

ビジュアルコンテンツ比較による協調型プログラミング学習の

試み : 中小 IT 企業における社内リカレント教育を例に

ゲン タン ハ, 清光 英成, 大月 一弘
神戸大学大学院国際文化研究科

Trial of Collaborative Learning in Programming Education with the Programed Visual Contents Comparison Method

Thanh Ha Nguyen , Hidenari Kiyomitsu, Kazuhiro Ohtsuki
Graduate School of Intercultural Studies, Kobe University

In a prior study, we proposed the Programmed Visual Contents Comparison Method (PVCC) to assess this panoramic understanding of programming, and developed a testing system based on this method. We confirmed that the PVCC method worked well to find programming abilities related with PUP. In this paper, we discuss the effectiveness of the PVCC method for collaborative learning in computer programming education. We carried out an experiment in an IT company. Employees with diverse occupations and diverse programming backgrounds participated the collaborative learning. From this experiment, we found that collaborative learning using our method is useful for employees of all levels in an IT company.

キーワード: 研究報告, 書式, 執筆要領

1. はじめに

近年のソフトウェア開発においては、サンプルプログラムのコピーペースト・開発ツールの利用・既存のライブラリの利用など、プログラムの一部がブラックボックス化された状況でプログラムを作成することが多くなっており^{(1),(2)}, コードが書けないソフトウェア開発者も増加していることが指摘されている⁽³⁾. このような状況においては、プログラミング言語の文法を完全に記憶していることよりも、プログラムの文脈を理解する力、つまりプログラムの俯瞰的な理解する能力が大切であると考えている. ここでいう文脈とは、アルゴリズムや、それよりもう少し大きな見地からのプログラム構成力や設計力のことをいう.

著者らは、プログラムに対する俯瞰的な理解力を測定するためのビジュアルコンテンツ比較法を提案し、Web 上で個別にテストを受けることができる自律型のプログラミング能力測定システムを作成している

(4),(5). 同方法は、プログラムにより作成された2つのビジュアルコンテンツを眺め、どちらのコンテンツを作成する方が難しいか回答するものである. 著者等⁽⁶⁾は、ソフトウェア開発を業務とする IT 企業 (以下では単に IT 企業という) における社員の理解度測定に同システムが適応できるかを調べ、プログラマならびにそれ以外の社員に対しても、俯瞰的な理解力を測ることができることを示した. また、同能力測定テストの持つ、「画面を眺めてどちらが難しいかの単純な回答をするだけであるため、ペーパーテストのような重たい気持ちではなく、クイズをやっているような楽しい気持ちで気楽にテストができる」という特性を利用して、社員が気楽に参加できる協調学習にも利用可能性が高いことを示した.

中小規模の IT 企業では、ソフトウェア開発者はプログラミング経験者、つまり、即戦力を採用し、入社後は実際の業務の中でスキルアップを OJT のみ学ぶことが多いと思われる. また、営業など非開発系の従

事者はコードそのものを書くような学習をすることは稀であると考えられる、これに対して、中規模以上のIT企業では、プログラミング経験のない人もソフトウェア開発者として採用し、入社後に新入社員研修でプログラミング教育を行っている場合も多い。塩崎等⁽⁷⁾は、新入社員研修の事例を紹介しているが、OJTなどにおいては、学習の方法が悪いと、「表面上だけ理解した気になって、一応動くものは作成可能だが、内容をきっちりと理解していないプログラマが生まれる。」ことを指摘している。また、学生時代にプログラミング教育を受けた者の中にも、「文法は知っているが、プログラミングに対する基本的な考え方が身につけていない」場合があることも述べており、プログラミング言語に依存しないプログラミングの考え方を身に付けることの重要性を指摘している。

本研究では、IT企業における社内リカレント教育に焦点を絞り、ビジュアルコンテンツ比較法に基づく協調学習の可能性を探ることを目的とする。IT企業において、職種やプログラミング経験の異なる様々な社員が集まり、同方法に基づく協調学習を試行する。試行実験を観察し、プログラミング言語に依存しないプログラミングの考え方を身に付ける可能性について議論する。さらに、同法をより効果的に行うための、教材の充実や学習用システムの改善について議論する。

2. ビジュアルコンテンツ比較法

ビジュアルコンテンツ比較法は、図1のような、比較的似通ったプログラムによって作成された2つのビジュアルコンテンツ（静止画、アニメーション、マウス操作によって変化する画像）を閲覧し、プログラムで作成することを想像してもらい、どちらのプログラムがより難しいか、あるいは、同程度の難しさであるかの3択の判定してもらうものである。1つのビジュアルコンテンツ全体のプログラミング構成を想像することはかなりの知識を要し、初中級者にとっては理解が難しいが、違いを比べる場合、全体像はわからなくても違いの部分のみに焦点をあてることができる。

図1は、いずれもサンプルも小さな正方形を画面上がランダムに描くものである。左図（サンプル1）は、正方形が重なっていない。プログラミングの知識のない

人でもこの違いは気づくことができると思われる。プログラミングの知識がある程度ある人であれば、「いずれの図も乱数を用いて作画しているであろうが、左図は正方形が重ならないようにするための処理をしている」と判断できる。つまり、自分ではソースコードが書けない人であっても、プログラミングに対する俯瞰的な知識を持っていたら、正しく回答することができる。

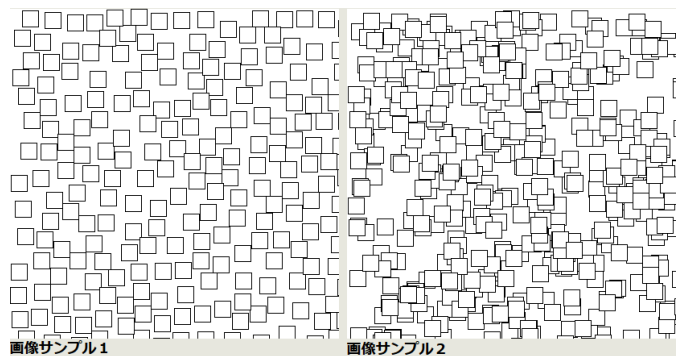


図1 ビジュアルコンテンツ比較法の教材

3. IT 中小企業の状況（事例紹介）

著者の一人は、ソフトウェア開発を行う中小IT企業を何社か経験してきた。ここでは、これらの会社の社員の学校教育時代のプログラミング経歴や入社後のプログラミングに関する技術習得について述べる。ソフトウェア開発部門以外の営業、管理に所属している人も対象にした。

(1) 教育機関での学習経験

ソフトウェア開発部門の多くは、情報系の学部や専門学校出身でプログラミング教育を受けている。情報系出身でない者は、オンライントレーニングコースなどで一定のプログラミング能力を習得していることが条件となる。営業系部門には、情報系出身と他の学部出身者が混ざっている。土木工学科や経済学部出身者などがいる。事務系部門には、ほとんど情報系出身者はいない。

(2) 入社後のスキル向上のための学習

入社後は、開発部門では、OJTが基本である。実際の業務を通しての既存のプログラムを参考にする、または同僚やチームリーダーからの指導を受ける、プロジェクトを遂行しながらプログラミング能力のスキルアップを行っていく場合が、ほとんどである。また、

オープンソースコミュニティは、開発者のほとんどが利用しているが、現在抱えている課題を解決するために検索する場合は、ほとんどで OJT の域を超えない。営業部門では、社内勉強会で学習することがあるが、専門用語や新しいトピックの解説を学ぶ程度である。営業部門は、顧客との対応のためにプログラミングの知識を増やしたいと考えている人が多いが、スキルアップのチャンスが少ない。中には、個人的に社外で提供されるオンラインコースのプログラミング教育を受講し、本で独習しながらプログラミングの学習をしている者もいた。

4. 実験の概要

中小 IT 企業に 7 名を対象会として、社内研修会という形式で開催した。参加者は、表 1 に示すように、上級のプログラマであるソフトウェア開発部門のマネージャーから、まったくプログラミング経験の無い者までさまざまであり、業務内容も異なる。

参加者は、画面に映し出された問題を閲覧し、全員で協議しながら回答を決定する。出題は例題を除いて 10 問である。研修会の時間は 60 分程度であった。

著者らは当該企業の代表取締役と共に議論を観察し、終了後にインタビューを行なった。

5. 結果と考察

(1) 初心者でも学べる

図 2 は、コンテンツ比較法の説明をするために示した例題である。このサンプル画面に対し、マネージャーが事務に回答を求めたところ、「右の方が難しい。だってたくさん四角をかかないといけないもの」と回答

した。他の参加者は、全員答えがわかっており、同じプログラムでパラメータを変更するだけでどちらも図も作成できることを説明していた。

本比較法は、ビジュアルコンテンツ間の差異に気づき、それがプログラム作成上でどのように異なるかを考えるものである。「同じ図形を複数書くことは、プログラムでは（繰り返し技法を使えば）図の個数は関係しない」ということは、初心者でも理解できたものとする。

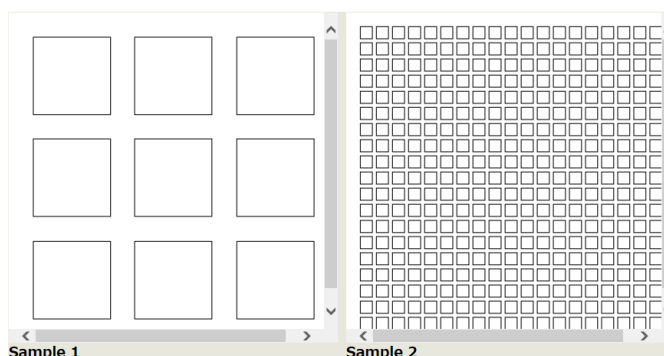


図 2 繰り返しの教材

(2) プログラミング技法を学ぶ

図 3 は、画面上でマウス操作をしてクリックを行うたびに、線が引かれていくコンテンツである。右図が常に左上の頂点からクリックした位置へ直線を引くのに対して、左図は、前回クリックした位置から今回のクリック位置に直線を引く。

本問題に対して、事務、営業とシスオペが、「差がないように思える」と言ったところ、マネージャーが「直線を始点が変化しているよね。」とヒントを与えた。シスオペは、「あっ、左は、前にクリックした場所を記憶しておかないといけないのですね。」と回答した。続い

表 1 実験参加者の概要

文中の略語	業種	プログラミング経験
マネージャ	ソフトウェア部門マネージャー	プログラミング経験20年以上
プログラマ1	プログラマ	プログラミング経験10年程度
プログラマ2	プログラマ	プログラミング経験は学生時代を含めて数年
上級シスオペ	上級システムオペレータ	若いころはプログラマをしていたが20年ほどプログラミング業務は行っていない。
シスオペ	システムオペレータ	プログラミングは学生時代に少し習った程度
営業	営業担当	数行程度のソースを書くことはあるが、プログラミングを習ったことはない
事務	事務担当	プログラミング経験なし、ワープロ、エクセルが使える程度

て、マネージャーがプログラマ2に対して、「どうやって記憶させる？」と質問したところ、「クリック位置を配列かリストに記憶されるのですか？もっといい方法あるのですよね。」と回答した。「配列やリストもある

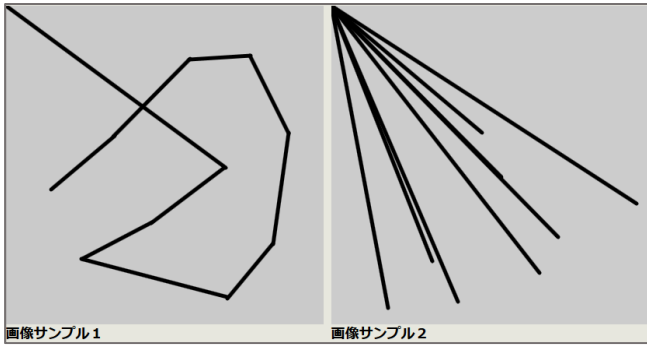


図3 マウス操作による直線の描画の教材

が、この場合は、前のデータという変数を作っておいて、そのデータを毎回書き換えていくのが定番である」とプログラマ1が答え、プログラマ2は、そのようなソースコードの記述方法については知っていて、プログラマ1の話により思い出したようであった。事務、営業は、プログラムの書き方までは、よくわからないようであったが、プログラムを作成する際の難しさに違いがあることは理解したようであった。このように、同じ題材に対して、学習者の学習レベルには差が出るが、技法について学習することができる。また、プログラミングの中上級者であれば、ソースの書き方についても、議論できることがわかった。

(3) レベルに応じて学習できる

図3のビジュアルコンテンツは、円を描いただけの単純な静止画である。この課題に対する「難しさ」の回答が2つに割れた。マネージャー・上級シスオペ・事務は、右図が難しいと回答し、他の4人はどちらも同じと回答した。それぞれが、自分の判断理由について、次のように回答した。

事務:「図を書くとき、重なる部分の線を書かないようにしないといけないので難しそう」

営業:「作画ツールで書く場合、どちらも「円」の図形をコピーするだけなので同じ作業でできるので、プログラムで書いても同じかと想像した。」

シスオペ:「ソースを書くための文法はちゃんとわからないけど、図1の時と同じ繰り返しの問題ですよ

ね。」

プログラマ2:「円を書く関数のパラメータを変えるだけで書けます。」(プログラマ1も同意)

このように、プログラムに対する知識が異なる場合でも、それぞれの知識・経験をもとに判断をし、一緒に議論ができ、各自のレベルに応じて学習できることが確認できた。

(4) 関数やライブラリの裏側を考える

前述の図3においては、「関数を利用する場合は、関数を記述するプログラムの複雑さも考慮してください」という前提を伝えていた。上級シスオペは、「そうか、プログラマ2さんは、同じ関数のパラメータが違うだけと考えるのですね。私ら昔の人間は、(隠れた部分の線を消す)陰線処理というテクニックを学んだものです。」と述べた。マネージャーは、「これは、技術レベルというより世代を調べる問題だね」と笑って言った。続けて、「事務さんの言うように、重なって隠れる部分に線を書かないということは難しいことです。人間は手で書く場合は円弧の始点を目で確認しながら書くことができるが、プログラムの場合だとそれを数値的に表現する必要がある。」と説明していた。プログラマ等は、同じ関数でもパラメータが違くと使用する内部プログラムが変わることを理解したようであった。上級シスオペは、「今のプログラムは、ライブラリやAPIが豊富なのでブラックボックス部分が多いですね。特に、ビジュアル表現はライブラリがないと大変だ。」と述べていた。この問題については、どちらの回答が正解かは判断しにくいという結論に至った。

このように、ライブラリや関数の裏側に隠れているプログラムについても理解を深めることができる。

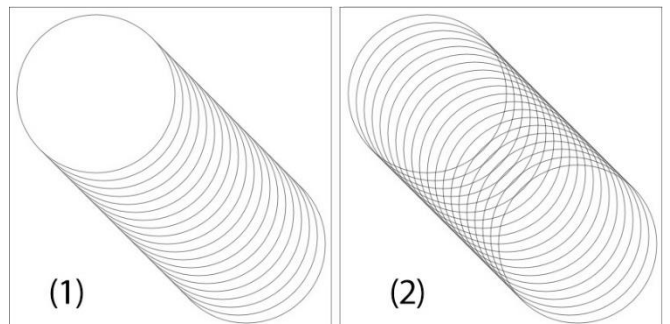


図4 図形の重ね書き(陰線処理)の教材

(5) 上級者も学習できる？

図5は、マウス操作を伴うコンテンツである。両図とも中央にある円は固定されており移動しない。左図は、画面上でマウスを移動していき、マウスが円内に入ると、図の黒と白の部分が反転する。右図では、マウスの位置を中心とした円がもう一つ描かれており、マウスの移動とともに円（例では、右側の円）が移動し、2つの円が重なると色が反転する。

このコンテンツに対して、最初は、全員が右図の方が難しいとの判断であった。右図は重なりを判定しなければならぬのでプログラムが難しくなると判定を下そうとしたとき、上級シスオペが、「あれっ、右の図の場合、マウスの位置が中央の円の半径の2倍以内にはいれば、いいのですよね。だったら左図の判定基準の半径を2倍にするだけでいい。」と発言した。マネージャーをはじめ全員がそれに納得した。マネージャーは「普段はデータ処理関係のプログラムばかり書いているので、気づかなかったわ。久々の幾何学の勉強だな」と答えていた。このように、本方法で出される問題は、単純にプログラムを書く技術だけでなく、問題に対する着眼点を見つけるといった、文法・アルゴリズム以外の要素を含む問題も含まれており、上級者でも「頭を使う」ことができると思える。

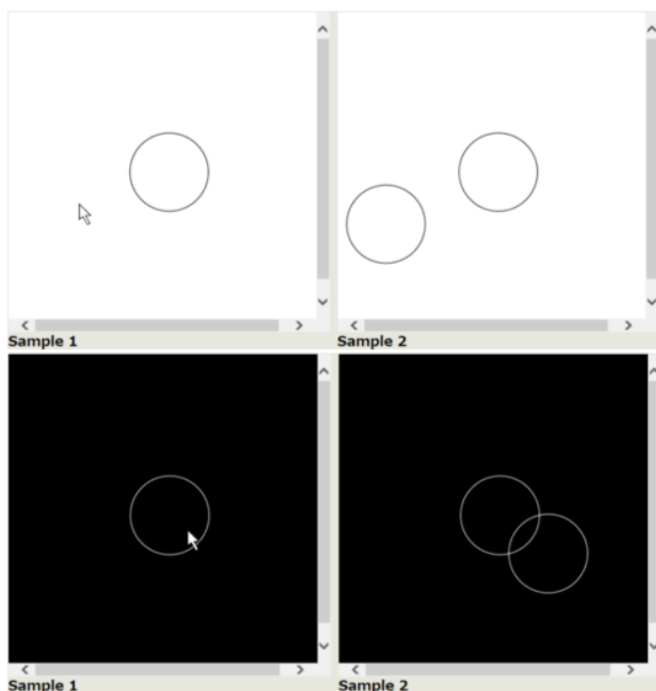


図5 条件によって色が反転する教材

(6) 学習成果

実験終了後に、参加者に今回の研修がプログラミングのレベルアップに役立ちそうかをインタビューした。事務は、高度な問題の際には、他者の議論を聞いているだけのことが多かったが、「今まで、皆さん同士の会話を横から聞いていても、難しくてさっぱり意味が分からなかったが、今日は、皆さんの議論の内容がわかりやすかった。話を聞いていて、どんなところに目をつけているのか少しわかった気がする。」と答えていた。コンピュータ技術者間の会話は専門用語を多用するため、素人からみれば、何を話しているのかさっぱりわからない、ということはよく聞く話である。今回の議論は、コンテンツを観ながらの会話であるため視覚的に違いをできること、細かな文法を議論するのではないため比較的専門用語が少ないこと、などがあり、技術用語をよく知らない初心者であっても、議論の内容が理解しやすいのではないかと考える。

また、他の者も「上級技術者の説明や議論を聞くことで、プログラミングに関する知識を深めた（シスオペ）」「私は10年以上プログラムコードを書いています。ソースコードを書くことを想像することは楽しかった。自分の知識レベルを確認できて嬉しかった（上級シスオペ）」など全員が勉強になったとの感想であった。

このように、本方式を用いた学習は、それぞれのレベルに応じてプログラムの考え方を俯瞰的に理解することに役立つと考える。

また、マネージャーは、「私にとっては、議論を通して他の人々のおおよその能力を把握できたことが一番役に立った。今までは、なかなか個々の人々の能力を知る機会がなかった。彼らと共同作業としたし、指示・指導をするときに役に立つ気がする」と答えていた。社内における業務を円滑に行う上でも本手法は役立つ可能性がある。

(7) 楽しく学ぶ

今回の協調学習においても、楽しみながら学習できたようであり、以下のような発言があった。

プログラマ1： 「プログラムコードを見ると通常ストレスを感じますが、このテストは2つのビジュアルコンテンツを比較するだけ良いので、ストレスを感じることはありませんでした」

プログラマ2「このテストはクイズの答え探しをして
いるような感じで、間違っても大丈夫と思えました」

営業:「一緒に議論に参加している感じがして、楽しか
った」

マネージャー:「プログラミングのスキルについて、こ
れほど多くの人々と楽しい方法で議論したことが
ありません」

このように、職場全体で一緒に楽しみながら気楽に
学習できることがわかった。

これらのことより、本学習方法は、社内におけるリ
カレント学習の一方法として十分に効果を発揮できる
ように思える。研修会を観察していた代表取締役から
は、「年に1,2度、このような研修会をしてみたい。
教材の種類と数の準備が必要になりますかねです。」と
の感想を得た。

6. 学習用システムの改善にむけて

今回の実験においては、教材の表示に、著達が開発
した Web ベースの能力測定システムを使用し、出題画
面をプロジェクタに投影した。同システムは、個々人
が自律的にテストを受験し、受験後に正解や解説が提
示されるようになっている。ここでは、同システムを
協調学習用のシステムとして使用する場合は、システ
ムの改善の方向性について述べる。

(1) 正解や解説の表示

この機能は、現在のシステムにすでに、準備されて
いる。今回の実験においても、全員で協議して正解を
決定した後に、システム側に準備された回答や解説を
みて確認を行っていた。教師役となる中上級プログラ
マがない場合に備えて、解説内容を充実させること
が重要かと思える。

(2) 類似問題の準備

前述の図5の問題の場合、左図も右図もプログラ
ムの難しさはそれほどなかった。しかし、描いていた「円」
が、別の図形の場合は、重なるの判定方法が今回の場
合とは異なり難くなる。今回の実験中には、図形が
変わった場合の判定法の違いについて議論が進展する
ことを期待したが、議論は進まなかった。新しい論点
に気づいてもらい議論の幅を広げる、議論を深める、
議論した内容を再確認する、といった学習ことを支援

するために、類似問題をいくつか準備しておくことが
重要であると考える。

(3) 正解の判断が難しい問題の利用

本システムは、もともと能力判定用の試験(テスト)
用に開発していたため、プログラミング教育に携わる
大学教員やプログラミング能力の高い大学院生に出題
する問題に関してコメントしてもらい、正解がはつき
りしない問題は出題の候補から省いていた。例えば、
「この2つのコンテンツは、2重ループで作成できる
か1重のループで作成できるかの違いであると思える
が、文法的には、繰り返しを使っているだけなので、
図形を描くことのプログラムの大変さからみたら軽微な
差であり、難しいという回答を差がないという回答の
どちらが正解であるか判断できない。」とコメントされ
た問題もあった。「初級者にとっては、差があると判断
した方がいいが、中級者以上であれば、差にはならな
い」、「出題者が意図する違いの部分以外にも違いがあ
る。」とコメントされた問題もあった。例えば、図1に
示した問題は、左図は重ならないような処理をしてい
るが、右図は陰線処理をしていると考えられる、(テス
ト問題として使用する場合は、陰線処理をしない作図
に変更した。)

これらの出題から外した問題についても、協調学習
の場合であれば、利用することができると考えている。
例えば、2つのコンテンツ間に複数の違いがある問題
は、「2つのコンテンツ作成における違いがいくつある
かを考える」といったテーマで議論を行う教材として
利用できる。特に、中級者レベルの学習には、このよ
うな問題の方が学習に有効かもしれない。これらの問
題を採用することで、学習に使用できる教材を増加さ
せることができる。

7. おわりに

本研究では、ビジュアルコンテンツ比較法を用いた
協調学習の効果について検討した。IT企業において、
職種やプログラミング経験の異なる様々な社員が集ま
り社内研修を試行した。試行実験の結果、本方式を用
いた協調学習がすべての社員にとって、それぞれのレ
ベルに応じた学習効果をあげることができるがわかっ

た.

謝辞

本研究を遂行するにあたり、快く本研究の社員研修実験を実施していただいた株式会社アイクラフトの山本裕計代表取締役ならびに社員の皆様に感謝いたします。

本研究の一部は科研費(19K03030)「コピペ時代のプログラミング学習方法の開発」の支援による。

参考文献

- (1) M. Burnett, Margaret and B. Myers, "Future of End-user Software Engineering: Beyond the Silos", Proceedings of the Future of Software Engineering Conference (FOSE 2014), pp. 201-211, (2014)
- (2) J. Brandt, P. J. Guo, J. Lewenstein, and S. R. Klemmer, "Opportunistic Programming: How Rapid Ideation and Prototyping Occur in Practice", Proceedings of the 4th international workshop on End-user software engineering (WEUSE '08), pp.1-5, (2008)
- (3) T. D. LaToza, B. Myers, "Hard-to-answer questions about code", Proceeding of PLATEAU '10, Article No.8, pp.1-6, (2010)
- (4) Martinez Calderon, D., Man, K., Kiyomitsu, H., Ohtsuki, K., Miyamoto, Y. and Sun, Y., "An evaluation method for panoramic understanding of programming by comparison with visual examples", IEEE Frontiers in Education Conference (FIE 2015), pp.1-8, (2015)
- (5) Martinez Calderon, D., Man, K., Kiyomitsu, H., Ohtsuki, K., Miyamoto, Y., Sun, Y. and Hirabayashi, M., "Measurement range increment in a method for evaluating Panoramic Understanding of Programming", IEEE Frontiers in Education Conference (FIE 2015), pp.1-8, (2016)
- (6) 清光 英成, ディック マルチネス, 孫 一, 大月 一弘, "ビジュアルコンテンツ比較法の法人利用の可能性とアプローチ", 第11回データ工学と情報マネジメントに関するフォーラム論文集, E2-5, pp.1-7, (2019)
- (7) 塩崎正人, 敦賀誠一, 中川宙, 古谷芳康, 吉田典弘, "企業の新入社員研修におけるプログラミング教育の実践例", 情報処理学会研究報告, Vol.2020-CE-153 No.8, pp.1-9, (2020)