

教授者のクラス配置に基づいた初学者向けクラス図作成支援手法

齊藤 悠太 國分 健弘 田中 昂文 櫛山 淳雄 橋浦 弘明

現在、情報系の高等教育機関ではソフトウェアの開発対象に関するモデリング演習が行われている。演習中に学習者が作成するモデルには要素の欠落などの欠陥が多く存在するだけでなく、クラス図の要素のレイアウトが教授者が作成した正答例と大きく異なっていることが多い。

本論文では、このような問題に着目し、演習中の学習者に対して成果物の正誤を示すだけでなく、学習者のクラス図のレイアウトを教授者のレイアウトに自動変換することで、モデリング演習中の学習者への効果的な支援を行うための手法を提案する。提案した手法をツールとして実装し、評価を行った結果、レイアウトの自動変換が学習者への有効なフィードバックになることが示された。

1 はじめに

情報系の高等教育機関では、学習者の情報システムの設計、構築を行う能力を培うためにソフトウェアの開発対象に関するモデリング演習を行っている。モデリング演習は、構成主義的な課題を通じて、学習者が概念に対する理解を深め、経験的にその知識を積み重ねていくことを主な目的としている [1]。

モデリング演習では成果物の表記方法として、主に ERD, DFD, UML(クラス図)などの図式言語が用いられている。Chren ら [5] は、学習者がモデリング演習で作成する UML のクラス図について、他の図と比較して特に誤りが多いことを指摘している。

このような問題が発生する原因として、演習特有の問題が考えられる。教育機関外で一般に行われてい

るソフトウェア開発においても、モデリング作業が重要な作業であることに変わりはない。この際、モデリング作業の始点となる要求はプロジェクトオーナーや顧客から文書や口頭、言外の方法も含めた様々な手段で提供されるだけでなく、開発者が要求を正確に獲得できているかどうかについて不安がある場合には、要求の提供元に確認の質問をしたり、必要に応じてヒアリングを行って提供される情報を自分で増やすことができることが多い。その一方で、学習者が行なうモデリング演習では、演習の始めに教授者から簡単な要求文が与えられるのみであることが多い。追加の情報を得るために学習者が教授者に質問をすることは可能であるが、多数の学習者に対して十分な数の教授者が割り当てられていることは少ない。よって、学習者には限られた要求文を駆使して膨大な解空間から要件に適したモデルを選定、作成する必要があるが、このような作業は初学者には難しい作業である。

また、モデリング演習の終了時に、教員は正解例となるクラス図を学習者に提示して、解説を行うことがある。しかしながら、学習者はその正答例や解説の意図を十分に理解できないことが多い。このため、学習者が新たな課題に取り組むと依然として要求を正確に表現できず、要素が欠落や誤りを含んだ成果物を繰り返し作成してしまうという問題もある [5]。

* A method for assisting novices creating class diagrams based on the instructor's class layout.

This is an unrefereed paper. Copyrights belong to the Author(s).

Yuta Saito, Takehiro Kokubu, 日本工業大学, Nippon Institute of Technology.

Takafumi Tanaka, 玉川大学, Tamagawa University.

Atsuo Hazezama, 東京学芸大学, Tokyo Gakugei University.

Hiroaki Hashiura, 日本工業大学, Nippon Institute of Technology.

初学者のモデリングが困難な原因として、前述のものも含めて以下のような問題が指摘されている [29] [9].

- UML の用語や設計方法に関する知識不足
- ツールの使い方に関する不理解
- 問題文に記載されたシステムのドメインや仕様の理解不足
- 教員のフィードバックが不十分（時間や労力のコストの問題）
- （プログラミングと比較して）開発環境から得られるフィードバック（出力やエラーメッセージなど）が不十分

本研究は、モデリング演習がプログラミング演習と比較して、開発環境から得られるフィードバックが不十分であるという問題に着目した。一方で、過度なフィードバックを与えると正解例を提示しているのと変わらず、学習者の学習効果を阻害する可能性も指摘されている [12]。このような問題を同時に解決するアプローチとして、一戸ら [11] は学習者に正解例と成果物の類似度を 0.0~1.0 で表される数値で提示することにより、学習者が正解例に頼ることなく自らの試行錯誤により成果物を自力で正解例に近づけることができる環境を提案している。

一方で、このような類似度のみのフィードバックでは成果物に対する学習者の実感と乖離が生じる場合がある。モデリング演習で扱われるクラスに含まれる要素はそれほど多くはないため、例えば属性を 3 つ持つクラスの場合、1 つの属性が正解例に含まれていて、2 つの属性が誤っている場合、類似度は学習者に 0.33 と提示される。膨大な解空間から学習者が選択した 3 つの属性のうち 1 つが正解しているのであれば、本来はそれを足がかりに属性を類推して類似度を向上させることが期待される。しかしながら、学習者は自分の成果物が否定されたように感じてしまうことがあり、この成果物を放棄して新たに一から成果物を構築しなおそうとすることがある。このようなことにより、類似度が安定して向上せず、不規則に変動してしまうという問題があった。

本研究では、前述の問題を解決するために、さらにモデルのレイアウトに着目した。リバースエンジニア

リングの分野ではクラス図のレイアウトが読み手の理解に与える影響について、多くの研究が行われている。例えば、Störrle は [23] クラス図のレイアウトは、レイアウトのルールや美的基準といわれる指針に従うことで、可読性が向上することを指摘している。レイアウトのルールとしては、クラス図の要素同士を重ねて配置しないという基本的なルールから、親クラスを子クラスを中心に配置するというルール [7]、さらにデザインパターンにおいてクラスをまとめて配置（クラスタ化）するなど、クラス図特有のルールまで多岐にわたるものが提案されている。

本論文ではクラス図のレイアウトを用いて学習者のモデリングを支援する手法を提案する。この手法は上述の学習者が直面する課題を解決するために、クラス図（正解例）のレイアウトを支援に用いる。本論文におけるレイアウトとはクラスの配置を指す。本研究の目的は、以下の 2 点を調査することで、学習者のモデリング演習においてクラス図のレイアウトを用いた支援が有益なフィードバックとなるかを明らかにすることである

- 成果物の類似度への影響
- モデリング演習中の類似度の推移

本論文の構成は以下の通りである。第 2 節では、クラス図作成支援とクラス図のレイアウトに関する研究を紹介する。第 3 節では、提案手法であるレイアウトの自動変換について説明する。第 4 節では、RQ とその評価方法について述べる。第 5 節で実験結果を示し、その考察を行う。第 6 節では本論文の結論と今後の展望について述べる。

2 関連研究

2.1 クラス図の支援

モデリング演習において、学習者に対して成果物の自動でフィードバックを行う研究は多数行われている。その例を表 1 に示す。

欠落要素の検知を行う研究として Wei ら [25] が挙げられる。Wei ら [25] は同一の学習者が作成したクラス図、シーケンス図、アクティビティ図から欠落した要件（セマンティックホール）を発見するための手法とフレームワークを提案している。この方法は正解例

表 1 研究の分類

#	分類	文献
1	欠落要素の検知	[25][22][13][20][17] [3][10]
2	自動採点	[15][11]
3	その他	[12][19][18][6]

を使わずに、学習者の作成したモデルの一貫性や完全性を向上させることができる利点がある。一方で、この手法は学習者が複数の成果物を作成した際に生じる表記のぶれに着目しているため、そもそも学習者が完全に見落としていて全てのモデルに対して記述が行われていない要件をセマンティックホールとして検出することはできない。したがって、本研究が想定しているクラス図のみのモデリング演習者、完全な初学者の支援には適していない。

Soler ら [22] は、UML クラス図の教育と学習を支援するウェブベースのツールを開発した。このツールは学習者に練習問題を与え、成果物を自動的に修正する。修正後、学習者は検出された相違点の概要を示すフィードバックを受け取ることができる。

Krusche ら [13] は、オンライン・モデリング・エディターを作成した。Krusche ら [13] のエディタは要素の横に欠けている要件を強調表示することができるため、学習者は個別かつ直接的なフィードバックを受け取ることができる。

これらの欠落要素の検知の共通の利点として、欠落する要素を明示することで、学習者が成果物を自ら修正でき学習者の意欲が向上するということが挙げられる。一方で、モデリング中に学習者がフィードバックを受けることができない、何度もフィードバックを受けることができないといった問題がある。

成果物、特にクラス図を自動採点する研究は多数行われている [2][30][4]、一戸ら [11] は正解例と成果物の類似度をフィードバックとして学習者に返すツールを開発した。これにより、モデリング中に学習者はフィードバックを受け取ることが可能になった。一方で、前節で述べたとおり、フィードバックが学習者の実感との乖離している場合、は安定して類似度を向上

させることは難しい。

Karasneh [12] らはソフトウェア設計のための事例集をモデリング演習に活用する手法を提案しており、事例集 (UML リポジトリ) が学習者のモデルの品質を向上させることに役立つと述べている。その一方で、このような方法では学習者が良い事例と悪い事例の判別を行うことができない学習者の存在や、事例を使用することに対し、学習者が「教授者の怠慢である」「現在の内容を学ぶ上で事例は必要以上の資料である」と感じる問題が発生すると指摘している。

この他にも、Rodrigues ら [19] や Reuter ら [18] はモデル (クラス図) を AR や 3D で表現することで、要素を直感的に理解させることを提案している。また、Rocco ら [6] はニューラルネットワークを用いてモデリングの次の行動を提案するシステムを提案している。

このように学習者のクラス図作成を支援する様々な研究が行われているが、モデリング中に何度もフィードバックを受け取ることができたり、フィードバックがヒントとして適切なレベルであることを確認している研究は少ない。

2.2 クラス図のレイアウト

前述のとおりリバースエンジニアリングの分野では、クラス図を自動生成する際にどのようなレイアウトが好ましいのかについて議論されている [23][14][21]。

Störrle [23] はレイアウトの基準を順守している「良い」レイアウトが認知負荷のいくつかの側面に対して有益な効果をもたらすことを明らかにした。特に、初心者のモデラーは上級モデラーよりもレイアウトから多くの恩恵を受けることを指摘している。

Adrian ら [14] はソフトウェアの可視化のために一貫したレイアウトを提案した。このレイアウトでは、ソフトウェアアーティファクトの位置はその語彙を反映し、距離は語彙の類似性に対応する。実体の語彙を m 次元空間に配置し、Multidimensional Scaling を用いて、これらの位置を 2 次元ディスプレイにマッピングすることで、最終的にデジタル標高や等高線が適用されたソフトウェアの地図が作成される。

Bonita ら [21] はデザインパターンをクラスタ化することで役割検出の精度が向上することを明らかにしている。また、初学者と経験者の視線の動きの違いがあることも示している。

Helen ら [16] の研究では、表記の仕方が作成者の直感と合わない場合、安心感がなく熱心にクラス図を見るため、被験者はエラーを特定しやすくなると述べている。

Ahmar ら [8] は、可読性を向上させるために、場面に合わせてレイヤーを変更する方法を提案している。このレイヤーを使うことで、元のクラス図の要素に背景色や明度、線の太さ、テキストのフォントなどの情報を追加することが可能になる。また、元のクラスを見えなくするレイヤーも存在し、表示するクラスを減らす。場面に合わせてレイヤーを変えることで、大きなクラス図の複雑さを減らし、コミュニケーションをより効果的にする。

このように、クラス図のレイアウトの良し悪しによって、理解に影響があることは明らかである。一方で、これらのレイアウトのルールはクラス図の自動生成などリバースエンジニアリングの分野では活用されているものの、モデリング分野への適用した研究は少ない。

2.3 田中らの類似度

本論文では、提案手法や実験、評価において、成果物と正解例から田中ら [28] の類似度を用いて算出する。田中らは類似度を以下のように定義している。以下の内容のスナップショットが本研究における成果物に該当する。

以下、スナップショットを S 、正解例のクラス図を A と記述する。また、クラス図の要素（クラス、属性、関連）の要素名（クラス名、属性名、関連名）の類似度を計算する関数として $nameSim(E1, E2)$ （要素 $E1, E2$ の要素名の類似度を 0（最小）～1（最大）の範囲で返す関数）を使用する。現在は、Xing ら [26] の研究で定義されている $nameSimilarity$ を $nameSim$ として使用することを検討している。 $nameSimilarity$ は、比較対象文字列に共通する「隣

接する 2 文字」の数を計算し、その 2 倍を比較対象文字列の文字列長の和で割った値を返す関数であり、従来から用いられてきた LCS (Longest Common Subsequence) アルゴリズムよりも要素名の比較に適していると述べられている。

(1) クラスの比較

①比較対象クラスの抽出とクラス名類似度 CNS (Class Name Similarity) の計算

クラス $C_i \in S$ 、クラス $C_j \in A$ とする。ある C_i と、対応する（類似度を計算する対象となる） C_j の組を比較対象クラスの組と呼ぶ。以下に計算手順を示す。

- I ある C_i について、比較対象クラスの組に属していないすべての C_j とクラス名の類似度 $nameSim(C_i, C_j)$ を計算する。
- II 計算されたクラス名の類似度の最大値 $\max(nameSim(C_i, C_j))$ が閾値 T_n を超えた場合、その (C_i, C_j) を比較対象クラスの組として抽出する。また、 (C_i, C_j) のクラス名類似度 CNS を、 $nameSim(C_i, C_j)$ とする。
- III 手順 I. および手順 II. をすべての C_i について行う。
- IV 比較対象クラスの組に属していない C_j を不足クラスとし、不足クラスの数 NMC (Number of Missing Classes) として計算する。

②属性名類似度 ANS (Attribute Name Similarity) の計算

比較対象クラスの組 (C_i, C_j) の各属性について、クラスと同様の手順で比較対象属性の組（類似度を計算する対象となる属性の組）を抽出し、各組の属性名類似度 ANS を計算する。また、比較対象属性の組に属していない C_j の属性を不足属性とし、その数を NMA (Number of Missing Attributes) として計算する。

③クラス類似度 CS_i (Class Similarity) の計算

あるクラス $C_i \in S$ に関する正解例との類似度 CS_i は、クラス C_i が比較対象クラスの組に属している場合は以下の式で計算し、そうでない場合は 0 とする。式中の N_A は C_i の属性数を表す。

$$CS_i = \frac{CNS + \sum ANS}{1 + NA + NMA}$$

A, S の全関連に関する類似度 CS_{all} は次の式で計算する。式中の NRS は S の関連の数を表す。

$$CS_{all} = \frac{\sum CS_i}{NC_s + NMC}$$

(2) 関連の比較

①比較対象関連の抽出

関連 $R_i \in S$ と関連 $R_j \in A$ から、それぞれの対応する端のクラスの組が比較対象クラスの組である (R_i, R_j) を比較対象関連の組として抽出する。比較対象関連の組に含まれない R_j を不足関連とし、不足関連の数を NMR (Number of Missing Relationships) として計算する。

②関連名類似度 RNS (Relationship Name Similarity) の計算

比較対象関連に属する関連の組について、 $\text{nameSim}(R_i, R_j)$ が閾値 T_n を超えた場合は (R_i, R_j) の関連名の類似度 RNS を $\text{nameSim}(R_i, R_j)$ とし、そうでない場合は 0 とする。

③多重度の類似度 CaS の計算

比較対象関連に属する関連の組 (R_i, R_j) の対応する両端について、多重度の類似度 CaS (Cardinality Similarity) を決定する。今回対象とするクラス図では「1」、「0..1」、「1..*」、「*」の4種類の多重度を用いることとし、これらの組み合わせに対してあらかじめ類似度を設定しておく。比較対象の多重度の組に応じて、対応する類似度の値を決定する。

④関連類似度 RS_i (Relationship Similarity) の計算

関連 $R_i \in S$ の正解例との類似度 RS_i は、対象の関連 R_i が比較対象関連に属している場合は以下の式で計算し、そうでない場合は 0 とする。

$$RS_i = \frac{RNS + \sum CaS}{1 + 2}$$

A, S の全関連に関する類似度 RS_{all} は次の式で計算する。式中の NRS は S の関連の数を表す。

$$RS_{all} = \frac{\sum RS_i}{NR_s + NMR}$$

(3) クラス図の類似度の計算

クラス図の組 (S, A) の類似度 CDS は以下の式で計算する。

$$CDS = \frac{CS_{all} + RS_{all}}{2}$$

3 提案手法

本論文ではクラス図 (正解例) のレイアウトを支援に用いて、成果物のレイアウトを正解例のレイアウトに自動変換する。具体的には、学習者が作成したクラス図 (成果物) のクラスの配置を正解例のクラスの配置と同じになるように再配置することである。この節ではレイアウトの自動変換について説明する。レイアウトの自動変換の例を図1に示す。右が教員が作成した正解例、左が学習者が作成した成果物、そして中央が正解例 (右) のレイアウトになるように成果物 (左) のレイアウトを変更した成果物である。レイアウトの自動変換では、成果物のクラス (カート, 顧客, 注文, 商品) を正解例のクラス (カート, 顧客, 注文, 商品) の座標へ再配置している。このとき、クラスや関連を作成することはなく、成果物に存在するクラスのみを用いて再配置を行う。そのため、これらのクラスの再配置が行われた成果物はクラス, 属性, そして関連が欠けている, もしくは余分に存在する場合も考えられる。学習者は, 正解例のレイアウトに変わることで, 正解例を作成した教員の視点でシステムを見ることで, 今の成果物に足りない要件を作成するように促される。

レイアウトの自動変換は瞬間的に行うことができ, 学習者は何度でもレイアウトの自動変換が行える。レイアウトの自動変換で学習者が得ることができる情報は正解例のクラスの配置のみである。学習者はレイアウトを変更する過程で必要となる類似度などの情報は得ることができない。

本論文では, モデル描画ツール KIfU [24] でレイアウトの自動変換が行えるように, ツールを拡張した。

レイアウトを変更する方法は大きく3つの工程で行われる。それが「対応するクラスの探索」, 「クラスの再配置」, 「元の関係を維持した関連の配置」である。以下それぞれについて述べる。

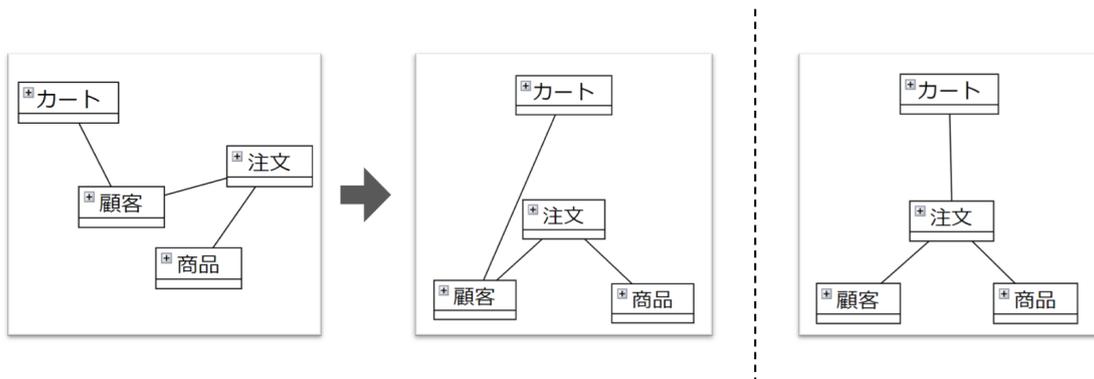


図1 レイアウトの自動変換 (左：成果物，中央：レイアウトの自動変換後の成果物，右：正解例)

3.1 対応するクラスの探索

はじめに、「対応するクラスの探索」について説明する。これは、再配置を自動的に行うためには正解例と成果物に含まれる各要素がどのように対応しているのかを同定する必要があるからである。

対応するクラスの探索では、クラスごとの類似度 CS を用いて成果物のクラスの中から「対応するクラス」を見つける。対応するクラスの定義は、正解例のある1つのクラスに対する CS の値が0.6以上かつ成果物のクラスの中で最も大きいことである。つまり、正解例のある1つのクラスに対して、対応するクラスは成果物のクラスの中から1つしか選ばれない。KIFUには、元々クラス図の類似度を算出する機能が備わっているため、対応する正解例のクラスと成果物のクラスはペアで記録されている。

3.2 クラスの再配置

次に「クラスの再配置」について説明する。座標の変更では、1つ目の工程で成果物の中から見つけた対応するクラスを正解例の対応するクラスの座標に再配置するために、成果物のクラスの座標データのみを書き換える。そして、1つ目の工程で対応するクラスと判断されていない成果物のクラスについては、描画面の左上、座標(0,0)に再配置するため、クラスの座標データをすべて(0,0)に書き換える。このとき、座標(0,0)に再配置されたクラスが複数存在する場合は、クラス同士が重なる可能性がある。KIFUでは、

クラス図を1回編集するごとにBase64コードにしてデータベースに登録している。そこで、対応する正解例クラスと成果物クラスのペアをもとに、Base64コードの座標の値を正解例クラスの値に変更する機能を追加した。

3.3 元の関係を維持した関連の再配置

最後に「元の関係を維持した関連の再配置」について説明する。レイアウトの自動変換では、クラスの座標のみを変化させるため、クラスとクラスを繋いでいる関連は元の関係を崩さないように再配置する。これは、学習者がレイアウトの自動変換によって、自分で作成したクラス図だと認識できなくなることを防ぐためである。このとき、座標(0,0)に再配置された対応するクラスと判断されていないクラスとの関連も残るため、関連とクラスが重なる可能性がある。KIFUでは、クラス図のデータはクラスの座標データのみを書き換えるため、関連は元の関係を維持するように再配置される。

これら3つの工程を踏むことでレイアウトの自動変換が完了する。本論文では、学習者が任意のタイミングで何度でもレイアウトの自動変換が行えるようにレイアウトの自動変換ボタンを作成した。レイアウトの自動変換ボタンが押されたときに、自動的に一連の工程が行われる。

4 実験

4.1 Research Question(RQ)

以下の Research Question に答えることで提案手法を評価する。

RQ1. 正解例のレイアウトは成果物作成にどのような影響を与えるのか？田中らの類似度 CDS によって学習者のクラス図への影響を調べる。また類似度 CDS だけでなく、クラスの類似度 CS_{all} と関連の類似度 RS_{all} からクラスと関連のどちらに効果があるのかを評価する。

RQ2. レイアウトの影響はいつから表れるのか？このリサーチクエストでは、RQ1 と同様に類似度 CDS 、 CS_{all} 、 RS_{all} を用いる。モデリング中の類似度を計測することで、学習者の手詰まりの解消できているかどうかを評価する。

4.2 実験方法

学習者のクラス図のモデリングに対するレイアウト自動変換の有益性を説明するために、実験を実施した。被験者は日本工業大学情報メディア工学科の学生 20 名であり、以下の 2 つのグループに 10 名ずつ無作為に割り当てた。

- レイアウト自動変換機能を持つ KIFU [24] を使用するグループ (以下、実験群)
- レイアウト自動変換機能を持たない KIFU [24] を使用するグループ (以下、統制群)

実験では、被験者がシステム概要に基づいてクラス図を作成するモデリング演習を行った。このシステム概要は、わかば商社が従来 FAX や電話で行っていた商品の注文を電子化するための電子商取引システムの構築を目的としている。概要には、わかば商社の業務フローや電子化の要望が含まれている。この概要は文献 [27](p.63) を参考に著者が修正したものである。

実験手順は以下の通りであり、実験群と統制群のどちらも同じ手順で進めた。

1. 被験者に対し、クラス図と描画ツールについて説明を行う
2. チュートリアルとして被験者が自由にツールを使う

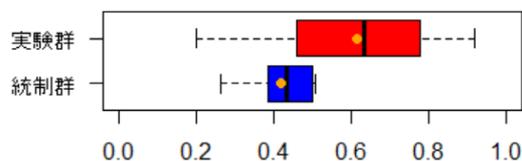


図 2 実験群と統制群の CDS の比較

3. 被験者に電子商取引システムのシステム概要が書かれた用紙を配布する
4. 被験者がツールを用いてモデリング演習を行う (1 時間)
5. 被験者がツール内でクラス図を提出する

実験群への描画ツールの説明では、基本的な操作の説明に加えて、対応する要素の定義と正解例のレイアウトに自動変換することについての説明を行った。このとき、類似度の定義の説明など詳細な説明は行っていない。そして、チュートリアルで実際に機能を使わせることで、どれだけクラスが一致していれば対応するクラスと判断されるのかを体感させた。

評価対象は、20 名の被験者が作成したクラス図である。これらのクラス図から類似度 CDS 、 CS_{all} 、および RS_{all} を算出し、両側の t 検定を行って有意差を検討した。また、クラスを分類する際、単語が 1 文字でも異なると CS の値に影響が出るため、クラス図に使用できる単語は問題文に記載されている単語に限定する制限を課した。

5 結果

5.1 RQ1: 正解例のレイアウトは成果物作成にどのような影響を与えるか？

モデリング演習で得られた類似度 CDS を図 2 に示す。類似度 CDS は t 検定の結果 $p=0.0235$ ($\alpha=0.05$) となり、有意な差が見られた。この結果から、レイアウトの自動変換は学習者のクラス図のモデリングに対して有益なフィードバックであることが示された。

次に、この効果がクラス図の各要素に対してどのような効果があるのかを調べた。 CS_{all} の結果を図 3, 2 に示す。 CS_{all} の検定の結果は $p=0.2886$ ($\alpha=0.05$) となり、有意な差は見られなかった。

一方で、クラスに関する結果は、著者らの予想とは

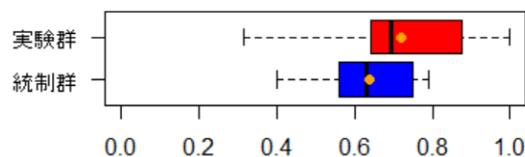


図3 実験群と統制群の CS_{all} の比較

異なるものであった。著者らは自動変換後のレイアウトの空白から、学習者が不足しているクラス存在に気付くと考えていたが、実験群と統制群の間に有意な差は見られなかった。

表2 各クラスの CS の平均値

#	クラス名	CS の平均値 (実)	CS の平均値 (統)
1	顧客	0.96	0.99
2	注文	0.52	0.65
3	カート	0.60	0.48
4	注文明細	0.53	0.05
5	商品	0.90	0.94
6	在庫	0.93	0.82

しかし、各クラスの CS に着目したところ、「注文明細」クラスにおいて有意な差を確認した ($p=0.0003734$, $\alpha=0.05$)。また、表2からわかる通り、有意な差ではないものの、「注文」クラスの CS は実験群より統制群の方が大きい。この結果は、「注文」と「注文明細」というクラス名が類似しているため、多くの学習者がこれら2つの概念を1つの概念として捉えたことが原因であると考えられる。クラスの再配置はクラス間の距離から不足しているクラス存在を示唆することはできるが、概念の理解に対する効果は限定的であると考えられる。

最後に RS_{all} の結果を図4に示す。 RS_{all} の検定の結果は $p=0.0704$ ($\alpha=0.05$) となり、有意な差が見られた。

この結果は、「注文明細」クラスの有無とクラスの配置が密接に関係していると考えられる。「注文明細」クラスが存在することで、3本の関連を作成する可能性

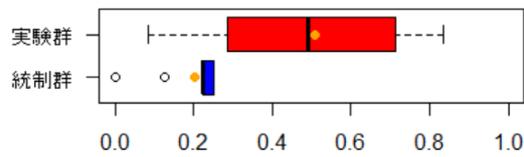


図4 実験群と統制群の RS_{all} の比較

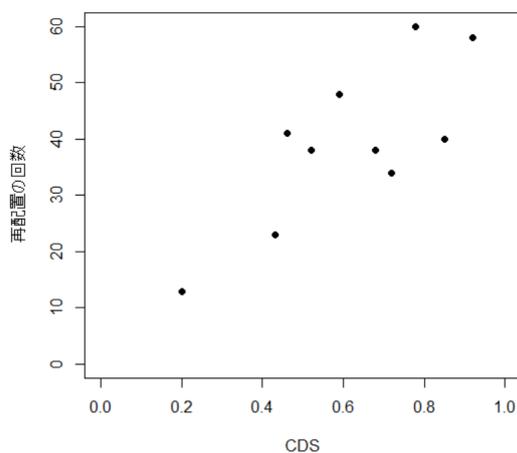


図5 類似度と支援回数の分布

が生じる。そして、クラスの配置により、学習者が作成する関連の選択肢が絞られていると考える。今回の演習で提出されたすべてのクラス図において、要素が重なっている箇所は確認できなかった。これは、学習者がレイアウトのルールについて言及せずとも、自然に要素の重なりを避けていることを示している。このような学習者の行動から、レイアウトによって選択肢を絞り込んでいると考えることができる。

CDS , CS_{all} , RS_{all} これらの類似度は、統制群より実験群の方が分散が大きい。この原因を調べるためにレイアウトの自動変換機能の使用回数を調べた。図5に再配置機能を使った回数と類似度の分布を示す。相関分析を行ったところ、 $r=0.7877$ となり、強い相関が見られた。このことから、類似度が高いクラス図を作成した学生は、より多くのフィードバックを求めると考えられる。

5.2 RQ2: レイアウトの影響はいつから表れる

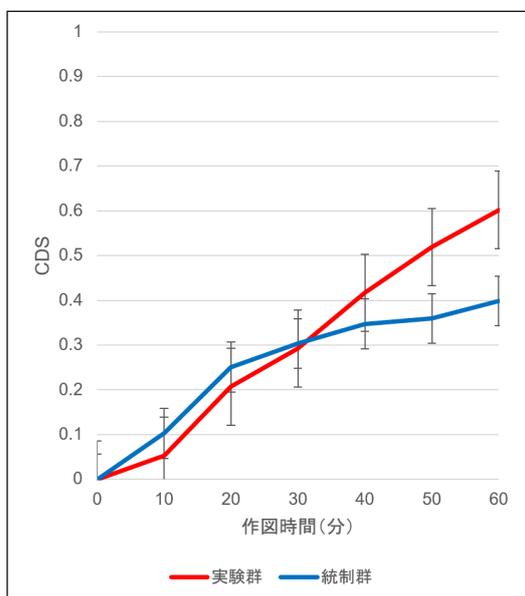


図6 実験群と統制群のCDSの推移

のか？

クラス図を1回編集するごとにクラス図を記録するKIFUの機能を用いて、モデリング演習中の類似度の推移について調査した。図6にモデリング演習中の類似度の変化を示す。

図6からわかるように、作図時間が50分までは、実験群と統制群に有意な差は見られない。実験群の分散は主に30~40分の間、2周目のサイクルの始まりに広がっていることがわかる。しかし、多くの被験者が関連を作成し始める20分以降、統制群のCDSはペースを落として緩やかに上がり続けるのに対し、実験群はペースを落とさずに類似度を向上させ続けている。この結果から、本手法が学習者のモデリングにおける手詰まりを解消する支援になっていると考えられる。そして、この統制群の結果から、学習者が一定のタイミングまでモデリングを行った場合、教員からのフィードバックなどを受け取らなければ非効率的な演習を続ける可能性があることを示唆している。そのため、演習を適切なタイミングで切り上げることも重要であると言える。

両群に共通する点として、10~20分間の類似度の上がり幅が大きい点、20分~40分間の類似度の

上がり幅が小さい点、そして40~50分間の類似度の上がり幅が大きい点が挙げられる。これは今回のモデリング演習において、類似度の推移にサイクルが存在していることを示唆している。このサイクルは1サイクル約30分周期であり、クラス、属性、関連などの要素を作成する0~30分の間が1周目、作成した要素の削除、移動、名前の変更等の修正を行う30分~60分の間が2周目である。そしてこのサイクルが存在すると仮定した場合、本手法の効果はクラス図の修正を行う、サイクルの2周目から表れているということになる。

本論文と同じKIFUを用いた一戸ら[11]の研究では、モデリング演習中に数値によるフィードバックを返している。しかし、学習者の実感とモデリングの評価結果に乖離が存在したため、類似度の向上が不安定である被験者も存在した。これに対し、本論文において実験群の類似度は安定して向上している。

6 まとめと今後の課題

ソフトウェア開発対象に関するモデリング演習は学習者にとって難易度が高く、学習者が作成するクラス図は誤りが多く、正答例とも大きく異なっていることが多い。この問題を解決するために本論文では、学習者のクラス図のレイアウトを教授者のレイアウトに自動変換する手法を提案し、提案した手法をツールとして実装した。

実験では演習を行い、レイアウトの自動変換機能が使えない実験群と使えない統制群の類似度を比較した。結果として、類似度は実験群の方が統制群より値が大きく、提案手法が学習者への有効なフィードバックになることが示された。他に特筆すべき点として、実験群の方が分散が大きいこと、演習中の両群の類似度の変化に共通する点が見られたことである。分散が大きい原因は、実験群の被験者によってフィードバックを求めるタイミング異なっていることによって、効果にばらつきが現れたと考える。演習中の類似度の変化から、クラス図のモデリングには周期があることを示唆していると考えられる。

今後の展望として、効果のばらつきを小さくするためにフィードバックの機会を増やす試みを予定して

いる。

謝辞 本研究の一部は JSPS 科研費 24K15214 の助成を受けた。

参考文献

- [1] Ben-Ari, M.: Constructivism in computer science education, Vol. 30, No. 1, Mar. 1998, pp. 257–261.
- [2] Bian, W., Alam, O., and Kienzle, J.: Automated Grading of Class Diagrams, *Proc. of the 2019 ACM/IEEE 22nd International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C)*, Sep. 2019, pp. 700–709.
- [3] Bogdanova, D.: Towards Personalized Feedback in a Smart Learning Environment For Teaching Conceptual Modelling, *Proc. of the 13th International Conference on Research Challenges in Information Science (RCIS2019)*, May. 2019, pp. 1–5.
- [4] Boubekur, Y., Mussbacher, G., and McIntosh, S.: Automatic assessment of students’ software models using a simple heuristic and machine learning, *Proc. of the 23rd ACM/IEEE International Conference on Model Driven Engineering Languages and Systems: Companion Proceedings (MODELS ’20)*, New York, NY, USA, Association for Computing Machinery, Oct. 2020.
- [5] Chren, S., Buhnova, B., Macak, M., Daubner, L., and Rossi, B.: Mistakes in UML Diagrams: Analysis of Student Projects in a Software Engineering Course, *Proc. of the 2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering Education and Training (ICSE-SEET)*, May. 2019, pp. 100–109.
- [6] Di Rocco, J., Di Sipio, C., Nguyen, P. T., Di Ruscio, D., and Pierantonio, A.: Finding with NEMO: a recommender system to forecast the next modeling operations, *Proc. of the 25th International Conference on Model Driven Engineering Languages and Systems (MODELS ’22)*, New York, NY, USA, Association for Computing Machinery, Oct. 2022, pp. 154–164.
- [7] Eichelberger, H.: Aesthetics of class diagrams, *Proc. of the First International Workshop on Visualizing Software for Understanding and Analysis*, Jun. 2002, pp. 23–31.
- [8] El Ahmar, Y., Gérard, S., Dumoulin, C., and Le Pallec, X.: Enhancing the communication value of UML models with graphical layers, *2015 ACM/IEEE 18th International Conference on Model Driven Engineering Languages and Systems (MODELS)*, Oct. 2015, pp. 64–69.
- [9] Ergin, H., Walling, I. L., Rader, K. P., and Dobbs, D. J.: A Study of Modeling Perception in a First-Time Modeling Class, *Proc. of the 2019 ACM/IEEE 22nd International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C)*, May. 2019, pp. 680–689.
- [10] Hasker, R. W.: UMLGrader: an automated class diagram grader, Vol. 27, No. 1, Evansville, IN, USA, Consortium for Computing Sciences in Colleges, Oct. 2011, pp. 47–54.
- [11] Ichinohe, Y., Hashiura, H., Tanaka, T., Hazeyama, A., and Takase, H.: Effectiveness of Automated Grading Tool Utilizing Similarity for Conceptual Modeling., Springer, Aug. 2018, pp. 117–126.
- [12] Karasneh, B., Jolak, R., and Chaudron, M. R. V.: Using Examples for Teaching Software Design: An Experiment Using a Repository of UML Class Diagrams, *Proc. of the Asia-Pacific Software Engineering Conference (APSEC2015)*, Dec. 2015, pp. 261–268.
- [13] Krusche, S., Reimer, L. M., von Frankenberg, N., and Bruegge, B.: An Interactive Learning Method to Engage Students in Modeling, *Proc. of the IEEE/ACM 42nd International Conference on Software Engineering: Software Engineering Education and Training (ICSE-SEET2020)*, Oct. 2020, pp. 12–22.
- [14] Kuhn, A., Loretan, P., and Nierstrasz, O.: Consistent Layout for Thematic Software Maps, *Proc. of the 15th Working Conference on Reverse Engineering (WCRE ’08)*, USA, IEEE Computer Society, Oct. 2008, pp. 209–218.
- [15] Modi, S., Taher, H. A., and Mahmud, H.: A Tool to Automate Student UML diagram Evaluation, *Academic Journal of Nawroz University*, Vol. 10, No. 2, Jun. 2021, pp. 189–198.
- [16] Purchase, H. C., Colpoys, L., McGill, M., Carrington, D., and Britton, C.: UML class diagram syntax: an empirical study of comprehension, *Proc. of the 2001 Asia-Pacific Symposium on Information Visualisation (APVis ’01)*, AUS, Australian Computer Society, Inc., Dec. 2001, pp. 113–120.
- [17] Reischmann, T. and Kuchen, H.: Towards an E-assessment tool for advanced software engineering skills, *Proc. of the 16th Koli Calling International Conference on Computing Education Research (Koli Calling ’16)*, New York, NY, USA, Association for Computing Machinery, Nov. 2016, pp. 81–90.
- [18] Reuter, R., Hauser, F., Muckelbauer, D., Stark, T., Antoni, E., Mottok, J., and Wolff, C.: Using Augmented Reality in Software Engineering Education? First insights to a comparative study of 2D and AR UML modeling, *Proc. of the 52nd Hawaii International Conference on System Sciences (HICSS2019)*, Jun. 2019.
- [19] Rodrigues, C. S. C., Werner, C. M. L., and Landau, L.: VisAr3D: An Innovative 3D Visualization of UML Models, *Proc. of the 2016 IEEE/ACM 38th International Conference on Software Engineering*

- Companion (ICSE-C)*, May. 2016, pp. 451–460.
- [20] Schots, M., Rodrigues, C. S. C., Werner, C., and Murta, L.: A Study on the Application of the PREViA Approach in Modeling Education, *Proc. of the XXIX International Conference of the Chilean Computer Science Society (SCCC2010)*, Nov. 2010, pp. 96–101.
- [21] Sharif, B. and Maletic, J. I.: An eye tracking study on the effects of layout in understanding the role of design patterns, *Proc. of the 2010 IEEE International Conference on Software Maintenance (ICSM'10)*, Sep. 2010, pp. 1–10.
- [22] Soler, J., Boada, I., Prados, F., Poch, J., and Fabregat, R.: A web-based e-learning tool for UML class diagrams, *Proc. of the IEEE EDUCON 2010 Conference*, Apr. 2010, pp. 973–979.
- [23] Störrle, H.: On the impact of layout quality to understanding UML diagrams: Diagram type and expertise, *Proc. of the IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC2012)*, Oct. 2012, pp. 49–56.
- [24] Tanaka, T., Hashiura, H., Hazeyama, A., Komiya, S., Hirai, Y., and Kaneko, K.: Learners' Self Checking and Its Effectiveness in Conceptual Data Modeling Exercises, Vol. E101.D, No. 7, Institute of Electronics, Information and Communication Engineers, Jul. 2018, pp. 1801–1810.
- [25] Wei, B., Delugach, H. S., Colmenares, E., and Stringfellow, C.: A Conceptual Graphs Framework for Teaching UML Model-Based Requirements Acquisition, *Proc. of the 2016 IEEE 29th International Conference on Software Engineering Education and Training (CSEET)*, Apr. 2016, pp. 71–75.
- [26] Xing, Z. and Stroulia, E.: UMLDiff: an algorithm for object-oriented design differencing, *Proceedings of the 20th IEEE/ACM International Conference on Automated Software Engineering, ASE '05*, New York, NY, USA, Association for Computing Machinery, Nov. 2005, pp. 54–65.
- [27] 株式会社テクノロジーアート: ゼロからはじめる UML モデリング入門, 株式会社翔泳社, Jan. 2003.
- [28] 田中昂文, 橋浦弘明, 樫山淳雄, 古宮誠一: 学習者のクラス図作成過程における成果物と正解例との類似度の変遷を用いた進捗状況可視化手法の提案, 情報処理学会ソフトウェアエンジニアリングシンポジウム 2015 論文集, Aug. 2015, pp. 206–207.
- [29] 小木曾禎, 遠山紗矢香, 湯浦克彦: 学生向けモデリング演習支援システムの開発と評価, 情報処理学会研究報告コンピュータと教育 (CE), Vol. 2011-CE-109, No. 5, Mar. 2011, pp. 1–9.
- [30] 宮島和音, 小形真平, 香山瑞恵, 海尻賢二: UML モデリング教育を支援するルールベースのクラス図採点支援ツール, 電子情報通信学会第 77 回全国大会講演論文集, Vol. 2015, No. 1, Mar. 2015, pp. 841–842.

A 付録

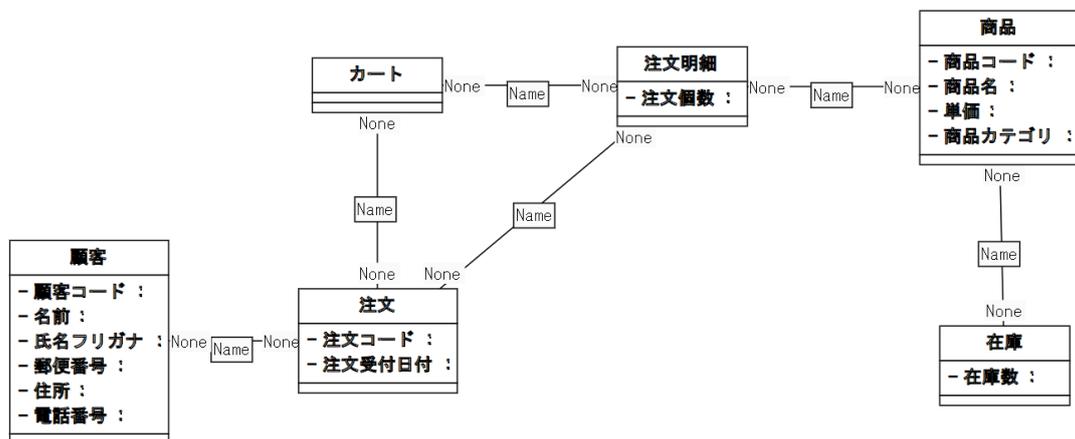


図 7 実験で用いた正解例