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

田中昂文[†1]　橋浦弘明[†2]　櫨山淳雄[†3]　古宮誠一[†4]

ソフトウェア設計の演習では，プログラミングにおけるコンパイル結果や実行結果のような作成中の成果物に関する情報が得られない．そのため，教授者が演習中に学習者の進捗状況（成果物がどの程度正解に近づいたか，進捗 or 停滞）を把握することは困難である．本研究ではクラス図を作成する演習を対象とし，作成過程における学習者の成果物と正解例との類似度の変遷を可視化することで進捗状況の可視化を行う手法を提案する．本手法により，教授者は各学習者の進捗状況を演習中に容易に把握することができるようになり，効果的な指導につながると期待される．

Progress Visualization Method by Visualizing Transition of Similarity of Learners' Artifact and Right Answer during Process of Class Diagram Creation Exercise

TAKAFUMI TANAKA[†1]　HIROAKI HASHIURA[†2]
ATSUO HAZEYAMA[†3]　SEIICHI KOMIYA[†4]

In software design exercises, teachers are unable to get information regarding learners' artifact during the creation process for example compilation errors or execution results in programming exercises. Therefore, it is difficult for teachers to grasp learners' progress (progression or stagnation, how are the artifact and right answer similar). In this study, we propose a method to visualize the learners' progress by visualizing transition of similarity of the learners' artifact and the right answer during the process of class diagram creation exercises. It contributes for teachers to grasp the learners' progress during the exercise easily. We believe that it is useful to conduct effective teaching in the exercises.

1. はじめに

ソフトウェア技術教育において，設計能力の育成は重要な課題である．現在，高等教育機関等において，設計能力育成のための演習型の授業が実施されている．

しかし，設計演習は，演習中の学習者の進捗状況（正解例にどの程度近づいたか，作業の進行 or 停滞）の把握が難しいという問題がある．例えばプログラミング演習においては，コンパイル結果，プログラムの実行結果，自動テストの結果などから演習中の成果物に関するデータを比較的容易に収集することができ，学習者の進捗状況の把握や支援が必要な学習者の抽出に活用が可能である[1]．しかし，設計演習においては，上で述べたような演習中の成果物に関するデータを収集できる環境が存在しない．

そこで本研究では，初学者向けの UML クラス図を作成する設計演習を対象とし，演習中の学習者の成果物と正解例の類似度の変遷を教授者に可視化することで，学習者の進捗状況の把握を支援する手法を提案する．

2. 提案手法

本節では，提案手法について詳しく述べる．本手法は，クラス図を作成する初学者向けの演習(クラス(クラス名，属性名)，関連（関連名，両端の多重度）のみをモデル化する）において学習者の進捗状況を教授者に可視化することを目的とする．

2.1 本手法における進捗状況

本手法では，学習者の成果物と教授者が作成した正解例の類似度を，演習における学習者の進捗と考える．また，作業が順調に進行している，あるいは停滞しているという学習者の状況を，作業時間または編集回数あたりの進捗の変化によって判断する．本手法では，これらを合わせて進捗状況と呼ぶ．

2.2 使用するデータ

筆者らは，クラス図エディタ KIfU を開発している[2]．KIfU は, 学習者がクラス図を編集するたびに, 編集の内容, 編集時刻，クラス図のスナップショット等を編集イベントとして記録する機能を持つ．本手法では，KIfU が収集するスナップショットと，教授者があらかじめ KIfU に登録した正解例のクラス図を入力とし，それらの類似度を計算する．また，学習者の状況の可視化には，編集時刻，編集イベント数をデータとして用いる．

2.3 正解例とスナップショットの類似度の評価方法

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(E_1,E_2)（要素 E_1, E_2 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[3]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

†1 東京学芸大学大学院
Graduate School of Education, Tokyo Gakugei University
†2 日本工業大学
Nippon Institute of Technology
†3 東京学芸大学
Tokyo Gakugei University
†4 国立情報学研究所
GRACE Center, National Institute of Informatics

クラス C_i∈S，クラス C_j∈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，不足クラスの数を 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∈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 NA は C_i の属性数を表す．

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

A, Sの全クラスに関する類似度 CS_{all} は次の式で計算する．式中の NC_S は S のクラス数を表す．

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

(2) 関連の比較

①比較対象関連の抽出

関連 R_i∈S と，関連 R_j∈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，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 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∈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 NR_S 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 CDS（Class Diagram Similarity）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．

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

2.4 進捗状況の可視化方法

可視化にあたっては，最新の進捗状況だけでなく，演習開始時からの進捗状況の変遷を可視化することで，より効果的な演習指導の支援が行えると考えている．

本手法における進捗状況の可視化のイメージを図 1 に示す．図 1 の左側のグラフは類似度の変化を表す．グラフの横軸は作業時間と編集回数で切り替えられるようにする．また，単位時間あたりまたは単位編集回数あたりの進捗の変化を計算し，変化量が閾値以下になった（停滞が検出された）箇所にはマーカーを表示する．これにより教授者は，作業時間または編集回数あたりの進捗状況の変遷およびその特徴（停滞しがち or 概ね順調等）を容易に把握することができる．図 1 右側のリストには，学習者の状況（現在停滞しているか否か）とクラス図の情報，および作業に関する情報を表示する．

各学習者について図 1 のような可視化を行い，学習者が編集を行うたびにこれらを更新することで，教授者は最新の進捗状況を随時閲覧することが可能になる．

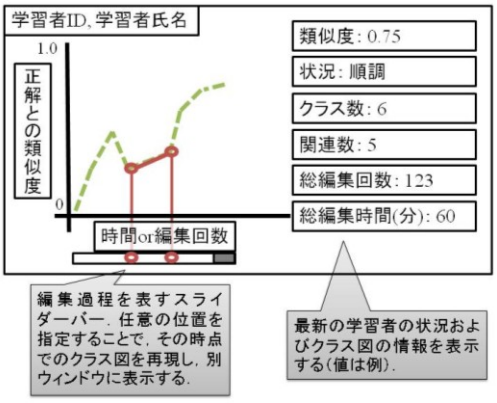

図 1 進捗状況の可視化のイメージ図

Figure 1 Image of visualization of a learner's progress.

3. まとめ

本稿では，学習者のクラス図の編集過程において収集されるスナップショットと，あらかじめ教授者が作成した正解例の類似度を編集が行われるたびに計算し可視化する手法を提案した．

今後は，KIfU に提案手法を実現する支援機能を実装する．その後，提案手法で用いる閾値 T_n および多重度の類似度を決定するため，人間の判断との比較実験を行い，適切な値を決定したい．

謝辞 本研究は科学研究費補助金基盤研究(C) 25750090, 26330394 の助成の下で行われている．

参考文献

1) 井垣宏, 他: プログラミング演習における進捗状況把握のためのコーディング過程可視化システム C3PV の提案, 情報処理学会論文誌, Vol.54, No.1, pp.330-339, (2013).

2) T. Tanaka et al.: Proposals of a Method Detectiong Learners' Difficult Points in Object Modeling Exercises and a Tool to Support the Method, Proceedings of SERA2014, pp.650-655, (2014).

3) Z. Xing and E. Stoulia: UMLDiff: an algorithm for object-oriented design differencing, Proceedings of ASE' 05, pp.54-65, (2005).