

## オブジェクト指向モデリング学習のための インスタンス図自動生成システム

早川 勝<sup>†1</sup> 野沢 光太郎<sup>†1</sup>  
松澤 芳昭<sup>†2</sup> 酒井 三四郎<sup>†2</sup>

クラス図に対応したインスタンス図を自動生成し初学者に提示するシステムを開発した。インスタンス図の基本生成アルゴリズムは、あるクラスのインスタンスを1つ生成し、リンクする相手のクラスのインスタンス数について、多重度を考慮してランダムに決定する。生成指示のたびに異なるインスタンス図が生成されることが特徴である。大学生の初学者を対象に、本システムの試用実験をおこなった。その結果、本システムによって、1) 初学者がモデルが表現している状況を網羅的に把握できる、2) 抽象クラスの継承が含まれているモデルの理解ができる、ことが分かり、これを通して初学者のモデリング学習を支援できることが確認できた。

### Instance Diagram Automatic Generator for Beginners of Object-Oriented Modeling

SHO HAYAKAWA,<sup>†1</sup> KOTARO NOZAWA,<sup>†1</sup>  
YOSHIAKI MATSUZAWA<sup>†2</sup> and SANSHIRO SAKAI<sup>†2</sup>

In this paper, we report the system named "Instance Diagram Auto Generation System". The system have been designed for beginners of object-oriented modeling with UML(Unified Modeling Language). The proposed system has a function generating instance diagrams from a class model. Learners can use the generated diagrams to keep a consistency between class diagram model and its instance representation, such as multiplicities and real number of instance, or associations and links. We examined the system for beginners. As a result, we have found that instance diagrams generated by the system succeeded to support beginners to understand a UML model.

### 1. はじめに

各種のオブジェクト指向方法論が80年代後半に提案され、その後、ソフトウェア、および情報システムの開発現場に本格的に普及した。その後、さまざまなオブジェクト指向の開発方法論が登場したが、UML (Unified Modeling Language) が90年代に登場して以後、UMLを用いてシステムをモデリングすることが、オブジェクト指向開発方法論の標準になった<sup>1)</sup>。現在、大学の情報系学科では、モデリングの知識と技能の習得を目指す授業が開講されている。

UMLの図の中の1つに、クラス図がある。クラス図は、モデリング対象の静的構造を描いたものである。クラス図において扱う「クラス」は、抽象的な概念であり、UML学習を始めたばかりの初学者にとって、クラス図を読み書きすることは難しい。モデリングの成果物には一意な正解がないため、初学者が自習を行うことが難しいことも、モデリング技能の習得しづらさを大きくしている。

クラス図のクラス同士の関連や多重度を読み解く上で、有効なのがインスタンス図である。インスタンス図はクラス図をより具体的に表現した図であり、モデリング上級者であっても、インスタンス図が書けることがクラス図を理解したことになる。

本研究は、クラス図に対応したインスタンス図のサンプルを自動で生成し、提示することで、初学者のクラス図が表現しているモデルの読解を支援することを考えた。本稿では、我々が開発したシステムについての概要およびインスタンス図を生成するアルゴリズム、およびその効果を検証した評価実験について述べる。

第2章で本研究に関連する先行研究について述べる。第3章では、我々が提案するシステムの概要やインターフェース、我々が想定しているユースケースについて説明する。第4章では、本システムがどのようにインスタンス図（以下、インスタンス図サンプルとする。）を生成するのか、そのアルゴリズムについて述べる。第5章では、UML初学者を対象に実施した本システムの評価実験の方法について説明し、第6章では、第5章の結果について述べる。第7章にて、第6章の考察を行い、第8章で総括を行う。

<sup>†1</sup> 静岡大学大学院情報学研究科  
Graduate School of Informatics, Shizuoka University

<sup>†2</sup> 静岡大学情報学部  
Faculty of Informatics, Shizuoka University

## 2. 関連研究

初学者に対してUMLを用いたモデリングの支援を行う研究のレビューを行う。Haskerらは、「多重度の付加における誤り」や「複数のクラスとコンポジションの関連を持つクラスが存在する」、「継承において、継承元と継承先の関係が逆になっている」などといった、クラス図によく見られる欠陥を定義し、それらの欠陥がクラス図内に含まれているかどうかを解析し、ユーザに提示を行うシステムであるUMLintを開発している<sup>2)</sup>。次にHaskerらは、UMLintをベースに、改良をほどこしたシステムであるUMLGRADERを開発している<sup>3)</sup>。UMLintが、クラス図に含まれる、機械的に見つけることができる欠陥を発見し、かつ学生が欠陥の修正を行う際に役立つ助言を行うことを目的としているのに対して、UMLGRADERは、学生が作成したクラス図によく見られるクラス図の欠陥について、対応する欠陥のパターンを限定することで、より具体的な助言をすることを目的としている。

上原らはクラス図の作成には、ア. クラス図の作成 イ. クラス図の確認の2つのフェーズがあり、クラス図を作成する者はこのア、イの作業を繰り返し行うことで、クラス図を洗練していくと主張している<sup>4)</sup>。上原らは、イ. における支援方法として、インスタンス図の利用の有効性を主張しており、2つのクラス間の関係に着目し、その関係を示すインスタンス図を生成する機能を提案している<sup>5)</sup>。

野沢らは、まず初学者にクラス図を作成させ、その後システムが提示する具体的なシナリオに沿ったインスタンス図を初学者に描かせることで、矛盾を発見させるという研究を行っている<sup>6)</sup>。杉浦らは、凝集度と結合度に注目して、初学者が設計の良さを評価できる指標とその計算支援ツールを提案している<sup>7)</sup>。

## 3. システムの提案

### 3.1 システムの目的

本研究では、UMLによるモデリングの学習を始めた初学者を対象とする。本研究の目的は、インスタンス図サンプルの自動生成によって、作成したクラス図をインスタンスレベルで確認する材料を提供し、クラス図内のクラス間の関連や多重度について矛盾していないかどうかを検証する確認作業を支援することである。

### 3.2 ユースケース

本システムの用途として、以下の3つを想定している。「ユーザ」は初学者を指し、「UC」はユースケース (UseCase) を表す。

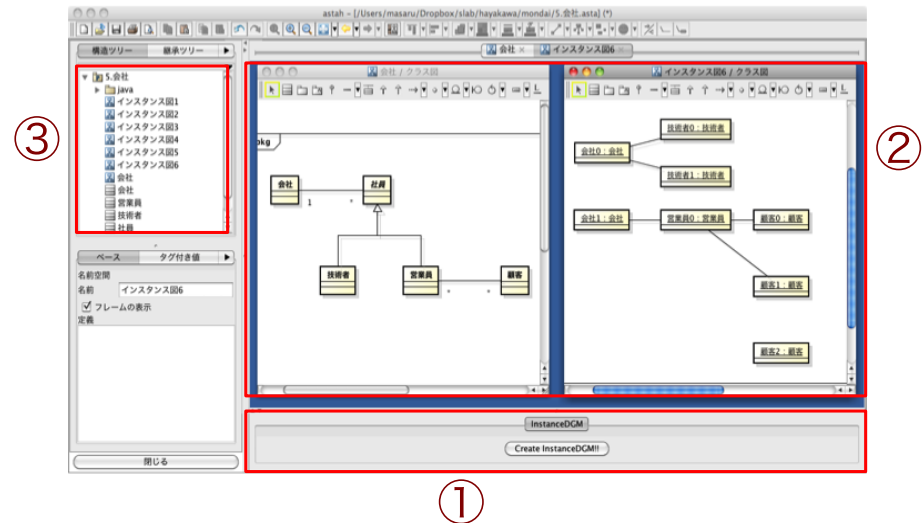


図1 本システムのスクリーンショット

**UC1** ユーザが、自分の認識と矛盾する点が含まれるクラス図に対応したインスタンス図サンプルを作成し、クラス図と照合、吟味することで、クラス図の修正すべき箇所を発見する。

**UC2** ユーザが、正しいクラス図からインスタンス図サンプルを作成し、クラス図と照合、吟味することで、クラス図が正しいことを確認する。

**UC3** 色々なパターンのインスタンス図サンプルを生成し、閲覧する。

UC1およびUC2は、クラス図に対応したインスタンス図サンプルをユーザに見せることで、クラス図の多重度やクラス同士の関連といったモデルを具体的に掴むことができるという仮説に基づくものである。それらクラス図内の要素が何を示しているかをインスタンス図によって正確に認識することを支援することを目指す。

UC3は、ユーザがインスタンス図サンプルを生成する度に、違うパターンのインスタンス図サンプルを提示することで、ユーザの想定していなかったものが閲覧できるという仮説に基づく機能である。

### 3.3 システムの概要

本システムは、ChangeVision社のソフトウェア設計支援ツール、astah\*のプラグインと

して機能する。

図1に、本システムのスクリーンショットを示す。図内の、数字を記入した場所の名称と機能については、以下のようにになっている。

- (1) 拡張タブビュー. この拡張タブビューの中央にあるボタンがインスタンス図サンプルの自動生成を実行するボタンである。(以下、インスタンス図サンプル生成ボタンとする。)
- (2) 図ビュー. クラス図や生成したインスタンス図サンプルはここに表示される。
- (3) プロジェクトビュー. 同一プロジェクト内に保存されている図やモデルがツリー状に表示される。

本システムでは、astah\*のプラグインの機能として提供されている拡張タブビューを追加し、そこにインスタンス図自動生成機能を実行するボタンを付加した。astah\*上にてクラス図を選択している状態でこのインスタンス図サンプル生成ボタンを押すと、選択しているクラス図に対応したインスタンス図サンプルを生成し、図ビューに表示する。その際、生成元のクラス図を左側、生成したインスタンス図を右側に並べて表示する。これはクラス図とインスタンス図サンプルを同時に参照できるようにするためである。

プロジェクトビューには、生成するたびに、そのインスタンス図サンプルが追加されていく。過去に生成したインスタンス図サンプルを再度見たい場合には、プロジェクトビューから選択し、図ビューに表示することができる。

#### 4. インスタンス図サンプル生成アルゴリズム

インスタンス図サンプル生成アルゴリズムについて、「基本アルゴリズム」「3クラス以上のモデルから、インスタンス図を生成する場合」「抽象クラスが含まれるモデルからインスタンス図を生成する場合」の3つに分けて説明する。

##### 4.1 基本アルゴリズム

本節では、インスタンス図サンプルを生成する際の基本となるアルゴリズムについて述べる。図2は、左上がクラス図である。出力インスタンス図A、B、Cはそのクラス図が表現しているモデルから生成されるインスタンス図サンプルの例である。以下が、基本アルゴリズムである。

- (1) クラス図にある関連を取得する。
- (2) 取得してきた関連から、一方の関連端のクラスを取得する。
- (3) 関連端のクラスが抽象クラスであるかどうかを確かめる。

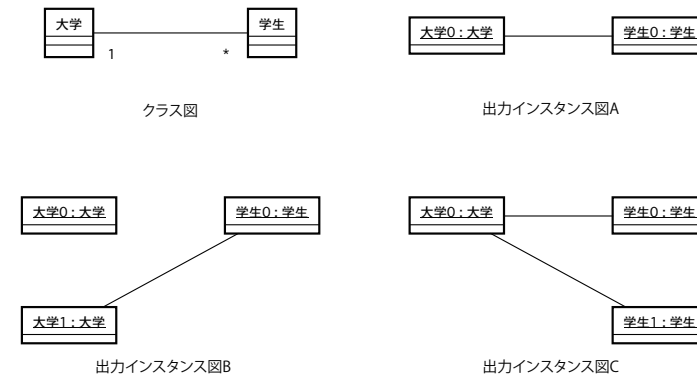


図2 生成されるインスタンス図サンプルの例1

- (4) (3)で抽象クラスでなかった場合、取得してきた関連端のクラスのインスタンスを1つ、生成する。
- (5) 多重度を確認する。
- (6) 他方の関連端のクラスが、抽象クラスであるかどうかを確かめる。
- (7) (7)で抽象クラスでなかった場合、多重度に応じた個数の、他方の関連端のクラスのインスタンスとリンクする。
- (8) (2)から(8)の手順を、(2)で選択しなかった方の関連端のクラスについて、再度行う。

上記の手順(5)にある「多重度に応じた」について説明する。本アルゴリズムでは、多重度が「\*」または「0..\*」であれば、0個、1個、2個のいずれかの個数のインスタンスとリンクする。多重度が「1..\*」であれば1個もしくは2個のインスタンスとリンクする。どの個数になるかは、ランダムに決定する。また、多重度が「1」のときには必ず相手方のインスタンスを生成し、リンクする。ランダムに決定しているため、インスタンス図生成ボタンを押すたびに異なるインスタンス図サンプルを提示することができる。

なお、他方の関連端のクラスのインスタンスが既に存在する場合は、既存のインスタンスとリンクをするか、新しくインスタンスを生成しリンクするかをランダムに決定する。存在しない場合は、多重度に応じて決定した個数のインスタンスを生成し、リンクする。

##### 4.2 3クラス以上のモデルからインスタンス図を生成する場合

本節では、3クラス以上のクラス図から構成されるモデルについてインスタンス図サン

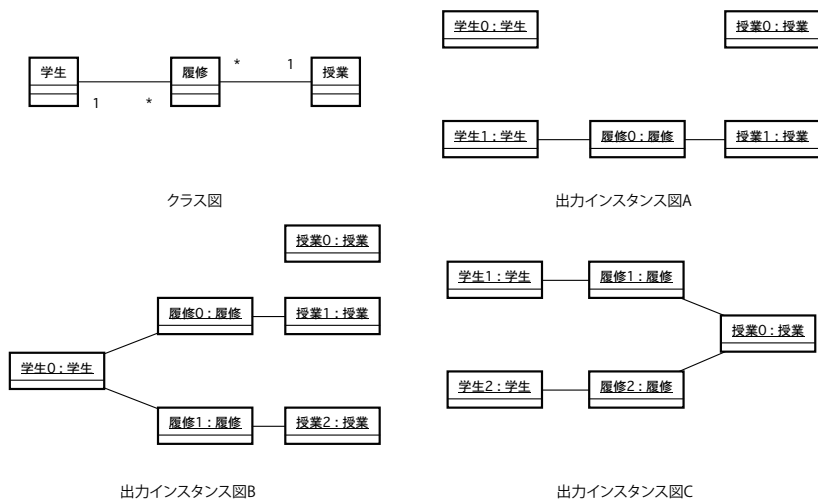


図 3 生成されるインスタンス図サンプルの例 2

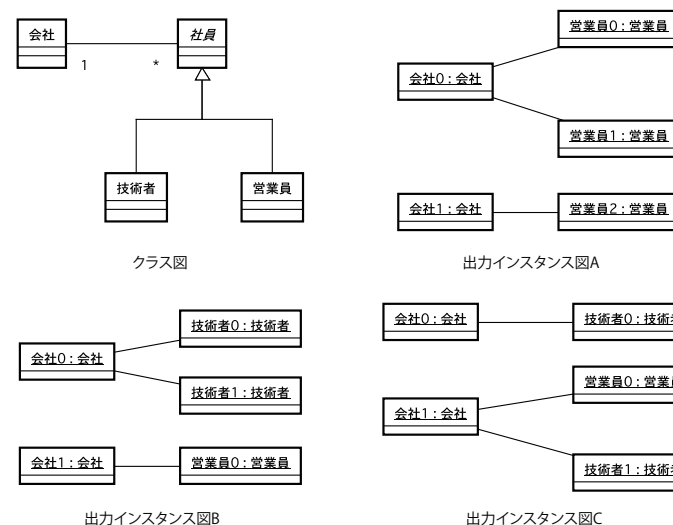


図 4 生成されるインスタンス図サンプルの例 3

ルを生成するときのアルゴリズムについて述べる。この場合でも、4.1 節に示したアルゴリズムにてインスタンス図サンプルの生成を行う。図 3 に 3 クラスで構成されるクラス図の例を示す、左上がクラス図であり、出カインスタンス図 A, B, C はそのクラス図から生成されるインスタンス図サンプルの例である。3 クラス以上のクラス図の場合、図 3 のクラス図を例にとると、クラス図を「学生」と「履修」、「履修」と「授業」のように分けて考える。共有のクラス（図 3 では「履修」、以下、共有クラスと呼ぶ）のインスタンスは、インスタンス図生成の実行が後になる 2 クラスでは既存の共有クラスのインスタンスを利用するか、新たに共有クラスのインスタンスを生成する。

しかし、インスタンス図生成の実行が後になった 2 クラスにおいて、共有クラスのインスタンスが新しく生成されると、先のインスタンス図生成を実行した 2 クラスにおける多重度によっては、インスタンス図にモデルとの不整合が起りうる。その不整合を解消するため、クラス図とインスタンス図の矛盾を検証を行う機能を実装した。本システムでは、インスタンス図サンプルを astah\* の図ビューに表示する前に、インスタンス図がクラス図と矛盾していないかを確認し、不整合が検出された場合は是正を行う。

### 4.3 クラス図中に抽象クラスが存在する場合

図 4 に示す、左上に示したクラス図のような、クラス図内に継承関係をもつクラスが存在

する場合のインスタンス図生成アルゴリズムについて述べる。ただし、4.1 節に示した手順 (3) までは同様であるので、以下では (3) で抽象クラスだったとき以後のアルゴリズムについて述べる。

- (1) 抽象クラスであったクラスを継承しているサブクラスを取得する。複数であった場合は、そのうちの 1 つをランダムに決定する
- (2) (1) で取得したクラスが抽象クラスであるかどうかを確認する。
- (3) 以下、4.1 節に示した手順 (4) 以降と同様。

図 4 の出カインスタンス図 A, B, C は、インスタンスの数やそのリンクのされ方が似ているが、「会社」のインスタンスが「営業員」のインスタンス 2 つとリンクされているケースや「技術者」のインスタンス 2 つとリンクされているケース、「技術者」と「営業員」のインスタンス 1 つずつとそれぞれリンクされているケースなど、クラス図が表現しているモデルを網羅したインスタンス図が生成されている。

表 1 実験の概要

問題番号	出題内容	システム使用	備考
問題 1	継承を含む, 7クラスからなるモデル		事前試験
問題 2	2クラスからなるモデル	○	
問題 3	3クラスからなるモデル	○	
問題 4	4クラスからなるモデル	○	
問題 5	継承を含む, 5クラスからなるモデル	○	
問題 6	継承を含む, 6クラスからなるモデル		事後試験

## 5. 実験方法

### 5.1 概要

本章では、本システムの評価方法について述べる。

本研究のユーザとして対象としているのは、大学の授業で UML を利用したモデリングの講義を受けたことがある初学者である。大学の授業でモデリングを受講した経験のある者を対象に、評価実験を行った。ただし、モデリングの研究を行っているなど、モデリングに関する知識および技術に長けたものは被験者から除外した。

実験は、問題 1 から問題 6 までの 6 問であり、順番に全て実施する。その後、アンケートおよびインタビューを実施した。どの作業も、(1) モデル化対象の記述、(2) モデルを説明した文章（以下、モデル記述とよぶ）、(3) クラス図、の以上 3 つから構成されたモデルを読み、解答を行う。実験群はシステムを利用して解答することができる。統制群はシステムを使用せずに解答を行った。

表 1 は実験に用いた各問の内容とその問題で確認できる本システムのユースケースについてまとめたものである。問題は UML モデリング技能認定試験入門レベルの問題集<sup>8)</sup>を参考に作成した。問題の詳細については付録を参照されたい。表 1 の「出題内容」は、各問題の内容を説明したものである。「システム使用」は、○のついた問題では実験群がシステムを利用できる問題であり、問題 1 と問題 6 は実験群、統制群どちらもシステムを使用せずに解答を行った。

実験は二つの課題に分けられる。1 つは、クラス図中に含まれる、モデル記述と矛盾する点を発見し、修正する課題（以下、クラス図修正課題と呼ぶ。）。もう 1 つは、与えられたモデルの典型的な場面を表現したインスタンス図を紙に書くというものである（以下、この課題をインスタンス図作成課題と呼ぶ。）。

### 5.2 クラス図修正課題

クラス図修正課題では、モデル内のクラス図に存在するモデル記述と矛盾する多重度もしくは関連の誤りを被験者が発見し、修正を行う。実験に用いた問題 2 を例にとって説明する。以下に示すのは問題 2 の問題文およびモデルである。

**問題** 以下のモデルを読み、次の 2 つの設問に回答しなさい。

- (1) クラス図に存在する誤り（モデル記述と矛盾する多重度）見つけて修正しなさい。
- (2) クラス図修正後のモデルについて、典型的な場面を表現するインスタンス図を 1 つ描きなさい。

**モデル化対象** ある団地における、住人と所有している車の関係を描いたモデルである。

**モデル記述**

- 住人の中には、車を所持していない人もいれば、複数台所持している人もいる。
- 共有所有および所有者のいない車は存在しない。

**クラス図**



<モデル記述>において、「住人の中には、車を所持していない人もいれば、複数台所持している人もいる」とある。ここで問題 2 の<クラス図>を見ると、クラス「住人」から見たクラス「車」の多重度は「1」である。よって、このクラス図が表すモデルをインスタンス図で表現すると、「住人」のインスタンスは必ず 1 つ、「車」のインスタンスとリンクするようになり、先に示した<モデル記述>と矛盾する。被験者は「住人」から見たときの「車」の多重度を「1」から「\*」または「0..\*」に修正を行う。以上がクラス図修正課題の一例である。

この作業は、3.2 節の UC1 および UC2 において、本システムが有効であるかどうかを測るものである。以下に示す手順は、本システムを利用した場合のクラス図修正作業のモデルケースである。

- (1) 被験者が、誤りのあるクラス図に対応したインスタンス図を生成する。
- (2) (1) で生成したインスタンス図を読み、誤りの部分に気がつく。
- (3) 被験者が、(2) において誤りだと考えた部分を修正する。
- (4) 修正したクラス図について、インスタンス図を生成する。
- (5) (4) で生成したインスタンス図を読み、正しく修正されていることを確認する。
  - (1), (2) が観察されれば、UC1 が、(4), (5) が観察されれば、UC2 の妥当性が確認で

きる。UC1 および UC2 が確認され、なおかつ修正結果が正しい場合、本システムは、クラス図の多重度やクラス同士の関連をクラス図から読み取る上で有効であったと言える。

なお、UC3 については、実験後に行うインタビューにて、ユーザに感想を聞き、効果があったかどうかを測る。

### 5.3 インスタンス図作成課題

インスタンス図作成課題は、被験者が、与えられたモデルの典型的な場面を表現したインスタンス図を書く課題である。この課題は、被験者がクラス図修正課題で施した修正を理解しているかどうかを測ることを意図している。被験者がモデルを正しく表現したインスタンス図を書くことができているならば、クラス図内にある多重度やクラス同士の関連が何を表しているのか読解できたとと言える。なぜなら、クラス図内の多重度や関連が何を表しているのかが理解できていれば、リンクの数や繋いでいるインスタンスが正しいインスタンス図を書くことができるからである。

ここでいう「典型的な場面」とは、クラス図から想像できる一般的な例のことである。多重度が「\*」や「1..\*」のようにその範囲に幅があるもの場合は、複数例を用いて、モデルから表現されるパターンをできるだけ網羅するように被験者に求めた。

## 6. 実験結果

### 6.1 概要

本章では、実験を実施した被験者 4 人をそれぞれ A,B,C,D とし、全体の総評とそれぞれの結果について述べる。なお、被験者 D にはシステムを使わず、同じ問題を解かせた。表 2 は、行に実施した問題、列に被験者をとり、被験者ごとの各問題の結果を集計したものである。行にある「修正」はクラス図修正課題、「作成」はインスタンス図作成課題を表す。列にある「時間」はクラス図修正課題、インスタンス図作成課題のどちらも解答するまでの時間である。「達成」は○であれば正答したことを示し、×であれば誤答もしくは解答不可であったことを表す。生成回数は「修正」と「作成」のそれぞれにおいて、本システムを用いてインスタンス図を生成した回数を示す。

### 6.2 被験者別の実験結果

#### 6.2.1 被験者 A

被験者 A は問題 2, 4, 5 については、適切に修正を行っていた。これらのインスタンス図作成課題についても、適当なインスタンス図が作成することができていた。

問題 2 では、モデル記述を見て、クラス図の誤りに気づき、修正を行っていた。

表 2 実験結果

被験者		問題 1		問題 2		問題 3		問題 4		問題 5		問題 6	
		修正	作成	修正	作成	修正	作成	修正	作成	修正	作成	修正	作成
A	時間	未実施		3'00"		5'30"		8'30"		5'00"		未実施	
	達成	-	○	○	○	×	×	○	○	○	○	-	-
	生成回数	-	-	0	0	7	6	9	0	2	1	-	-
B	時間	4'00"		5'10"		8'00"		10'40"		10'30"		5'00"	
	達成	-	○	○	○	×	×	○	○	○	○	-	○
	生成回数	-	-	4	0	4	4	1	5	1	0	-	-
C	時間	5'00"		5'40"		13'00"		6'50"		9'40"		13'50"	
	達成	-	○	○	○	×	×	○	×	○	○	-	○
	生成回数	-	-	0	0	7	6	9	0	2	1	-	-
D	時間	未実施		1'20"		5'00"		4'20"		9'00"		未実施	
	達成	-	-	○	○	×	×	○	○	○	○	-	-
	生成回数	-	-	-	-	-	-	-	-	-	-	-	-

問題 3 は、5 分 30 秒ほど多重度の修正とインスタンス図の生成をくり返し、モデル記述と矛盾している点を見つけようと試行錯誤していたが、被験者本人から解答の終了を申し出たため、途中で解答を打ち切った。インタビューにて、「履修クラスの役割や概念が分からなかった。」と答えた。問題 4 は、本システムが「百貨店」のインスタンスが 1 つもフロアのインスタンスとリンクしないケースのインスタンス図を示したために、誤りに気づいた。ただし、後のインタビューにて、「百貨店がフロアのインスタンスとリンクしないケースを確かめるため、何回もインスタンス図生成ボタンを押した。なかなか出てこないために確認するのが大変であった。」と述べた。問題 4、問題 5 について、本システムの利用方法が 5.2 で述べたモデルケースに合致していた。

#### 6.2.2 被験者 B

被験者 B は問題 4, 5 については、適切に修正を行った。インスタンス図作成課題は問題 1, 2, 4, 5, 6 について、適当なインスタンス図を作成することができた。被験者 B についても、問題 2 において、クラス図の修正ができていないのにも関わらず、インスタンス図作成課題は適切に遂行できている点が見られた。

被験者 B は、インタビューにて、インスタンス図を書くのが問題を経るごとに面倒な作業になったので、自動生成することでその作業が軽減されることが良いと答えた。また、自動生成によって、自分が想定していないインスタンス図が提示されることもあるのではないかと答えた。

#### 6.2.3 被験者 C

被験者 C は問題 2, 4, 5 について、適切に修正を行った。インスタンス図作成課題では、

問題 1, 2, 5, 6 について, 適当なインスタンス図を作成することができた. 問題 4 ではシステムを使わずにモデル記述を読んで修正ができたが, インスタンス図作成課題では, クラス図修正後のモデルを表現したインスタンス図を作成することができなかった. 問題 5 については, 5.2 節で述べた手順で, システムを利用していった.

#### 6.2.4 被験者 D

被験者 D は問題 2, 3, 5 については, 適切に修正を行った.

問題 4 は 5 分ほど考えたが, クラス図内の多重度について, 誤りを見つけることができず, 被験者本人が終了を申し出た.

インスタンス図作成課題は, 問題 2, 3, 4, 5 について, 適当なインスタンス図が作成することができた. 問題 2 は, 被験者 A と同じく, モデル記述を見て, クラス図の修正を行った. 問題 4 では, クラス図の修正ができていないのにも関わらず, インスタンス図の作成は適切に遂行できていた.

## 7. 考 察

### 7.1 モデルが表現しうる状況を網羅的に把握することに対する効果

本システムでは, クラス図が表現しているモデルに対応したインスタンス図を生成のたびに異なるものを示し, 特徴的な現象として, 被験者 B の問題 2 および被験者 D の問題 4 に見られる, クラス図の多重度の修正が適切に行われなかったにも関わらず, インスタンス図作成課題においては, 題意を満たすインスタンス図を作成していたことが挙げられる.

まずは被験者 B のケースについて述べる. 問題 2 のクラス図修正課題の正答は「住人」からみた「車」の多重度を「1」から「\*」にし, 「車」からみた「住人」の多重度を「\*」にすることである. このケースでは, 被験者 C は前者の多重度のみを修正しているため, 「多対多」関連のクラス図になっている. 多重度「\*」と「1」が包含関係にあるため, 被験者 C の描いたインスタンス図は修正後のクラス図と矛盾しない. よって, 一見, クラス図を正しく修正したかのように見える. 次に被験者 D のケースについて述べる. 問題 4 のクラス図修正課題の正答は, 「百貨店」からみた, 「フロア」の多重度「\*」を「1..\*」に修正することである. 多重度が「\*」である場合, 「フロア」のインスタンスを持たない百貨店のインスタンスが生成され得るため, 問題 4 のモデル記述と矛盾する. しかし多重度「\*」と「1..\*」は包含関係にある. よって被験者 B のように 0 のケースを見落としていると, クラス図の多重度を「\*」のままであっても, インスタンス図を書くとき「1..\*」の多重度と矛盾しないインスタンス図を書くことができる. このケースのような多重度の誤りに気づきにくいと考

えられる. 以上のことから, UML 初学者がインスタンス図を書くとき, 対象のモデルが表現しうるケースを網羅的にイメージできない場合があることが推察される.

一方, 被験者 A は問題 4 において, 本システムが「フロア」のインスタンスを 1 つも持たない「百貨店」のインスタンスが存在するインスタンス図サンプルを提示したため, クラス図内の誤りに気づくことができた. これは 3.2 節で述べた UC1 に対して, 本システムの有効性を示すものである.

### 7.2 抽象クラスの継承が含まれているモデルの理解に対する効果

本システムでは, 継承関係, および抽象クラスで表現されるモデルについても, 対応したインスタンス図を示すことができる. 我々は当初, 初学者に対する抽象クラスが含まれるモデルの理解については UC1 が有効であると推測していた. 評価実験の結果, この内容が含まれている問題 5 において, 2 人の被験者がクラス図の修正後, 自身の修正内容の確認のために本システムを利用していった. この 2 人に対してインタビューを実施したところ, 「モデル記述とクラス図を見て, 矛盾している点についてはおおよその検討がついていて, システムを使ってみると思った通りであった. 施した修正内容が自分の意図した通りのものなのか自信がなく, その確認のために利用した.」といった旨の返答が得られた. これにより UC1 および UC2 の妥当性が確かめることができ, 本システムが初学者の抽象クラスの理解に役立ったといえる.

### 7.3 提示するインスタンス図のバリエーション

評価実験後のインタビューで, システムを利用した 3 人の被験者に対して, 「自分が考えつかなかった状況を表現したようなインスタンス図は生成されたか」を聞いた. これは, UC3 の妥当性を確かめるためである. しかし, このユースケースを指示するような回答は得られなかった. その原因として, (1) 実験問題のクラス図に含まれるクラス数が少ない. (2) ランダムによる生成. の 2 点を挙げる.

まず (1) についてであるが, システムを利用して解くことを設定した問題の中で, もっともクラス数が多いもので 5 つであった. 被験者に対するインタビューでは「クラス数が 2 つや 3 つであれば, 自分の頭の中でイメージができるので, わざわざインスタンス図を提示されなくても良い.」との意見があった. クラスの多いクラス図モデルからインスタンス図を生成する場合, 第 4 章で示したように部分最適を積み重ね, モデル全体をインスタンス図で表現するアルゴリズムでは図が煩雑になる. よって, 例えば生成するインスタンス図全体で用いるインスタンスの数の上限をあらかじめ決定し, その数にあわせてインスタンス図を生成するような全体最適のアルゴリズムによるインスタンス図生成機能も実装し, 場合に

じて使い分けられるのが良いのではないかと考える。

次に、本システムでは、多様なインスタンス図を提示することでクラス図の多重度やクラス同士の関連を確認、インスタンスの生成個数やそのリンクの仕方をランダムに決定する。よって実際には、同じような状況を描いたインスタンス図が生成されてしまうケースが散見された。加えて、現在は特定のケースを狙ったインスタンス図を作成することはできない。1度生成したインスタンス図サンプルと同じものは生成しないようにし、少ない生成回数でモデル全体を網羅できるように、インスタンス図サンプル生成方法の改良が必要である。

## 8. おわりに

本稿では、クラス図が表現しているモデルに対応したインスタンス図を自動で生成しユーザに提示するインスタンス図自動生成システムの概要と、それをを用いた場合の学習効果について述べた。モデリングの経験が浅い者に対して実施した評価実験の結果、クラス図内にある多重度や関連の読解において有用であるという感触を得られた。引き続き、システムの改良およびより多くの被験者による評価実験を行う予定である。

## 参考文献

- 1) 児玉公信：UML モデリングの本質 良いモデルを作るための知識と実践，日経 BP 社 (2004)。
- 2) Hasker, R.W. and Rowe, M.: UMLint: Identifying defects in UML diagrams, *118th Annual Conference of the American Society for Engineering Education* (2011)。
- 3) Hasker, R.W.: UMLGrader: an automated class diagram grader, *J. Comput. Sci. Coll.*, Vol.27, pp.47-54 (2011)。
- 4) 上原幹正, 増田英孝, 笠原 宏：オブジェクトモデルの作成過程における四要素の役割分担についての考察，全国大会講演論文集第 55 回平成 9 年後期 (1), pp.393-394 (1997)。
- 5) 奥平光進, 上原幹正, 増田英孝, 笠原 宏：オブジェクトモデルを構築する際に可能な支援機能の実装，全国大会講演論文集第 53 回平成 8 年後期 (1), pp.249-250 (1996)。
- 6) 野沢光太郎, 松澤芳昭, 酒井三四郎：矛盾の発見を支援するモデリング初心者向けの UML 図作成ツール，教育システム情報学会 2011 年春学生研究発表会, pp.176-177 (2011)。
- 7) 杉浦啓孝, 松澤芳昭, 酒井三四郎：凝集度と結合度に注目する OOD 学習支援システム Fourcs の提案，情報処理学会第 73 回全国大会講演論文集 Vol.1, pp.439-440 (2011)。
- 8) 竹政昭利：UML モデリング技能認定試験<入門レベル (L1)>問題集-UML2.0 対応，技術評論社 (2007)。

## 付 録

### A.1 問題 1

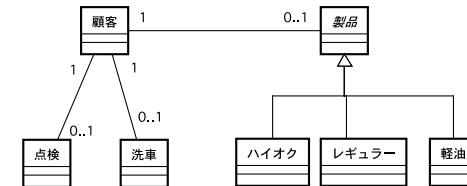
問題 以下のモデルを読み，典型的な場面を表現するインスタンス図を 1 つ描きなさい。

モデル化対象 あるガソリンスタンドの販売モデルである。

モデル記述

- あるガソリンスタンドでは，顧客は，ハイオク，レギュラー，軽油のいずれかの製品を購入する。
- 顧客は，点検や洗車をすることもある。

クラス図



### A.2 問題 2

問題 以下のモデルを読み，次の 2 つの設問に回答しなさい。

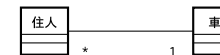
- (1) クラス図に存在する誤り (モデル記述と矛盾する多重度) 見つけて修正しなさい。
- (2) クラス図修正後のモデルについて，典型的な場面を表現するインスタンス図を 1 つ描きなさい。

モデル化対象 ある団地における，住人と所有している車の関係を描いたモデルである。

モデル記述

- 住人の中には，車を所持していない人もいれば，複数台所持している人もいる。
- 共有所有および所有者のいない車は存在しない。

クラス図





### A.3 問題 3

問題 問題2と同様のため省略

モデル化対象 ある大学での授業履修システムである。

モデル記述

- 学生は複数の授業を履修できる。
- 履修を希望した授業は、必ず受講できるものとする。

クラス図



※履修クラスは、学生の成績や履修年度などを記録するクラスである。

### A.4 問題 4

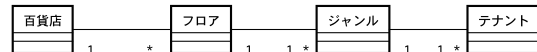
問題 問題2と同様のため省略

モデル化対象 ある百貨店の店内構成のモデルである。

モデル記述

- ある百貨店の店内構成のモデルである。
- ある百貨店はいくつかのフロアで構成されている。
- 各フロアには1つ以上のジャンル（婦人服、生活雑貨など）が設定されている。
- ジャンルには、それぞれに応じたテナントが1つ以上入っている。
- 1つのジャンルが複数のフロアにまたがることはない。

クラス図



### A.5 問題 5

問題 以下のモデルを読み、次の2つの設問に回答しなさい。

- (1) クラス図に存在する誤り（モデル記述と矛盾する関連）見つけて修正しなさい。
- (2) クラス図修正後のモデルについて、典型的な場面を表現するインスタンス図を1つ描きなさい。

モデル化対象 ある会社の社員の社員管理のシステムである。

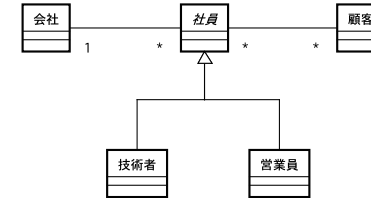
モデル記述

- 社員は、技術者と営業員の2職種の社員がいる。
- 営業員の社員には、担当する顧客がいる。なお、技術者には担当する顧客は存在し

ない。

- 複数の営業員が1つの顧客を担当することもあれば、1人の社員が複数の顧客を担当することもある。

クラス図



### A.6 問題 6

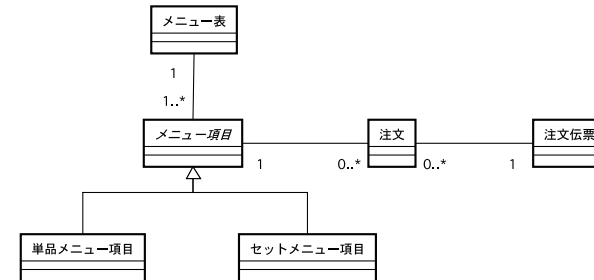
問題 以下のモデルを読み、典型的な場面を表現するインスタンス図を1つ描きなさい。

モデル化対象 あるレストランでの注文システムである。

モデル記述

- 注文伝票は客ごとに作成する。
- メニューには単品メニューとセットメニューがある。

クラス図



※注文クラスは数量と金額などを記録するクラスである