

離散構造化モデルに基づく開発環境 OOJ を用いた プログラム開発教育の実践と評価

岡田 信一郎^{1,a)} 畠山 正行^{2,b)} 池田 陽祐^{2,c)}

概要: 現在, ソフトウェア開発に携わる技術者にとってオブジェクト指向によるプログラム設計・開発技能を習得することは必須である. この技能の向上を目的とした多くの教育手法が提案されているが, そのほとんどは UML を必要とするため, そのような手法を用いる学習者は UML を習得しなければならない. 筆者らはオブジェクト指向の最も重要な要素であるオブジェクトとメッセージパッシングを重視し, より簡易に扱えるようにした離散構造化モデルに基づくプログラム開発環境 OOJ の開発を行っている. 本論文では, OOJ を用いてプログラムの設計・開発技能の向上を図る授業方法を検討し, 茨城大学情報工学科の3年生向け科目において実践を試みた結果を報告する. この実践における受講者のほとんどはオブジェクト指向によるモデリングの経験がなかった. しかし, 受講者から提出されたレポートを検証した結果, プログラムを動作させられなかった受講者を含めたほとんどの受講者が設計までは妥当な記述を行っていたことが確認された. したがって, OOJ を用いた授業は受講者のプログラム設計・開発技能の向上に貢献したと考えられる.

キーワード: OOJ, プログラム開発教育, オブジェクト指向, 離散構造化モデル, モデリング教育

Practice and Evaluation of Program Development Education using Program Development Environment OOJ based on Discrete and Structure Model

OKADA SHIN-ICHIROU^{1,a)} HATAKEYAMA MASAYUKI^{2,b)} IKEDA YOUSUKE^{2,c)}

Abstract: Recently, the skills for the program design and development based on the object oriented paradigm must be mastered by engineers who are engaged in software development. Many education methods for the purpose of improving these skills have been proposed. But, because most of the methods needs the UML, the students using such methods must study the UML. We have proposed the discrete and structure model that is based only on the some definitions of the object and the message passing. To attain the proposal, we have developed the OOJ that is the program development environment based on this model.

In this paper, we will report some results of considering the teaching plan using the OOJ. We will report some results of applying the plan to an actual lecture for students in the Ibaraki University. Before starting this lecture, the students had not experienced any modeling technique using the object oriented paradigm. After the lecture, we had examined all reports submitted by the students. As the results, we have confirmed that the students have completed more reasonable design. We have concluded that the teaching plan using the OOJ contribute to the improvements of student's skills for the program design and development.

Keywords: OOJ, program development education, object oriented, discrete and structure model, education of modeling

¹ 茨城大学
Ibaraki University, Hitachi, Ibaraki 316-8511, Japan
² (株) ソフトソリューションズ

Soft Solutions Inc., Tsukukba, Ibaraki 305-0047, Japan
a) s-okada@mx.ibaraki.jp
b) m.hatakeyama@ssi-ooj.com
c) y.ikeda@ssi-ooj.com

1. はじめに

近年、オブジェクト指向に基づくソフトウェア開発は一般的なものになっている。そのため、ソフトウェア開発に携わる者には、オブジェクト指向の概念に精通していることが求められる。ここで言うオブジェクト指向への精通とは、単にオブジェクト指向プログラミング言語の文法を知識として持っているということではなく、現実世界の事象のデータやそれらの相互の関係をコンピューター上で効率よく扱うために、過不足なくソフトウェア部品であるオブジェクトへ変換する技能、所謂モデリングの技能を有することである。

オブジェクト指向プログラミング言語の教育は、すでに大学等の教育機関においては一般化している。さらに、オブジェクト指向によるソフトウェア開発全体の教育支援についても、UML そのものの学習支援 [1] から、CASE ツール、MDD を利用したモデリング学習の支援 [2]、さらには PBL[3] の手法を用いてオブジェクト指向に基づくソフトウェア開発プロセスを一通り実践するような教育 [4] など多くの提案が行われている。これらの事例のほとんどは、オブジェクト指向によるモデリング・設計記述法のデファクトスタンダードである UML[5] の使用を前提としており、モデリングや設計の結果を正しく UML で記述できる技能の習得が教育目標となっている。一方、筆者らが着目する教育目標はオブジェクト指向によって開発対象の本質を捉える技能の習得であり、UML 習得の前段階に相当する。しかし、そのような段階に対する教育手法の提案はほとんど例がない。

さて、オブジェクト指向に基づいてプログラムを作成する場合、現実世界（対象世界）の分析、オブジェクトの設計、実装といった段階を経ることになる。それらの段階が適切に行われた場合、分析、設計、実装のそれぞれは視点や表現が異なるだけで同じ対象の同じ性質を扱っていることになる。これを分析-設計-実装の間の一貫相似性 *1 と考える。この一貫相似性を保った分析-設計-実装を行うためには、それぞれの段階が何を表現すべきであるのかを理解した上で、さらにそれらの表現の間に存在する一貫相似性を理解していなければならない。

筆者らの一部はシミュレーションプログラム用の開発環境 OOJ の提案、開発を行ってきた [6][7][8][9][10]。OOJ のユーザーは専用のエディタ、トランスレータを用い、分析、設計、実装を日本語や数式で記述することで、オブジェクト指向プログラミングに関する知識を持たなくとも、実際に動作するシミュレーションプログラムを作成することができる。

OOJ を用いたプログラム開発では、分析段階から実装段

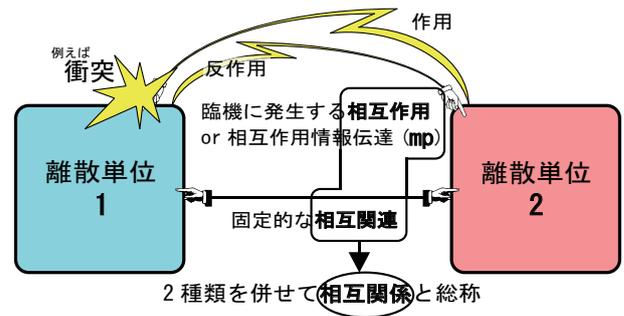


図 1 離散・構造化モデルの模式図

Fig. 1 A pattern diagram of discrete and structured model

階までの記述の中の一貫相似性が保たれる。そのため、モデリング経験の浅い学習者に、OOJ を用いたプログラム開発を経験させることで、一貫相似性の存在を認識させることができ、モデリング技能を含むオブジェクト指向開発技能の向上につながるのではないかと考えた。

そこで、Java 言語によるプログラミング経験はあるが、モデリングの経験は浅い本学情報工学科 3 年生を対象に、OOJ を用いたシミュレーションプログラム開発演習を中心とした授業を行い、どのような教育効果があったのかを検証した。本論文ではその結果を報告する。以降、2 章で OOJ の概要を説明した後に、3 章で授業および演習の内容、レポートの評価結果、アンケートの集計結果を示し、4 章で教育効果に関する考察、5 章で関連研究との比較について述べる。

2. OOJ の概要

2.1 離散構造化モデルとオブジェクト指向モデル

プログラム開発において最初に必要なのは、対象世界をどう捉え・どう分析し・どう記述するかという点の方法である。そのためにはシンプルな「対象世界捕捉モデル」が必要である。OOJ では時間や空間の「離散化モデル」を、そして離散化された「離散（化）要素」を対象世界と同じ構造を持たせるための構造化も導入した。つまり従来科学技術計算で良く使われている離散化モデルだけではなく、構造化という特性も捕捉モデルに持たせてそれらを 1 つのモデルで表す。これを離散・構造化モデルと呼んでいる [11][12][13][14]。

離散・構造化モデルを模式的に図 1 に示す。図の左右にある正方形は離散化された単位（離散単位）を表す。例えば空間を離散化した立方体のセル等がそれに当たる。2 つのセルは当然ながら何らかの関係性を持ちながら相互に影響する。本モデルではそれを相互関係で表現する。相互関係は図 1 の様に 2 種類あり、相互関連と相互作用である。相互関連は固定的な関わりで、例えば 2 つのセルが隣接関係で固定されている場合がそれである。もう 1 つは相互作用であり、臨機に（有限時間内で）発生し相手に影響を与える。例えば、隣接セルから空気が流入したり、熱を貰っ

*1 一貫相似性の意味は 2.3 で明確にする

表 1 離散 (化) モデル, 離散・構造化モデル, OO モデルの比較
Table 1 Comparisons among discrete model, discrete and structured model, OO model

離散モデル	離散・構造化モデル	OO モデル
配列の 1 単位 変数 計算処理 (非明示的) × ×	離散単位 属性 振舞い 相互関係 (図 1) × ×	クラス, インスタンス 属性 メソッド, ステートメント 関係 カプセル化 (情報隠蔽) 継承
(非明示的) 計算処理 × ×	集約 (相互関連) 相互作用 (図 1) × ×	集約 メッセージ/シグナル ポリモルフィズム 動的結合

たり, 押したり引いたり, という相互作用がそれに当たる。
 すなわち従来からしばしば用いられてきた離散化モデルに構造化という要素も加えた 1 つのモデルとなっているのが特徴である。また, この離散・構造化モデルは離散化モデルの拡張であると共に, オブジェクト指向モデルとも多くの共通点といくつかの異なる特徴を持つ。3 つのモデルの比較を表 1 に示す。

- (1) 3 つのモデル共に離散単位 (或いは相当要素) とその離散単位に属性と振る舞いを持っている。
- (2) 離散構造化モデルで言う 2 種類の相互関係を三つのモデル共に, 何らかの形で持つ。
- (3) オブジェクト指向モデルが持つプログラミングに関する要素を離散モデルと離散・構造化モデルは持たない。
 つまり離散・構造化モデルは, プログラミング技術の要素を外し, 対象世界をシンプルにモデル化・分析・記述するためだけに設計された。しかしこのモデルはプログラム開発においても有用でなくてはならない。そのため OOJ では, プログラミング技術に関わる事項は別途に供給する。

2.2 OOJ の構成と記述言語の特徴

OOJ ではプログラム開発を 3 つの段階に分けており, 各段階に記述言語を規定している *2。それらの段階と記述言語である OONJ, ODDJ, OPDJ, そして OOJ の全体構成を図 2 に示し, 各記述言語の分担と特徴を表 2 に示す。OONJ, ODDJ, OPDJ は全て離散・構造化モデルを基盤とした記述言語であり, 記述言語間の関係は変換規則として定めている [7]。

- (1) 分析段階の記述言語 OONJ は表 2(A) の条件, すなわち離散単位と構造に忠実な記述を作成できる言語に設計した。
- (2) 設計段階言語 (ODDJ *3, 同表 2(B)) は, 離散・構造化モデルを忠実に引き継ぐと共に OONJ 記述との相

*2 3 つの記述言語に分割したのは 2.1 節でも示したように実際のプログラム開発の段階化, それによる分析からプログラムに至る全段階の追跡の形式化を行うためである。

*3 Object-oriented Design Descriptive Japanese

似性を確保・保証しつつ, 計算機処理できる記述規則に設計する。新たに記述されるデータ型とアクセス属性は相似性を崩さない *4。

- (3) 実装段階言語 (OPDJ *5[10], 同表 2(C)) オブジェクト指向プログラミング言語に関わる事項はこの段階で扱う。

2.3 OOJ の特性: 一貫相似性

前節で述べた OONJ, ODDJ, OPDJ の各記述は同一の離散・構造化モデルで表現されるため記述 *6 間に相似性 *7 と呼ぶ特性がある [7]。そして, 分析段階から実装段階まで一貫した相似性を一貫相似性と呼ぶ。この特性により記述間の比較作業による開発過程の妥当性を形式的に検討することができる。

実際の作業は, ある離散単位内に書かれた内容だけを記述間で比較する。そのため取り付きやすい作業であり, 経験が浅い人でも記述の誤りや矛盾の早期の検出, 設計の妥当性の検証などの各種問題へも簡易に取り組むことができる。

2.4 OOJ の記述例: OONJ の記述法とその構造

図 3 は後述する OOJ エディタで, 「アルコールの蒸留」を対象世界とした記述を行っている例である。この記述例を用いて OONJ の記述法とその構造を説明する。この記述例において最も外側の枠線は最大離散単位である (A-1)。これはオブジェクト相当, あるいは実世界の “もの, モノ, 物, 者” に相当する単位であり名前が必要である (A-2)。最大離散単位には複数の中間離散単位が含まれる。中間離散単位には最大離散単位間の相互関係や内部の共通属性 (つまり, 変数相当), 纏まった振る舞いなどのメソッドやフィールドに相当する単位が記述される。中間離散単位も最大離散単位同様に枠線があり (B-1), 纏まった振る舞いの総称名 (B-2) を記述する。

中間離散単位の内部には最小離散単位が記述される。これは中間離散単位の振る舞いを詳細にしたものであり 1 行ずつ記述される (C)。最小離散単位の枠線は省略されているが, これは煩雑さを避け読み易くするためである。これら 3 つの離散単位の間には固定的な相互関連と動的な相互関係が存在する (D)。これらを結ぶことで対象世界の構造が表現されている。

*4 なお離散・構造化モデルから意図的に外れ, より複雑で高度なオブジェクト指向モデルを使った技術の適用も可能ではあるが, 想定ユーザは望まないので本論文では扱わない。

*5 Object oriented Program Descriptive Japanese

*6 OOJ の記述については 2.4 節で説明する。

*7 変化の評価基準を「表現は異なっても同等/同値の記述や構造であるか」とし, 変化を離散単位の特性として捉えるためこのような名称とした。

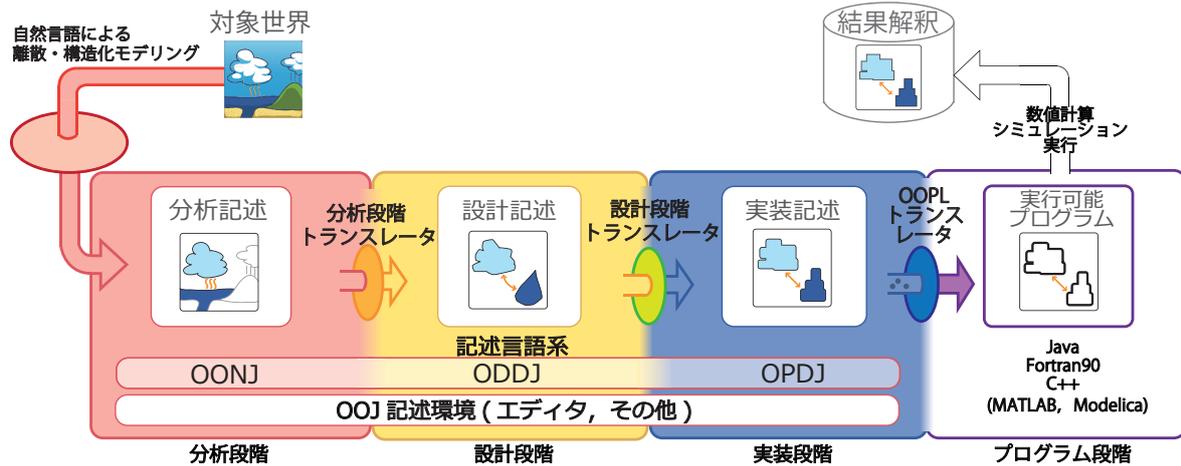


図 2 OOJ を基盤としたプログラム開発過程の全体概念図

Fig. 2 A whole concept diagram of program development processes based on OOJ

表 2 OOJ の各記述段階言語の役割分担

Table 2 Role sharings of each sub language in OOJ

(A) 分析段階記述言語 OONJ	対象世界に関する情報で、プログラム変換に至るまでに必要となる情報 (離散単位や構造についての情報) を全て記述すること。最小離散単位の日本語文は単文であることが推奨される。そしてその離散単位の構造化を要求される。数式は各記述者の考えや通常使う記号と方式で記述して良い。この段階では計算機世界に関わる情報は記述しない。
(B) 設計段階記述言語 ODDJ	計算機特有な要素の追加 (データ型やアクセス属性の追加) し、属性を変数と呼び変えて、離散変数に配列やリストを導入する。ユーザは日本語文を数学関数やライブラリ表現に変換する。設計段階では特定の PL 表現は使わない。
(C) 実装段階記述言語 OPDJ	特定オブジェクト指向プログラミング言語向けのプログラムに自動変換するための準備記述を作る。現時点では Java, C++, Fortran90 に変換するためにこれら 3 つのオブジェクト指向プログラミング言語共通の言語仕様を満たす仕様を導入している。
(D) 記述言語系 OOJ	上記 3 つを統合したのではなく、一貫相似性という狙いを実現するためにまず先に OOJ を設計した。その目的に合わせて記述言語に機能と役割をユーザに分かりやすく区分して分割設計した。(必要上個別の記述言語を先に発表)

2.5 記述環境 OOJ エディタ

前節までに OOJ を利用したプログラム開発は、離散・構造化モデルと一貫相似性の特徴を持つ 3 段階の記述を経て順次行われることを示した。そこで、OOJ によるプログラム開発に特化した開発環境として我々は OOJ エディタを開発した。OOJ エディタは図 3 の (G) にあるように 3 段階の各記述を一括して作成できる機能と各段階を繋ぐための変換機能の 2 種類から成る。

1 つ目の作成機能の説明は図 3 を用いて説明する。図 3 は分析記述の作成画面であり、その内容を離散・構造化モデルで記述している場面である。基本的な画面構成は、記述単位である離散単位をセルとしたテーブル表現である。これは離散化と構造化のイメージに近く、かつ円滑な操作感を目的としたためである^{*8}。入力方法として自然言語で記述する箇所である離散単位の名称 (A-2,B-2) や単文 (C) にはテキスト形式を採用し、相互関係 (D)、単文間の構造化 (E)、離散単位の識別子 (F) などの規則で表現が定められている箇所には、個々に適した入力形式を採用している。これにより、ユーザは記述言語の規則を理解しなくても離

散・構造化モデルに基づいた分析記述、設計記述、実装記述の作成が可能である。

2 つ目の変換機能は、各段階間の変換規則^{*9}に基づいた自動変換により記述間を繋いでおり、各段階間の変換に対する名目上の作業負担は無い。これにより形式的な一貫性を持つ分析記述、設計記述、実装記述の作成が可能である。

以上の機能を持つ OOJ エディタを利用し、分析記述からプログラムまでの一連の開発過程を行い、一貫性を持つ各記述を検討することでモデリングとプログラム間の関係等の調査が可能となり、OOJ の目的を実現することができる。

3. 実際の授業への適用

3.1 授業計画と課題の構成

2012 年度の本学情報工学科 3 年生向け科目「オブジェクト指向プログラミング」を OOJ を利用して分析から実装までを一通り経験する演習を中心とした授業として設計し、学生に受講させ、OOJ の利用が学習成果に与える影響を検証した。表 3 に授業計画を示す。3 年生はモデリングの経験が少ないことを考慮し、モデリング、すなわち分析段階に関する説明を多くし、設計、実装段階については必

^{*8} 離散・構造化モデルの特性上、直感的にはグラフ表現が考えられる。しかし、テーブル表現とグラフ表現を対象に操作テストを行ったところ、グラフ表現では編集の困難さ等の操作への不満の回答があった。

^{*9} 正確には、OONJ, ODDJ, OPDJ の記述規則間の変換規則

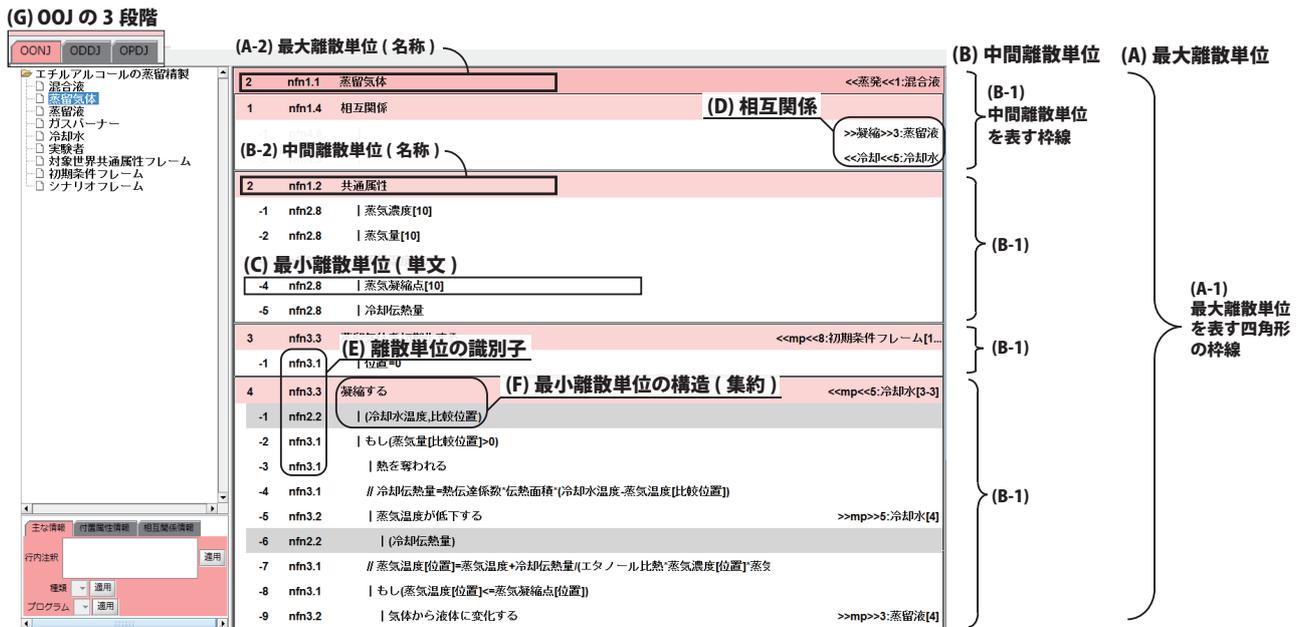


図 3 分析段階の記述例 (アルコールの蒸留)

Fig. 3 Description example of analysis stage “distillation of alcohol”

表 3 講義&演習の授業計画

Table 3 Schedule of lectures and exercises

回数	講義および演習の内容	段階
1 回	講義のガイダンスと全体イメージ形成	基礎概念
2, 3 回	オブジェクト指向と離散構造化	
4, 5 回	分析段階の基礎 (+モデリングの講義)	
6 回	OOJ 記述エディタ演習と OONJ 講義	分析段階
7, 8 回	分析段階の講義	
9, 10 回	分析段階の記述演習	設計段階
11 回	設計段階の講義	
12 回	設計段階の記述演習	
13 回	実装段階の講義	実装段階
14 回	実装段階の記述演習	
15 回	統合された言語系 OOH の講義と演習	概念総合

要最小限の説明となる構成とした。

受講者には、初回の授業で対象世界（高校物理または化学の事象）を 1 つ決めさせ、授業の最後までに OOH を用いてモデリングからプログラムの生成、実行までを行う課題を課した *10。この課題は表 4 に示すように、分析記述に重点を置いた 4 段階に分割されている。まず課題 1 では対象世界を日本語の文章で説明し、課題 2 で離散構造化する。そして課題 3 で OONJ の記述規則に従う記述を行う。最後に課題 4 で ODDJ, OPDJ へ変換、記述を修正した上で、プログラムの生成、実行を行う。

なお、授業時の説明を補うため、多くの記述例を用意して Web で閲覧できるようにした。また、課題 3 で提出されたレポートに対し、分析記述をよくするための助言を個別に行った。

*10 途中で対象世界を変えてもよいが、それ以前の段階の記述をやり直すことになる

表 4 レポート課題の一覧

Table 4 List of report assignments

課題番号	課題	提示した授業
1	対象世界の日本語での分析記述	第 1 回
2	1 回目レポートの離散構造化分析記述	第 5 回
3	OONJ 記述	第 9 回
4	ODDJ 記述以降、プログラム、実行結果、考察、アンケート	第 14 回

表 5 学生レポートに対する評価項目

Table 5 Evaluation items for student's report

(1)	狙い: シミュレーションの狙いが明確であるかどうか
(2)	分析: 対象世界の分析が狙いどおりに行われているかどうか (2-1) オブジェクト抽出 (2-2) 属性抽出 (2-3) 振る舞いの抽出 (2-4) 相互関係の抽出
(3)	設計・実装への変換
(4)	実行結果の妥当性

3.2 レポートおよびアンケートによる学習効果の検証

出席回数を充足する履修者は 23 名で、そのうち 18 名が最終レポートを提出した（提出率 78%）。このレポートを筆者らで評価した。評価項目は表 5 に示す 4 項目で、(2) の分析についてはさらに 4 項目に細分化して評価した結果の平均を取った。なお、評価点は表 6 に示す 5 段階である。すべてのレポートの評価結果を、4 つの項目の平均値の降順に並べたものを表 7 に示す。全受講者の平均値は 3.72 であった。

また受講者には最終レポート提出の際にアンケートへの回答を添付するよう依頼した。回収件数は 16 件（回収率 89%）であった。アンケートの設問は 12 問である。その内

表 7 全ての受講者のレポートを評価した結果

Table 7 Evaluation result for all student's reports

受講者	対象世界	狙い	分析					設計・実装への変換	実行結果の妥当性	平均	成績グループ	レポート作成にかかった時間					変更	
			オブジェクト抽出	属性抽出	振る舞いの抽出	相互関係の抽出	分析平均					1	2	3	4	合計	対象世界	OONJ記述
1	列車の運行	5	5	5	5	5	5	5	5	5	A							
2	アルコールの蒸留	5	5	5	5	5	5	5	5	5	A	2	12	20	6	40		
3	斜面上の物体の運動	5	5	5	5	5	5	5	5	5	A	1	2	4	6	13	なし	あり
4	斜法投射と跳ね返り	5	5	5	5	5	5	5	5	5	A	10	10	50	10	80		
11	水の電気分解	4	5	5	5	5	5	4	4.5	4.5	A	2	8	50	20	80	なし	あり
14	ビリヤード	3	5	5	5	5	5	4	4.25	4.25	A	10	6	20	60	96	あり	あり
9	ミニ四駆	5	5	4	5	4	4.5	3	3.88	3.88	B						なし	若干
8	メタンの燃焼	4	3	4	5	4	4	3	3.75	3.75	B	3	1	4	15	23		
13	フリーキック練習	3	5	4	5	5	4.75	4	3.69	3.69	B	1	3	3	5	12		
15	バドミントン	3	5	4	5	5	4.75	4	3.69	3.69	B	2	4	4	10	20	あり	あり
6	フリースロー練習	3	5	4	4	5	4.5	4	3.63	3.63	B	2	3	5	10	20		
7	二次電池	4	5	5	5	5	5	3	2	3.5	B							
10	食物連鎖	4	5	4	4	3	4	3	2	3.25	B	1	2	6	4	13	なし	若干
16	心臓の動き	2	5	5	5	5	5	3	3	3.25	B						あり	若干
5	三体衝突	5	5	5	5	3	4.5	2	1	3.13	B	2	2	6	8	18		
12	蛍光灯の発光	3	4	3	4	2	3.25	3	2	2.81	C	1	1	5	4	11		
17	組織の世界	2	3	2	2	3	2.5	2	2	2.13	C							
18	放物線運動	2	2	1	3	1	1.75	1	1	1.44	C	2	2	2	3	9	あり	若干
	平均	3.72	4.56	4.17	4.56	4.17	4.36	3.67	3.11	3.72		3	4.31	13.77	12.38	33.46		

表 6 各評価点に対する評価基準

Table 6 Evaluation criteria for each evaluation value

評価点	基準
5	妥当
4	わずかな記述の修正が必要
3	構造的な修正が必要
2	根本的な見直しが必要
1	記述法、教員の指示に従った記述になっていない

容と選択肢による回答を表 8 に示す。自由記述による回答は量が多く、すべてを記載することができなかつたため、考察に必要な回答を抜粋して本文中に示すこととする。なお、設問 9 におけるレポートの記述にかかった時間は受講者の主観による値であるが、客観的な値は計測困難ため、この値を用いて考察を行なった。表 8 には平均、最小、最大のみを記載し、個々の受講者の値は前述の表 7 へ記載した。

4. 考察

4.1 受講者の事前知識

まず、受講者の事前知識をアンケートの設問 12 から確認した。分析/モデリングを普段からプログラム開発で行っている受講者は 0 であり、その一方で経験がないと答えた受講者が 8 名であることから、受講者達にはこれまでに分析/モデリングの経験がほとんどなかつたことが確認さ

れた。

4.2 OOH 利用に対する受講者の印象

アンケートの設問 6、設問 7 の自由記述に「日本語を用いて理解するためクラスなどの文法 (Java でいう) のような他のことに気を使わなくていいぶん理解しやすかつた」、 「読み下す場合には日本語で書かれている分、プログラムの流れや目的は簡単に抑えることができる」といった回答があつた。このことから、記述言語として一貫して自然言語である日本語をそのまま使えるようにしたことが、受講者の理解の一助となつたことが確認できる。

一方、アンケートの設問 2 では、この授業の悪かつた点として、「エディタが使いにくかつた」と回答した受講者が 7 名いた。とくに設問 4 で「プログラムへの変換に関わる作業」を苦勞した点に挙げた受講者が 9 名おり、同自由記述欄にも「エディタでいちいち一文ずつ修正 (プログラムを OOH にする時の OOH の修正) を行うのが大変だつた。」との記述があつた。従来のプログラミングはテキスト形式が大半であるが、OOH エディタでは離散・構造化モデルを模したセル形式を採用したために、従来と異なる操作となつたことが原因だと考えられる。

4.3 レポートの評価結果から考察する学習効果

学習成果の客観的な評価は表7から検証する。この表の評価の分布からは、受講者は3つのグループに分類することができる。上位6名は、プログラムの実行まで完成（あるいはほぼ完成）している（Aグループ）。このグループの評価値の平均は4を越える。

7番目から15番目までの9名は、プログラムの実行までたどり着くことがなかったが、途中段階までの記述（とくに分析段階）の記述は概ね妥当である。分析、設計段階での記述のわずかな不備が後段に影響して実行までたどりつかなかったものが多く、これらの不備を解消すればプログラムの実行まで行えると考えられる（Bグループ）。このグループも平均で3点台と良い評価となっている。

一方、下位の3名はいずれの項目でも評価が低く、学習効果はなかったと考えられる（Cグループ）。評価の平均は3未満である。

AグループとBグループの受講者にはOOJによる学習効果があったと考えられる。この2グループの人数を合わせると15名になり、最終レポート提出者の83%となる。

レポートの作成時間に注目すると、Bグループの合計時間が20時間程度であるのに対し、Aグループでは40時間を越えていることが多い。レポートは作成時間の長さに応じて評価が高くなる傾向があると言える。

また、全体にみられる傾向として、設計・実装段階に進んだ後に、以前の段階、とくに分析段階の記述を修正していることが挙げられる。このことは同じ学習者が提出した複数のレポートの間で、同じ段階の記述を含むものを比較することで確認された。修正された箇所を筆者らで検証したところ、後の段階への変換後に問題となる記述が修正されている傾向があった。設計・実装段階へ進んだ後に問題点に気づき、分析段階に戻ってその原因を見つけ、修正したものと考えられる。その結果、レポート評価の向上にもつながっている。このような過程を経たことによって、受講者の分析段階に対する理解は向上しているものと考えられる。

一方、途中で行き詰まった場合には、対象世界を変えた方がより後段の作業を意識した対象世界の選択と分析ができるのではないかと考え、そのように指示したが、高評価となったものもあれば、低評価に止まったものもあり、必ずしも最終評価の向上にはつながらなかったようである。

問題点として、Bグループの受講者にみられるように、分析段階の評価が高くても、軽微な記述の誤りにより、プログラムの実行に至らない場合が多いことが挙げられる。

4.4 受講者が得られたと感じたもの

設問3では、回答1,2を選んだ受講者が多く、オブジェクト指向の概念理解、およびオブジェクトの抽出に対する理解・経験が得られたと感じる受講者が多かったことが確

認できた。自由記述欄においても、「やり直しが一番多かったのはOONJ記述。記述しているうちにOONJというものの理解が深まり、それによって間違いを見つけて修正という流れが非常に多かったためである。」とある。この感想から受講者がOONJ, ODDJ, OPDJの段階を経ることによって記述の間違いを発見できたと理解していると考えられる。

また、設問1の自由記述欄にも「自分で一つ世界をモデリングし、オブジェクトと相互関係というOOの世界におとし込む作業は物事をOO的視点で再確認でき、視野が広がった。」とあった。実際にモデリングを自分で行い、正しくモデリングできるまで、何度も作業を繰り返すことにより、正しいオブジェクト指向の概念の理解を深めることになったと考えられる。

一方で、アンケートの設問2に対し、「概念が理解できなかった」と回答した受講者が5名おり、設問4の自由記述欄にも「概念が理解できず、自分のやりたいことがあってもどのようにすればいいかわからずに、全体的なすすみ具合が悪かった」との記述があった。概念の理解が難しいことは、オブジェクト指向そのものが持つ問題点であり、分析からプログラム実行まで一貫した開発における成功体験を積ませることで、その理解を促すことが本研究の目的であるが、成功体験を得られない受講者もいたことになる。最初ももっと簡単な対象世界から始め、次に本格的な対象世界を扱うような繰り返しを行えば、より理解しやすくなるのではないかと筆者らは考える。

4.5 考察のまとめ

今回の対象となった受講者はこの授業を受講するまでは、モデリングから始まる本格的なプログラム開発を行った経験はなかった。しかし、この授業の結果として、全体の8割となるA, Bグループの受講者は妥当な分析結果を導き出している。またそのうち半数はレポートの作成過程において設計、実装段階から分析段階に戻って修正を行っていた。

このことから、まず受講者はOOJエディタを用い、OOJの記法に従って分析、設計、実装を行うことによって自然に妥当な記述へと導かれていることが考えられる。その際に、自然言語（日本語）による記述が抵抗感の軽減につながっていることが、アンケート回答から伺える。

また、設計、実装段階へ移行した後に、分析記述を修正している点からは、分析結果に対応する設計、実装の結果を知ることにより、分析結果が設計、実装に与える影響を実感し、設計・実装段階の不具合を解消するために、分析段階に戻って設計、実装が妥当となる修正を加えていると考えられる。分析段階はモデリング作業の一部であるが、モデル（オブジェクト）の外形をほぼ決定する重要な役割をもつ。設計、実装に合わせた妥当な分析を行った経験は、

モデリング技能全体の向上に大きく寄与していると考えられる。なお、実際のソフトウェア開発において、仕様や設計の不備をコーディングの段階でのみ解消してしまい、文書（仕様書や設計書）とソフトウェアの動作が一致しないことが問題視されることがある。今回の試みにおいて、受講者は分析記述に起因する問題点の修正は、必ず分析段階へ遡って修正を施しており、相似性を損うような後段のみの修正を行っていないことが確認されている。

以上により、OOJを通してプログラム開発を行ったことにより、一貫相似性を保ったプログラム開発を理解し、プログラム開発技能の向上に寄与していると推察する。

5. 関連研究

本論文では、個人向けの OO モデリングから OO プログラムを得るまでの開発教育手法を提案した。

一方で OO のモデリング教育やソフトウェア開発教育には、数名から成るチームによる演習の実施が効果的であるとされており、大学における教育でも協調学習として取り入れられている [4]。このようなチームで行う教育手法に PBL(Project Based Learning) が存在する。PBL とは、チームで 1 つの課題に取り組み、その中で主体的に知識や技術、景観を獲得していく教育手法のことである [3]。ただし、PBL では技術力以外にもコミュニケーションスキルやプロジェクトマネジメント技術なども必要であり、ある技術の初学者が利用するのは難しく、個々の技術の習得は学習者自身行う必要があるなど効果的に実施するには多くの課題がある。また、PBL でも使用される UML や MDD によるモデリング教育も行われている。モデリング教育には、CASE ツールや解答のある課題が与えられ UML の表記方法や MDD による開発方法の取得を目標とする教育 [1] と UML や MDD を利用してモデリング技術やソフトウェア開発方法の取得を目標とする教育 [2] がある。ただし、これらの技術を適切に理解し、活用できるようになるには、OO の概念の理解が必須であるため必然的に大学 3 年以降から大学院生を対象となり、ある技術の初学者が利用するのは難しい。

以上のことから、OO の技術や実装スキルがある学習者がソフトウェア開発の実践教育を行う点では PBL を利用した教育やモデリング教育が有利である一方で、OO モデリングとプログラム間の関係を初学者でも簡易に学べる点では OOJ の方が優位である。このように PBL を利用した教育、モデリング教育と OOJ を利用した教育では教育目的が異なる。また、これらの特徴から OOJ を利用した教育手法の後にモデリング教育、さらには PBL を利用した教育へは円滑に移行できると考えられ、相補的な関係にある。

6. まとめ

筆者らは離散構造化モデルに基づくプログラム開発環境 OOJ がオブジェクト指向プログラミングの技能向上に利用できると考えた。そこで、情報工学科 3 年生向けの科目「オブジェクト指向プログラミング」で OOJ を利用した授業、演習を行い、その教育効果を検証した。対象となった受講者のほとんどはモデリングの経験がなかったが、最終課題のレポートでは、ほとんどの受講者が妥当な OOJ 記述を行っていた。すなわち、相互関係をもつオブジェクトの集合として妥当なプログラムを設計することができた。妥当な設計に到達した受講者の半数近くは ODDJ の設計段階以降になって OONJ の分析記述を修正していることから、OOJ における一貫相似性の存在が、分析段階の間違いが後段の記述に与える影響を理解することに繋がったものと考えられる。アンケートにおいても、オブジェクト指向の概念が理解できたと回答した受講者が多かった。

今後の課題としてはエディタの使い勝手の向上などが挙げられる。

参考文献

- [1] 野沢光太郎, 松澤芳昭, 酒井三四郎: 一貫性・明瞭性診断による静的 UML モデリング学習支援システムの設計と評価, 情報処理学会論文誌, Vol.55, No.5, pp.1471-1484 (May 2014).
- [2] 赤山聖子, 久住憲嗣, 部谷修平, 福田晃: オブジェクト指向モデリング教育におけるモデル駆動開発ツールの活用方法の検討, 情報処理学会論文誌, Vol.55, No.1, pp.72-84 (Jan. 2014).
- [3] 岩本智裕: ペタ語義: PBL Summit —与えられる教育からの脱却—, 情報処理, Vol.55, No.1, pp.90-93 (Jan. 2014).
- [4] 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 (Feb. 2012).
- [5] Object Management Group: UNIFIED MODELING LANGUAGE, ObjectManagement Group(online), 入手先 (<http://www.uml.org/>) (accessed 2014-04-03).
- [6] 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, 情報処理学会研究報告, Vol.2011-MPS-86, No.30 (Oct. 2011).
- [7] 畠山正行, 池田陽祐, 三塚恵嗣, 加藤木和夫, 大木幹生, 上田賀一: 離散・構造化モデル記述言語系 OOJ の構築と科学技術計算教育への適用-分析からプログラムまでの一貫開発と V&V 評価実現の検討-, 情報処理学会論文誌: 数理モデル化と応用, Vol.6 No.3, pp.23-42 (Dec. 2013).
- [8] 池田陽祐, 三塚恵嗣, 加藤木和夫, 大木幹生, 上田賀一, 畠山正行: 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.5, No.3, pp.63-78 (Sep. 2012).
- [9] 池田陽祐, 三塚恵嗣, 上田賀一, 畠山正行: UML との比較評価に基づくオブジェクト指向分析設計記述言語 OONJ の記述技法の特徴, 情報処理学会論文誌: 数理モデル化と応用, Vol.6 No.1, pp.156-171 (Mar. 2013).
- [10] 三塚恵嗣: オブジェクト指向一貫記述言語系 OOJ における変換機構と実装記述言語 OPDJ の開発, 平成 22 年度

表 8 アンケート結果
Table 8 Questionnaire results

設問 1. この授業の良かった点を挙げてください (複数選択可)			
1. 概念が理解しやすかった	4 名	4. エディタが使いやすかった	0 名
2. 資料が読みやすかった	2 名	5. 演習時間が充分確保されていた	12 名
3. 演習課題の内容が適切であった	3 名	6. その他	1 名
設問 2. この授業の悪かった点を挙げてください (複数選択可)			
1. 概念が理解できなかった	5 名	4. エディタが使いにくかった	7 名
2. 資料が読みにくかった	6 名	5. 演習時間が不足していた	0 名
3. 演習課題の内容が不適切であった	0 名	6. その他	1 名
設問 3. この授業で得られたと思うことを挙げてください (複数選択可)			
1. オブジェクト指向の概念に対する理解	14 名	5. オブジェクト間の相互関係抽出に対する理解・経験	5 名
2. オブジェクト抽出に対する理解・経験	11 名	6. 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 名
3. オブジェクトの属性抽出に対する理解・経験	3 名	7. その他印象に残った点などを自由記述欄に書いてください	0 名
4. オブジェクトの振る舞い抽出に対する理解・経験	5 名		
設問 4. この授業において苦労したことを挙げてください (複数選択可)			
1. オブジェクト指向の概念理解	4 名	7. オブジェクトの属性抽出作業	4 名
2. 対象世界の調査・理解	7 名	8. オブジェクト間の相互関係抽出作業	7 名
3. OOJ 記述法の理解	8 名	9. プログラムへの変換に関わる作業	9 名
4. OOJ エディタの操作	10 名	10. レポートの作成	6 名
5. オブジェクトの抽出作業	2 名	11. その他	1 名
6. オブジェクトの振る舞い抽出作業	4 名		
設問 5. この授業を通じて、オブジェクト指向による分析・設計の有用性や問題点を理解できましたか？			
1. 完璧に理解できた	0 名	4. 一部しか理解できなかった	3 名
2. 大体は理解できた	4 名	5. 全く理解できなかった	1 名
3. 自身で利用できる範囲では理解できた	7 名		
設問 6. OOJ を用いた分析/モデリングは使いやすかったですか？			
1. 大変使いやすかった	1 名	4. 使い勝手が悪いが、なんとか使えた	3 名
2. 幾つか抵抗感を感じたが、使い勝手に問題はなかった	5 名	5. 使いにくかった	3 名
3. 使いやすくないが、問題なく使えた	2 名		
設問 7. OONJ の記述法は Java の文法に比べて理解が容易でしたか？			
1. OONJ の方が理解は容易だった	4 名	3. Java の方が理解は容易だった	6 名
2. 同じくらいだった	2 名	4. 両者は様々な部分で違うので比較できなかった	1 名
設問 8. OOJ エディタを操作できるようになりましたか？ (複数選択可)			
1. 十分に操作できるようになった	1 名	3. 最低限はできるようになった	8 名
2. 目的を達成できる程度にはできるようになった	5 名	4. 使い方が分からず、全く慣れなかった	1 名
設問 9. 以下の記述の作成にかかった時間を教えてください。			
(第 1 回レポート) NJ 記述に掛かった時間	平均:3.25 時間	最小:1 時間	最大:10 時間
(第 2 回レポート) オブジェクト指向記述に掛かった時間	平均:4.1 時間	最小:1 時間	最大:12 時間
(第 3 回レポート) OONJ 記述に掛かった時間	平均:10.7 時間	最小:2 時間	最大:50 時間
(最終レポート) 最終レポート作成に掛かった時間	平均:13.6 時間	最小:3 時間	最大:60 時間
設問 10. プログラミング演習 III や IV を受けた方のみ回答をお願いします。この授業とプログラミング演習 III や IV と比べて印象として最も違う点を教えてください。			
設問 11. 今までに書いた最長のプログラムの大体の行数を教えてください。また、そこで使用したプログラミング言語を教えてください。			
設問 12. この授業以前に分析/モデリングを行った経験はありますか？			
1. 経験があり、プログラム開発時にはよく利用している			0 名
3. 授業や自身で勉強した経験はあるが、実際に使用した経験はない			4 名
2. 経験はあるが、あまり利用していない			2 名
4. 行った経験はない			8 名

- 茨城大学大学院修士論文 (Feb. 2011).
- [11] 数值流体力学編集委員会編：数值流体力学シリーズ 2, 圧縮性流体解析, 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, 森北出版株式会社 (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)SRC (1993).
 - [14] メイヤー B. 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)翔泳社 (Jan. 2007).