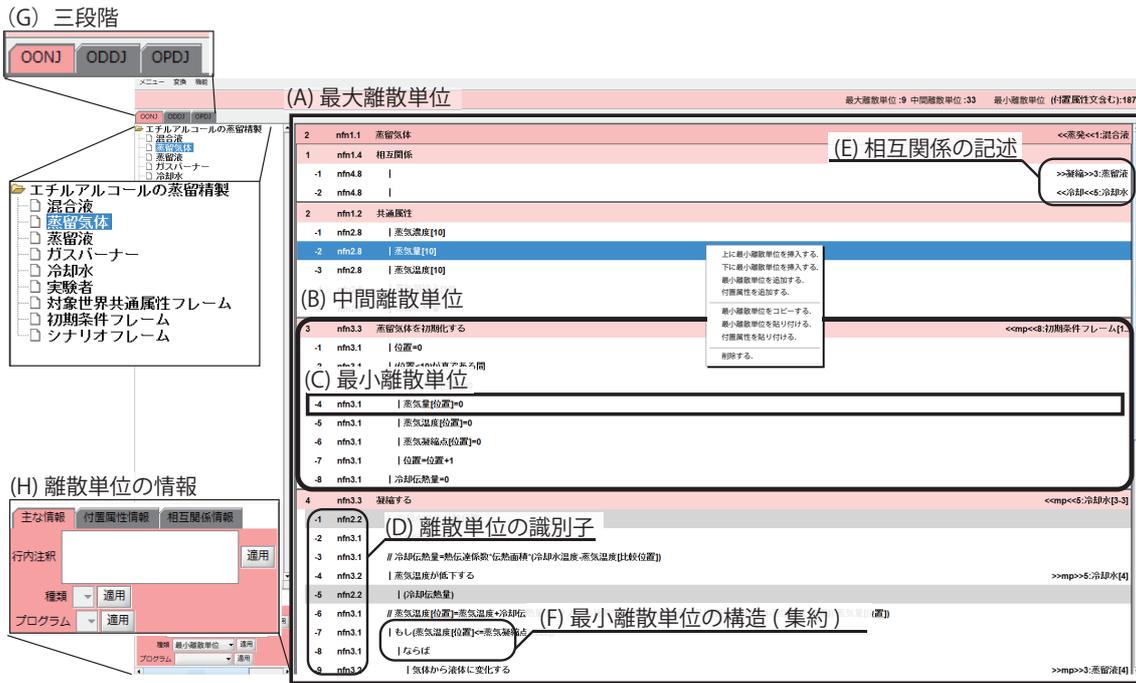


図 3 OONJ の記述例:エチルアルコールの蒸留
 Fig. 3 Description example of OONJ for distilling ethyl alcohol

4.2 講義と演習の構成

まず、講義の目的はオブジェクト指向モデリングの理解である。ただし、ソフトウェア工学の分野で通常用いられているオブジェクト指向モデリング [14] など高度な技術ではなく、その前提となるオブジェクト指向モデリングの基礎である。講義の構成は、始めにオブジェクト指向パラダイム等の基礎概念の説明を行い、OOJ の各段階の説明と演習を交互に行い、離散構造化モデルをプログラムへ変換する作業を促す。

4.3 レポート課題の構成

レポート課題は四回構成とした。まず対象世界を選択させ、日本語で対象世界の分析記述を課した。つぎに第二回目では日本語や何か適当な記号を使うことを許し離散構造化モデルを用いた記述を書かせる。最後に OONJ 記述から Java プログラムまでを纏めた最終レポートを課した。

このレポートの構成は、想定ユーザが対象世界内部の問題領域を選択して、分析し、OONJ 記述に纏めるとともに、プログラム開発に向かって動く手順と全く同じように構成してある。したがって三年生の受講生が、プログラム開発の課程に自然に入り込んでしまう様に構成した。

5. OOJ のプログラム開発教育の実施結果

5.1 評価対象とする受講生の設定

2.4 節で示した検証項目の達成状況を調査するために想定ユーザに近い特徴を持つ受講生を選択する。

選択項目は受講生への【離散構造化モデリングの理解度】

のアンケートと OONJ 記述の評価の 2 点とした。まず、離散構造化モデリングのアンケート結果から離散構造化による分析・設計を理解できたと回答した 11 名の受講生を対象とした。次に該当した受講生の最終レポートの OONJ 記述が、ある程度の離散構造化モデルとして記述されているかを調査し、10 名を選択した。

また、別の記述実験を 4 名の大学院生に行った。大学院生は講義には出席せずに、一昨年からの知識と経験を基に幾つかの記述例を参照しながら Java プログラムへの変換までを行ってもらった。この 4 名も想定ユーザとして含める。以上の選択方法から OOJ の評価を行なう 14 名を設定した。

5.2 最終レポートの分析から見た講義の達成度の評価

2.4 節の検証項目 2 である「受講生が離散構造化モデルを Java プログラムへ変換することが可能であるか？」の達成状況を検討するために各最終レポート毎に記述を検討することで調べた。対象としたのは前節で示した受講生の最終レポートと大学院生の OOJ 記述である。以降、本節で用いる受講生は前節で設定した受講生を指す。

表 1 に示している結果から全ての受講生が実装段階以降まで達成できていた。また実装段階で止まった受講生の場合は、細かい文法事項 (例えば数式等の計算処理の文法) が規定通りでないとかエディタの使い方のミス等で Java プログラムへの変換が出来ていないことが殆どなので、これら二つの段階の受講生は講義の目的を達成したと見なして

表 1 OOJ 記述の達成状況とその要因

Table 1 Status of the achievement of OOJ descriptions and their causes

達成状況	レポート	達成できない原因
実装段階まで	6 通	文法に従っていない
Java プログラム	8 通 (大学院生が 4 通)	

良いと考える。大学院生においては、モデリングの詳細度について幾度か打ち合わせを行うことで Java プログラムを得ることができ、また目的とした実行結果を得ていた。

OOJ 記述の完成に使用した時間を比率で回答してもらったところ分析段階が 6、設計段階が 2、実装段階が 2 となり、分析段階が半分以上を占める結果となった。

全ての受講生が実装段階以降まで到達できたことから設計段階以降の操作を十分に可能であったことが分かる。また、分析段階の記述時間が半分以上を占めたことは、記述方法の理解や離散構造化モデルの構築が問題であったことが分かる。この結果は次節の受講生のアンケートによっても裏付けられる。また、この結果は 2.4 節で示した検証項目 1, 2 に該当する。全ての受講生が実装段階以降まで達成できたことは、OOJ を用いて離散構造化モデルを記述することが可能であり (検証項目 1)、離散構造化モデルをオブジェクト指向プログラムへ変換できたこと (検証項目 2) を表しているといえよう。

つまり、講義 (9 回) と演習 (6 回) 合わせて 15 回の回数の中で十分とは言えないまでも、ほぼ講義目的は実現できる程度の講義の構成 (第 4.2 節) であったことが分かる。OOJ の説明自体は 9 回の講義で終わることが可能であるため演習時間を取れる点も重要であろう。

以上から少なくとも、受講生に対しては自然現象を対象世界とした離散構造化モデルからオブジェクト指向プログラムへ変換を理解し、実行できたと言える*5のではないかと。

5.3 アンケート結果からの考察

最終レポート提出時に受講生に対して、特に分析段階以前の項目についての複数回答の選択形式のアンケートを行った。表 2~表 5 に各アンケート項目を示し、5.1 節で示した受講生を対象とした結果と考察を示していく。以降、本節で用いる受講生は 5.1 節で設定した受講生を指す。

【本講義の全体的な感想】を二つの項目を設けてアンケートを行った。講義で全体的に得られたことに関するアンケート結果を表 2 に示し、苦勞したことに関するアンケート結果を表 3 に示す。表 2、表 3 の結果からオブジェクト指向の概念の理解やオブジェクトの抽出は理解できていると判断しているとの回答が多かった。また、苦勞した点では OOJ 記述エディタの操作性や記述法の理解、対象

*5 次節で示すコメントの一つが「訳すように感じた」という表現からも推測できる。

表 2 アンケート結果 (1/4)

Table 2 Questionnaire results (1/4)

アンケート項目	
この授業で得られたと思うことを挙げてください。	複数回答可
回答項目	回答数
オブジェクト指向の概念に対する理解	10
オブジェクト抽出に対する理解・経験	9
オブジェクトの属性抽出に対する理解・経験	3
オブジェクトの振る舞い抽出に対する理解・経験	5
オブジェクト間の相互関係抽出に対する理解・経験	5
オブジェクト指向プログラムの設計・実装に対する理解・経験	5

表 3 アンケート結果 (2/4)

Table 3 Questionnaire results (2/4)

アンケート項目	
この授業で苦勞したと思うことを挙げてください。	複数回答可
回答項目	回答数
オブジェクト指向の概念理解	3
対象世界の調査・理解	6
OOJ 記述法の理解	8
OOJ エディタの操作	8
オブジェクトの抽出作業	1
オブジェクトの振る舞い抽出作業	3
オブジェクトの属性抽出作業	3
オブジェクト間の相互関係抽出作業	5
プログラムへの変換に関わる作業	5
レポートの作成	1

世界の調査など、本講義で新たに学んだ内容について苦勞したとの回答が多かった*6。しかし結果としては、記述式の試験の平均点は 72 点と十分なスコアを取っており、新規なあるいは類例の少ないプログラム開発法の理解に苦勞はしたが (前章の成果から見れば、) 理解は出来たと分析出来るのではないかと考える。

以上のことから受講生は 4.2 節で示した目的は達成できたと推測できよう。

【OOJ を用いた離散構造化モデリングの使い勝手】について項目を設けて質問してみた。その結果を表 4 に示す。使い易いと答えた受講生と使いにくいと答えた受講生の二つに分かれる結果となった。使いやすいと答えた受講生のコメントには、「日本語を用いて理解するためクラスなどの文法のような他のことに気を使わなくて良い分理解しやすかった。」、「OONJ は日本語から英語に訳すようなものだと感じ、例えば Java は英語をそのまま使うようなものだった。」等の日本語を用いることによる評価があった。

一方で、使いにくいと答えた受講生は、「考え方をしっかり身につけるまでに時間が掛かったため、苦勞した思い出

*6 これは演習の際の質問が離散単位の書き方、記述エディタの操作等が多かったことから妥当な回答だと考えられる。

表 4 アンケート結果 (3/4)
Table 4 Questionnaire results (3/4)

アンケート項目	
OOJ による分析・モデリングは使いやすかったですか？	
回答項目	回答数
大変使いやすかった	1
幾つか抵抗感を感じたが、使い勝手に問題はなかった	6
使いやすくないが、問題なく使えた	1
使い勝手が悪いが、なんとか使えた	4
使いにくかった	0
未回答	2

表 5 アンケート結果 (4/4)
Table 5 Questionnaire results (4/4)

アンケート項目	
この授業以前にモデリングを行った経験はありますか？	
回答項目	回答数
経験があり、プログラム開発時にはよく利用している	0
経験はあるが、あまり利用していない	2
勉強した経験はあるが、実際に行った経験はない	4
勉強の経験も行った経験もない	6
未回答	2

が強く残ってしまった。だが、記述の仕方を理解した上で使いやすい OOJ 記述エディタ・環境があれば非常に使いやすくなる可能性も感じた。」等のコメントがあった*7。

本講義では離散構造化モデリングの使い勝手は、記述作業まで含むことは受講生のコメントからも分かる。そのため OOJ 記述エディタへの慣れやユーザインタフェースの完成度も関係してくるので結論は出しにくいですが、本講義の目的の達成に支障はない程度には使えたとは言っても良いであろう。

【分析・モデリングの経験】について項目を設けて質問した。受講生のこれまでの経験の実績を表 5 に示す。多くの受講生は、モデリングの経験が無いと回答した。また、演習中やアンケートのコメントから経験がある受講生は UML を用いたソフトウェア工学のモデリングであり、自然現象に対しての分析やモデリングの経験は無かった。この結果から受講生は、自然現象の分野の知識が少ないにもかかわらず OOJ の基礎概念や離散構造化モデリングは十分に理解可能であり、実際に講義目的を達成できるほどのモデリング経験を得ていることも推測できる。

5.4 想定ユーザに対してのより効果的な適用方法の考察

前節までのレポートの評価とアンケートの考察から 5.1 節で設定した受講生に対しての効果を五項目に纏めた。

*7 そもそも他のモデリング言語の経験が無い受講生が大半であるため実際の作業量や苦勞した点の印象が結果に出ていることも考えられる。

項目 1. 大半の受講生がオブジェクト指向パラダイムを修得できたと感想を示した。

項目 2. 半数以上のモデリングを行ったことの無い受講生が離散構造化モデリングを理解できたと回答した。

項目 3. プログラムへの変換作業は大半の受講生が行うことができおり、それは実装段階以降まで達成したことから分かる。また、設計段階以降の作業は記述エディタの影響を受けるようである。

項目 4. 対象世界をモデリングする作業が受講生にとって問題であったようである。

項目 5. OOJ 記述エディタに対しての評価は二極化していた。

まず、本講義はオブジェクト指向パラダイムを理解することが目的であり、項目 1 で大半の受講生が理解できたと回答したことで、講義の目的は達成できたといえる。2.4 節で示した検証項目に対しては、項目 2 を満たす受講生においては、項目 3 から達成できたと推測できる。5.3 節の考察から項目 4、項目 5 の問題も明らかになった。

以上の結論から 4.1 節で示した想定ユーザと受講生が異なる点について考察していく。

【専門分野が異なる点】と【対象世界に関する知識が異なる点】は、分析段階の離散構造化モデルの記述内容と関係する。想定ユーザは、問題領域が明確であるため対象世界を調査しながら記述していた受講生よりは、十分に離散構造化モデルを記述することが可能である。

【プログラミングに関する知識が異なる点】は、設計段階以降の作業と関係する。受講生に比べ想定ユーザはプログラミングに対する経験と知識が少ないため受講生よりも問題になると考えられる。ただし、設計段階以降の作業は OOJ 記述エディタの操作性との影響が強いことが分かっている。これに関しては、例えばアンケート等のコメントから多くの受講生はこの段階間の変換操作をバグ取りと類似した作業として捉えていることから、これに相応しいインターフェースを設定することで想定ユーザに対しても設計段階以降の負担の軽減が期待できる。

【モデリング方法に関する知識が異なる点】は、離散構造化モデルの構築と関係する。モデリングを理解していない受講生に比べて、想定ユーザは離散構造化モデリングを理解することは容易であろう。また、本講義では予め受講生がモデリング経験が無いことを想定できたため前半でモデリングの基礎から離散構造化モデリングの講義を追加している。想定ユーザの場合はこの時間を設計段階以降の講義とすることで、より想定ユーザに対してより効果的な講義構成となろう。

以上のことから OOJ をより効果的に運用するには、開発環境の向上と想定ユーザに合わせた講義構成を適切に設定することが必要であるといえよう。

6. 関連他研究との比較評価と考察

本章はプログラム開発教育と想定ユーザ向けのプログラム開発環境との二つの側面から比較を行う。

6.1 オブジェクト指向言語ドリトルとの比較

プログラム開発教育用に広く用いられているオブジェクト指向プログラミング言語として「ドリトル」[15]がある。

ドリトルは、初学者向けのプログラム開発教育を目的としており、中学生を対象とした情報教育や企業における文系の新人教育に用いられている。そのため学校教育で容易に扱うように設計されている。プログラミングの方法は専用エディタを用いて画面上のオブジェクトにメッセージを送る形で行われ、メッセージは日本語と簡潔な構文を用いて記述される。これにより、ドリトルは教育に適したプログラミング言語である事が分かる。

その一方で OOJ は特定のプログラミング言語を用いずにモデリングを中心としたプログラム開発を行うため、ドリトルとは教育方法が異なることが分かる。

以上からドリトルと OOJ は教育目的と方法は異なり、プログラム開発教育の目的毎に適切に用いる必要がある。

6.2 Modelica との比較

想定ユーザの専門分野に特化したプログラム開発環境として Modelica[8] が存在する。

Modelica は工学分野のモデルの振る舞いを表せるモデリング用言語*8であり、Dymola, Omola, ObjectMath 等のモデリング用言語を統一することで開発されている。そのため電気回路、油圧系等の応用分野から微分方程式、ペトリネット等の多くのモデリングや表現形式を用いることが可能である。それらを実現するために大規模なライブラリを構築しており、テキスト記述を用いて実装を行う。

Modelica に比べて OOJ の想定ユーザの専門分野は限定しており、さらに Modelica はよりプロユース向けに開発されており、教育用として開発されている OOJ とは目的が異なる。そのため Modelica への入門用と位置づけて、Modelica を学習する際に OOJ による教育を事前に行うのも適切な方法であることが考えられる。

7. 結論と今後の課題

本論文では、2.4 節で示した教育目的の達成度の検証を行うために、第 4 章で示したオブジェクト指向プログラミングの講義に OOJ を用いた。レポート評価やアンケート結果から 5.1 節で示した受講生はオブジェクト指向パラダイムを理解できたことが分かった。また、OOJ によってモ

デリングからプログラミングまでを一貫して行う教育方法が可能であることも分かった。

以上の結果を考察したところ、想定ユーザに対しても適切にモデリングとプログラミングの学習負担を調整した講義を実現することで、同等の効果を示すことが可能であると推定された。

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 OOJ 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] ASME NAFEMS, What is Verification and Validation?, 2009-July-24.
- [2] S. シュレイアー, S.J. メラー, 本位田真一, 山口亨訳, オブジェクト指向システム分析 上流 CASE のためのモデル化手法, 啓学出版, 1990 年.
- [3] 島山正行, オブジェクト指向分析自然日本語構造化フレーム OOSF の設計と表現技法, 日本シミュレーション学会誌, Vol.22, No.4, pp.195-209, Dec., (2004).
- [4] 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).
- [5] 峯村吉泰, 流体・熱流動の数値シミュレーション, 森北出版株式会社, ISBN4-627-91751-1.
- [6] MathWorks MATLAB(online), 入手先 (<http://www.mathworks.co.jp/products/matlab/>), (accessed 2011-12-26).
- [7] MathWorks Simulink, <http://www.mathworks.co.jp/products/simulink/description1.html>, (access 2011/11/15).
- [8] Michael M. Tiller, Modelica による物理モデリング入門, オーム社, 平成 15(2003 年) 年 11 月.
- [9] 赤山聖子, 久保秋真, 久住憲嗣, 二上貴夫, 北須賀輝明, ソフトウェアモデリング教育におけるモデル駆動開発の活用, 情報処理学会研究報告, Vol.2012-CE-113, No.5, (2012).
- [10] 池田陽祐, 三塚恵嗣, 加藤木和夫, 大木幹生, 上田賀一, 島山正行, 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, 情報処理学会論文誌: 数理モデル化と応用 Vol.34, (2012 年発行予定).
- [11] 大木幹生, 片野克紀, 三塚恵嗣, 沼崎隼一, 浦井智寛, 加藤木和夫, 池田陽祐, 島山正行, 三言語独立のオブジェクト指向記述言語 OOJ の実装と検証, 第 163 回 SE 研究会報告, 2009-SE-163, pp.49-56, (2009).
- [12] 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), (accessed 2012-03-12).
- [13] 島山正行, 池田陽祐, 三塚恵嗣, ドメインユーザ・プログラミング: 記述言語をベースとする仕組みの提案, 第 125 回 HPC 研究会報告, 2010-HPC-125, (2010.06.17).
- [14] 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).
- [15] 兼宗進, 御手洗理英, 中谷多哉子, 福井真吾, 久野靖, 学校教育用オブジェクト指向言語「ドリトル」の設計と実装, 情報処理学会論文誌: プログラミング, Vol.42, No.SIG 11, pp.78-90, (2001).

*8 UML や SysML と区別するためにモデリング用言語 [8] と記述した。