Vol.2026-HCI-220 No.38
Vol.2026-UBI-92 No.38
2026/11/27

プログラミング課題の解答における手書きスケッチを用いた初学者の思考過程の分析

江森 柊哉[1]　関口 祐豊[1]　中村 聡史[1]

概要：プログラミング初学者が提示されたプログラミング課題に取り組む際には，課題で求められる要素や動作を整理する必要がある．その手段の一つとして，コーディング前にスケッチを用いて思考を整理することがあげられる．しかし，初学者にとって，何をスケッチに表現すべきか，描いた内容が課題要件を十分に捉えているかを判断することは難しい．そのため，初学者が作成したスケッチにどのような特徴が現れるのか，適切なスケッチとそうではないスケッチとの間でどのような違いがあるのかを把握する必要がある．そこで本研究では，プログラミング学習者 127 名が 2 課題に対して作成した手書きスケッチを対象に，課題解決に必要な主要な処理とその具体的な実現方法の記述，および処理のまとまりや処理間の関係を分析した．その結果，高い評価を得るスケッチでは，主要な処理と具体的な実現方法の充足率が高く，処理結果の利用関係を明示する割合に有意な差が認められた．

1. はじめに

大学などのプログラミングにまつわる初年次教育では，学習者にプログラムを作成する課題を提示し，その課題を解決する過程でプログラミングの知識や技能を習得してもらうことが多い．学習者はこれらの課題の解決にあたり，課題文から必要な要素や処理を抽出したり，それらの要素や処理の関係や実行順序を整理したりすることで，具体的な解法へと落とし込んでいくことになる．しかし，プログラム全体の構造を統合的に理解・把握する必要があるため，そうした抽出や整理は初学者にとって認知的な負荷が大きく，容易ではないという指摘がある [1]．

このような負荷を軽減する手段の一つとして，外的表現を適切に用いる方法が活用されている [2]．また，外的表現としてスケッチを用いることが，記憶や情報処理を補助することも示唆されている [3]．そのため，プログラミング課題に取り組むうえでも，適切なスケッチを用いることがその課題解決に役立つ可能性がある．しかし，初学者にとって課題をスケッチとして適切に表現すること自体にも難しさがあるため，まず初学者に対し，適切なスケッチを作成できるようにするためにどのような支援が必要かを明らかにする必要がある．その支援を実現するには，初学者が作成したスケッチにどのような特徴が現れるのか，適切なスケッチとそうではないスケッチとの間でどのような違いがあるのかを把握する必要がある．

そこで本研究では，様々なレベルのプログラミング学習者がコーディングに先立って作成した自由形式の手書きスケッチとその評価をもとにし，そのスケッチが課題解決につながるかどうかと，課題解決につながるスケッチまたは課題解決につながらないスケッチに描かれた内容を分析することで，そ解決可能性を分析する．

具体的には，以下の 3 つの仮説のもと分析に取り組む．

H1：課題解決に向け適切なスケッチにはそうでないスケッチに比べ，課題における必要な主要処理と，その具体的な実現方法が多く現れる．

H2：課題解決に向け適切なスケッチにはそうでないスケッチに比べ，処理のまとまりや処理間の関係を示す表現が多く現れる．

H3：課題解決に向け適切なスケッチにはそうでないスケッチに比べ，同一参加者内において異なる課題での項目の一貫性がみられる．

これらの分析を通して，課題解決において適切なスケッチとそうではないスケッチとの間で現れる要素や項目にどのような違いがあるのかを把握し，スケッチを用いた問題解決支援を検討するための基礎的な知見を得る．

2. 関連研究

2.1 プログラミング初学者の解決方針構成支援

プログラミング初学者の問題解決を支援するため，Corno ら [4] は，自然言語で記述されたプログラミング課題に対

[1] 明治大学
Meiji University

© 2026 Information Processing Society of Japan

して，問題の理解，部分問題への分解，各部分問題に対する実装方法の検討を支援する TextCode を提案している．また，Choi ら [5] は AlgoSolve を提案し，学習者が部分目標に関する説明を選択・作成し，他の学習者の説明と比較できるようにすることで，解法の構造に対する理解と解決計画の構成を支援している．

これらの研究は，初学者が課題を分解し，必要な処理や部分目標を明示的に整理することを支援するものである．しかし，あらかじめ情報を整理するための枠組みを与える方法では，学習者の問題理解や解決方針の構成が，提示された枠組みに沿って促される可能性がある．それに対し本研究で対象とする自由形式のスケッチは，表現内容やその粒度，配置，処理同士の関係を学習者自身が選別できるため，事前に設定された枠組みの影響を抑えながら，学習者自身の解決方針を外在化できる可能性がある．

2.2 スケッチによる思考の外在化

ソフトウェア開発において，開発者が設計上の判断や考えを一時的な図として外在化することや，ホワイトボード上のスケッチを用いて設計内容の確認や推論を行うことが報告されている [6,7]．このように，スケッチは単に考えを記録するだけでなく，考えている内容を外部に置き，それらを見返しながら整理するための外的表現として機能する．

また，スケッチには，考えが十分に具体化されていない段階からでも表現を始められるという特徴がある．Self [8] は，設計活動におけるスケッチの曖昧さが，設計上の考えを探索的に検討する上で重要な役割をもつことを整理している．さらに，Tversky [3] は，スケッチが記憶や情報処理を補助するとともに，描き手が重要と判断した情報を選択的に表現するため，対象に対する捉え方が表現に反映されることを指摘している．つまり，スケッチは，完成した解決方針だけでなく，描き手がどの情報に着目し，それらをどの程度具体化し，どのように整理しているかも表す可能性がある．

これらの知見から，スケッチは，考えを外部に記録して整理・検討するための手段であり，描き手が選択した情報やその関係付けを調べる手がかりにもなると考えられる．本研究では，初学者のスケッチに明示された内容に着目し，処理の具体性と整理の特徴を捉える．

2.3 プログラミングにおけるスケッチおよび解決計画

プログラミング学習において，学習者が作成したスケッチや解決計画を分析し，問題解決の特徴を明らかにする研究が行われている．Cunningham ら [9] は，プログラミング初学者がコードの読解，修正，並べ替え，作成などの課題に取り組む際に記述したスケッチを収集し，記述方法を分類するとともに，課題の種類や正答との関係を分析している．この研究では，課題の種類によってスケッチの利用方法や正答との関係が異なることが示されている．また，Kwon [10] は，初学者がコーディング前に作成した解決計画を分析し，変数の誤用，冗長な処理，問題解決戦略の不足など，解決計画に表れる問題点を明らかにしている．これらの研究では，スケッチや解決計画の記述方法，および問題解決における誤りや困難に着目している．

一方，学習者が作成する図の表現内容や構造に着目した研究もある．Gonzalez ら [11] は，プログラミング初学者が基本的なプログラミング概念について描いた図を分析し，コードやプログラミング概念の視覚的な表現方法が学習者によって異なることを報告している．また，Vadaparty ら [12] は，初学者がプログラムの問題分解について作成した図を分析し，関数の階層関係や実行順序などの表現方法を分類するとともに，表記方法，抽象化，処理順序などに関する問題点を明らかにしている．これらの研究では，学習者がプログラミングに関する情報をどのように視覚的に表現し，構造化するかが検討されている．

以上のように，先行研究では，スケッチの記述方法と課題成績との関係，解決計画に表れる問題点，プログラミング概念の視覚的表現，問題分解における構造的な表現などが分析されてきた．しかし，コーディングに先立って作成された自由形式のスケッチにおいて，課題解決に必要な処理がどの程度表出され，それらがどのような構造で整理されているかを併せて捉えることは，十分に検討されていない．そこで本研究では，初学者が複数のプログラミング課題に対して作成した手書きスケッチを対象に，課題解決に必要な処理の表出状況と，処理のまとまり，実行順序，処理間の関係などの構造的な特徴を分析する．

3. スケッチ分析のための調査

3.1 調査概要

本調査では，プログラミング初学者が描くスケッチにおいて適切なものとそうではないものとの間に現れる要素や項目にどのような違いがあるのかを把握することを目的とし，プログラミングに関するレベルの異なる学習者がコーディングに先立って作成する手書きスケッチを対象として，課題解決に必要な処理と具体的な実現方法の記述状況，および必要な処理の整理の特徴について調査した．調査は，大学初年次相当のプログラミング科目（CS1）を受講する学生 130 名を対象として実施した．

調査はプログラミングの講義で行い，計 4 課題実施した．4 課題の中には，対象の授業回まででは未学習の内容が含まれるものがあったため，今回はそれらを除いた 2 つの課題についてデータ収集および分析を行った．参加者には，各課題のコーディングを開始する前に，紙とペンを用いて解決方針を自由にスケッチするよう指示した．スケッチの表現形式や記述内容，作成時間は指定しなかった．作成されたスケッチは，各講義の終了時に回収した．

© 2026 Information Processing Society of Japan

表 1 調査で使用した課題

課題	課題の概要	主な処理
課題 1	サイコロを 4 回振って 4 桁の整数を生成し，100 万回の試行から素数となる確率を求める．	乱数生成，整数の構成，素数判定，繰り返し処理，確率計算
課題 2	8 桁の数字「abababab」の a と b に 1 から 9 の数字を割り当て，約数の個数が最小となる組み合わせを求める．	組み合わせの列挙，整数の生成，約数の個数の計算，最小値の更新

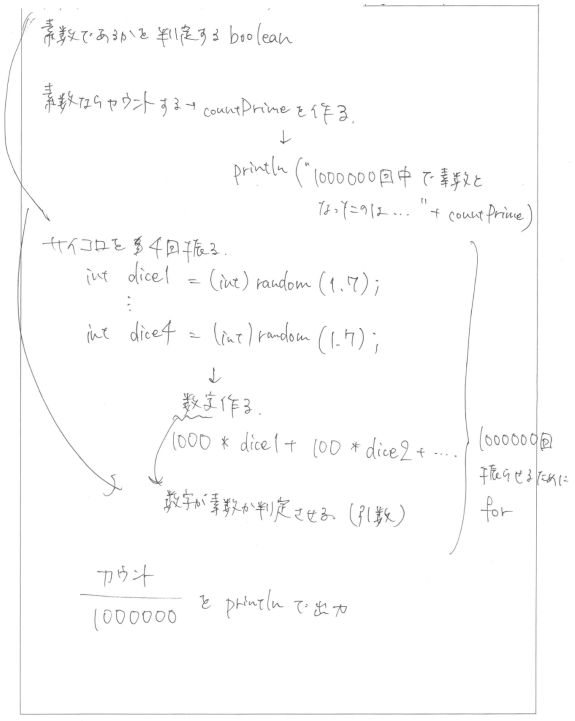

図 1 課題 1 のスケッチ例

なお，収集においては，学習者がデータとしての利用を望まない場合にはデータから除外することとその除外のための方法を連絡するとともに，スケッチの個人情報は ID 化することで匿名化の処理をし，分析対象のデータとして利用した．

3.2 利用した課題

調査で使用した 2 課題の概要と，各課題の解決に必要となる主な処理を表 1 に示す．課題は，条件分岐や反復といった複数の処理を用い，かつスケッチとして表現がしやすい要素を持つものとした．また，課題内容は「乱数を用いた反復試行」と「数値の組み合わせの探索」とし，処理の内容や構造が異なるものとした．

収集したスケッチの例として，課題 1 を解くために作成されたスケッチを図 1 に示す．

表 2 DicePrime 課題における要素項目の一例

番号	アノテーション項目
1	**1〜6 の乱数を 4 回生成する**
2	**4 つの値に 1000，100，10，1 を掛け，4 桁の整数を生成する**
2.1	4 桁の整数を保持する変数が存在する
3	**生成した 4 桁の整数を素数判定の対象とする**
4	**対象が素数か否かを判定する**
4.1	剰余演算（num%i など）の記述がある
4.2	ループの範囲を具体的に設定している
4.3	割り切れた場合に非素数とする記述がある
4.4	素数判定を関数化している
5	**100 万回の試行を行う**
6	**素数となった回数を数える**
6.1	回数を保持する変数（count など）が存在する
7	**素数となった割合を算出する**
7.1	count/100 万に該当する記述がある

3.3 スケッチのアノテーション項目

収集したスケッチに記述された内容を分析するため，課題ごとに異なる処理内容を対象とする「要素項目」と，全課題に共通するスケッチ上の整理の特徴を対象とする「スケッチ項目」の 2 種類のアノテーション項目を設定した．

要素項目は，各課題の解決に必要な処理がスケッチ上に表出されているかを確認するための項目であり，「主項目」と「補足項目」から構成される．主項目は，課題解決に必要となる主要な処理そのものを表す．一方，補足項目は，主項目に対応する処理の具体的な実現方法や条件に加え，変数の使用や関数化など，実装上の選択に関する情報を確認するための項目である．課題固有のアノテーション項目の例として，課題 1 に設定した項目を表 2 に示す．

一方，スケッチ項目は，個々の課題特有の処理内容ではなく，処理がスケッチ上でどのように整理されているかを確認するため，全課題に共通して設定した．具体的には，S1「処理のまとまり」，S2「実行順序」，S3「処理結果の利用関係」，S4「制御関係」の 4 項目とし，課題を解くうえで考慮する必要のある関係性として選定した．各項目の判定内容を表 3，具体例を図 2 に示す．

3.4 スケッチへのアノテーション手順

各スケッチについて，要素項目およびスケッチ項目に該当する表現の有無を判定し，該当する場合を 1，該当しない場合を 0 として記録した．判定対象には，文章による記述だけでなく，図，数式，記号，矢印，囲みなどによる表現も含めた．なお，アノテーションは，事前に複数回アノテーション作業を行い，著者間で議論して調整したうえで，実作業自体は筆頭著者 1 名で行った．

要素項目のうち主項目については対象となる処理を行うことが示されている場合に該当とし，具体的な実装方法まで記述されていることは求めなかった．例えば，課題 1 に

© 2026 Information Processing Society of Japan

表 3　全課題共通のスケッチ項目

番号	評価対象	判定内容
S1	処理のまとまり	同一の処理単位に属する記述が，視覚的にまとまって表現されているか
S2	実行順序	異なる処理単位間の実行順序が，視覚的または明示的に表現されているか
S3	処理結果の利用関係	ある処理の結果が別の処理で利用される関係が，明示的に表現されているか
S4	制御関係	条件・反復などによって，どの処理が実行されるかという関係が，明示的に表現されているか

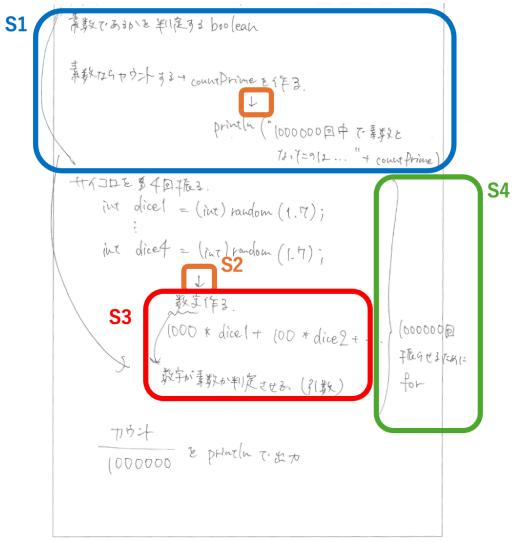

図 2　スケッチ項目の例

おける「素数か調べる」という記述は，具体的な判定の手順が示されていなくても，素数判定を行うことが示されているため，主項目 4 に該当すると判定した．

補足項目については，主項目に対応する処理をどのように実現するかについて，各項目で定めた具体的な内容が明示されている場合に該当とした．例えば，素数判定における剰余演算やループ範囲などがこれにあたる．つまり，主項目は「何を行うか」，補足項目は「それをどのように行うか」に対応する項目として扱った．

スケッチ項目については，スケッチ上の記述の配置，囲み，矢印，記号などを基に，処理のまとまりや処理間の関係が明示的に表現されているかを判定した．課題の正しい解法や記述内容から処理間の関係を推測できる場合であっても，スケッチ上にその関係を示す表現が確認できない場合には該当としなかった．

3.5　スケッチに基づく課題解決可能性の評価

本研究では，そのスケッチを行う時間統制や，成績との紐付けを行っていないため，収集したスケッチが，課題を解くための解決方針としてどの程度適切であるかを成績を用いて評価することができない．そのため，本研究の評価項目を把握していないプログラミングについて十分知識のある 2 名の評価者に依頼し，スケッチの評価を行ってもらった．

ここで評価者には，各参加者が作成したスケッチを 1 枚ずつ提示し，そのスケッチに記述された内容のみを参考にして，課題を解くことができると考えられる程度を 5 段階で評価してもらった．評価尺度には 5 段階のリッカート尺度を用い，「1：解けない」から「5：解ける」までの値を付与してもらった．なお，課題 1 および課題 2 では，それぞれ異なる 1 名の評価者が当該課題のすべてのスケッチを評価した．以降，本稿ではこの評価値をスケッチ評価と呼ぶ．

表 4　課題 1 と課題 2 における評価群の組み合わせ

	課題 2		
課題 1	低評価群	高評価群	合計
低評価群	**49**	18	67
高評価群	11	**5**	16
合計	60	23	83

4.　結果

4.1　分析対象

本章では，データを収集した 130 名のうち，2 つの課題の両方について分析に必要なデータが揃っている 127 名を分析対象とした．

要素項目については，課題ごとに設定された項目数が異なるため，参加者間および課題間の比較には単純な充足項目数ではなく，各課題で対象となる要素項目のうち，参加者のスケッチに表出していた項目の割合を用いる．以降，本稿ではこの割合を要素充足率と呼ぶ．

また，スケッチ評価とスケッチの特徴との関係を分析するため，5 段階で付与したスケッチ評価に基づき，評価が 1 または 2 のスケッチを低評価群，4 または 5 のスケッチを高評価群とした．評価が 3 のスケッチについては，高評価群と低評価群の比較から除外した．なお，群分けは参加者単位ではなく各課題で作成されたスケッチ単位で行うため，同一参加者であっても課題 1 と課題 2 で異なる群に分類される場合がある．

表??に，課題ごとの高評価群および低評価群の内訳を示す．課題 1 では低評価群が 86 件，高評価群が 21 件，課題 2 では低評価群が 70 件，高評価群が 28 件であった．

4.2　スケッチ評価と要素充足率の関係（H1）

高評価群と低評価群における要素充足率の違いを明らかにするため，主項目，補足項目，全要素の充足率について比較を行った．

図 3 に，高評価群および低評価群における主項目，補足項目，全要素の平均充足率を示す．主項目充足率の平均は，

© 2026 Information Processing Society of Japan

高評価群で 0.941，低評価群で 0.709 であった．補足項目充足率の平均は，高評価群で 0.746，低評価群で 0.342 であり，全要素充足率の平均は，高評価群で 0.813，低評価群で 0.492 であった．いずれの指標においても，高評価群の平均充足率が低評価群を上回った．特に補足項目では，両群の平均値の差が 0.404 と，3 指標の中で最も大きかった．

同一参加者が 2 課題の分析に含まれることを考慮し，参加者をクラスタとした一般化推定方程式（Generalized Estimating Equations: GEE）を用いて高評価群と低評価群を比較した．GEE は，同一参加者から得られた複数の観測間の相関を考慮しながら，群間の差を推定する手法である．また，課題による違いを考慮するため課題を説明変数に含め，主項目，補足項目，全要素の 3 指標について得られた p 値に対して Holm 法による多重比較補正を行った．その結果，主項目充足率では高評価群が低評価群より 0.236，補足項目充足率では 0.392，全要素充足率では 0.327 高く，すべての指標において有意な群間差が認められた（$p < .001$）．

次に，各群における要素充足率のばらつきを確認するため，図 4 に要素充足率の分布を示す．主項目充足率の標準偏差は，高評価群で 0.120，低評価群で 0.257 であり，低評価群においてより広い分布がみられた．補足項目充足率の標準偏差は，高評価群で 0.237，低評価群で 0.262 であり，両群とも比較的広い分布を示した．全要素充足率については，高評価群で 0.178，低評価群で 0.242 であり，主項目と同様に低評価群の方が広い分布を示した．

また，分布の広がりを群間で比較するため，各スケッチについて課題および群ごとの中央値からの絶対偏差を算出し，この絶対偏差を目的変数とした GEE による分析を行った．その結果，主項目では高評価群の平均絶対偏差が 0.060 であったのに対し，低評価群では 0.204 であり，低評価群の方が有意に分布の広がりが大きかった（$p < .001$）．全要素についても，高評価群の平均絶対偏差が 0.134，低評価群が 0.197 であり，低評価群の方が有意に分布の広がりが大きかった（$p < .001$）．

一方，補足項目では，高評価群の平均絶対偏差が 0.178，低評価群が 0.216 であり，低評価群の方が大きい値を示したものの，分布の広がりに有意な群間差は認められなかった（$p = .095$）．

4.3 スケッチ評価とスケッチ項目の関係（H2）

スケッチ評価の高低によって，スケッチに表出する構造的な特徴がどのように異なるかを明らかにするため，S1「処理のまとまり」，S2「実行順序」，S3「処理結果の利用関係」，S4「制御関係」の 4 項目について，高評価群と低評価群の表出傾向を比較した．

図 5 に，各スケッチ項目の表出率を示す．S1 の表出率は，高評価群で 0.735，低評価群で 0.551 であり，両群とも 4 項目の中で最も高い表出率を示した．S2 の表出率は，高評価群で 0.306，低評価群で 0.205，S3 は高評価群で 0.367，低評価群で 0.147，S4 は高評価群で 0.347，低評価群で 0.205 であった．いずれの項目についても，高評価群の表出率が低評価群を上回った．

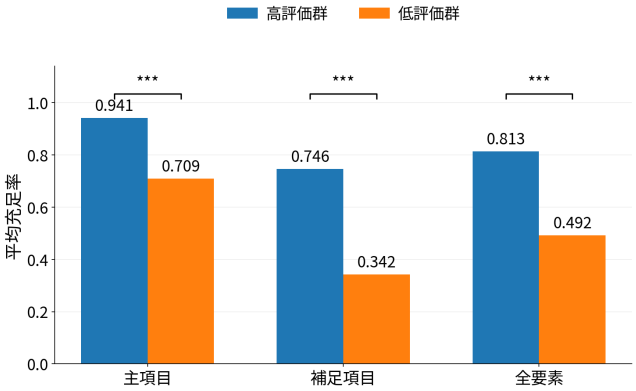

図 3 要素項目充足率

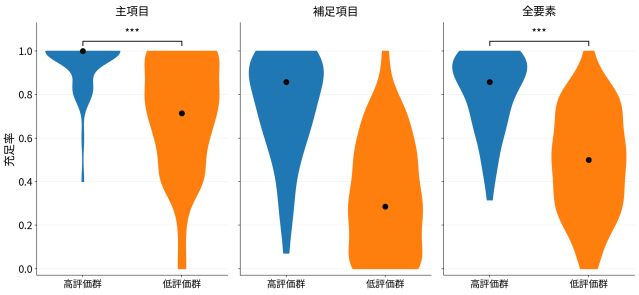

図 4 要素項目充足率の分布

また，S1 から S4 までの 4 項目について得られた p 値に対して Holm 法による多重比較補正を行った．その結果，S3「処理結果の利用関係」について高評価群の表出率が低評価群より有意に高かった（オッズ比 = 3.12，$p = .019$）．一方，S1，S2，S4 については，高評価群の表出率が低評価群を上回っていたものの，多重比較補正後には有意な群間差は認められなかった．

次に，スケッチ中に複数の特徴がどの程度表出しているかを確認するため，1 スケッチあたりに表出した S1 から S4 の項目数を比較した．各群における 1 スケッチあたりの平均特徴数は，高評価群では平均 1.76 項目（SD= 1.11）であったのに対し，低評価群では平均 1.11 項目（SD= 1.05）であった．特徴数を目的変数とし，同様に参加者をクラスタとした GEE によって群間比較を行った結果，高評価群では低評価群よりも 1 スケッチあたりの特徴数が有意に多かった（$p = .003$）．このことから，高評価群では特定のスケッチ項目のみが多く表出するだけでなく，1 つのスケッチ内に複数の特徴が表出する傾向がみられた．

4.4 課題間における表出傾向の一貫性（H3）

参加者ごとに，課題 1 と課題 2 における要素項目および

© 2026 Information Processing Society of Japan

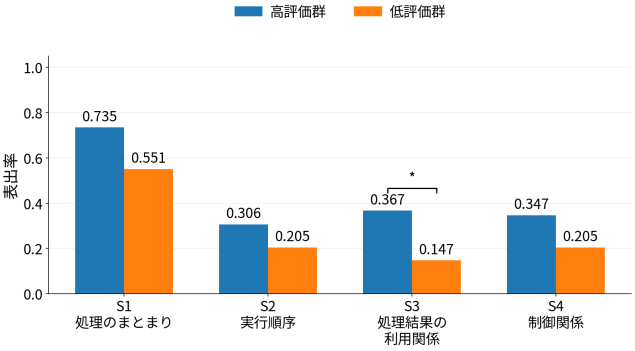

図 5　スケッチ項目ごとの充足率

表 5　課題 1 と課題 2 におけるスケッチ評価区分の組み合わせ

	課題 2			
課題 1	低評価	評価 3	高評価	合計
低評価	**49**	19	18	86
評価 3	10	5	5	20
高評価	11	5	**5**	21
合計	70	29	28	127

スケッチ項目の表出傾向がどの程度共通しているかを確認するため，2 課題間の一致について分析を行った．

本分析では，2 課題ともスケッチ評価が 4 または 5 であった参加者を高評価群，2 課題とも 1 または 2 であった参加者を低評価群とした．一方の課題のみが高評価または低評価であった参加者，および評価 3 を含む参加者は，本節における群間比較の対象から除外した．

表 5 に，課題 1 と課題 2 におけるスケッチ評価区分の組み合わせを示す．2 課題とも低評価であった参加者は 49 名，2 課題とも高評価であった参加者は 5 名であり，本節の比較対象は計 54 名であった．残る 73 名は，いずれかの課題に評価 3 を含むか，2 課題で評価区分が異なっていたため比較対象から除外した．想定よりも人数に偏りがあり，高評価群が 5 名に限られるため，本仮説については探索的な分析にとどめ，統計的な検証は行わない．

要素項目については，各参加者の課題 1 と課題 2 における充足率の差に基づき，2 課題間の一致度を算出した．具体的には，課題 1 の充足率を R_1，課題 2 の充足率を R_2 としたとき，課題間一致度 C を以下の式で定義した．

$$C = 1 - |R_1 - R_2| \tag{1}$$

一致度は 0 から 1 の値をとり，1 に近いほど 2 課題で同程度の充足率を示したことを表す．本分析では，主項目，補足項目，全要素についてそれぞれ一致度を算出した．

図 6 に，各群における要素項目の平均一致度を示す．主項目の平均一致度は，高評価群で 0.960，低評価群で 0.701 であった．補足項目では，高評価群が 0.857，低評価群が 0.735，全要素では，高評価群が 0.857，低評価群が 0.738 であった．いずれの指標においても，高評価群の方が高い一致度を示した．特に主項目では，両群の差が他の指標と

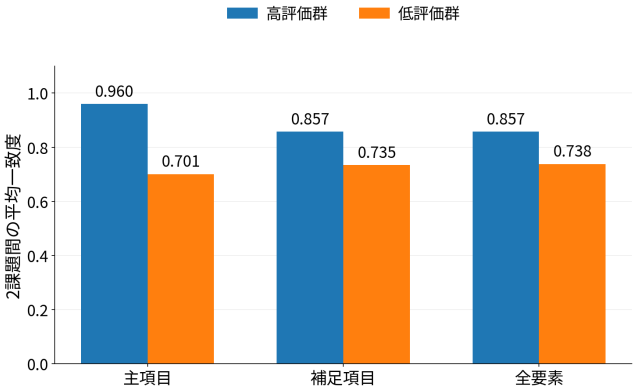

図 6　要素項目の一貫性

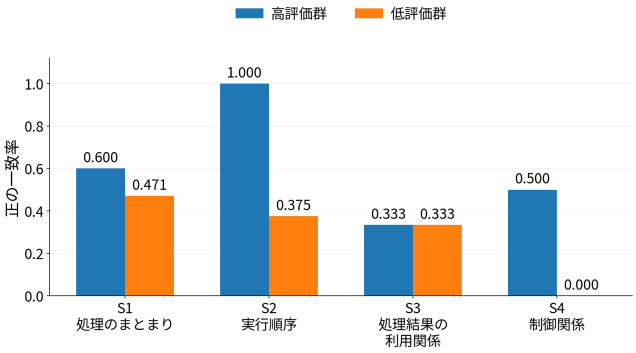

図 7　スケッチ項目の一貫性

比較して大きく，高評価群では 2 課題を通して主項目の充足率が近い傾向がみられた．

次に，S1 から S4 のスケッチ項目について，2 課題を通して同じ特徴が繰り返し表出しているかを確認した．ここでは，あるスケッチ項目が少なくとも一方の課題で表出した参加者を対象とし，そのうち課題 1 と課題 2 の両方で当該項目が表出した参加者の割合を「正の一致率」とした．すなわち，両課題で当該項目が非表出であった場合は一致として扱わず，正の一致率の算出対象から除外した．

図 7 に，S1 から S4 の正の一致率を示す．S1「処理のまとまり」の正の一致率は，高評価群で 0.600，低評価群で 0.471 であった．S2「実行順序」では，高評価群で 1.000，低評価群で 0.375 であった．S3「処理結果の利用関係」では，両群とも 0.333 であった．S4「制御関係」では，高評価群で 0.500 であったのに対し，低評価群では 0.000 であった．

以上より，高評価群では，要素項目について 2 課題間で比較的近い充足率を示すとともに，一部のスケッチ項目についても複数の課題で繰り返し表出する傾向がみられた．なお，先述の通り高評価群は 5 名と少ないため，これらの結果は群間差を示すものではなく，今後の研究において検証を行う一つの切り口を示唆するものにすぎない．

© 2026 Information Processing Society of Japan

5. 考察

5.1 主要な処理と具体的な実現方法の記述（H1）

4.2 節の結果から，高評価群では低評価群と比較して，主項目，補足項目，全要素のいずれについても要素充足率が高いことが示された．このことから，課題を解くために必要となる要素をスケッチ上に表出することは，解決方針として評価されるスケッチの特徴の一つであると考えられる．特に，主項目だけでなく補足項目についても高評価群と低評価群の差が大きかったことから，課題の中心となる処理だけでなく，それに付随する情報まで捉えて表出することが，解決方針を具体化する上で重要である可能性が示唆される．

一方，主項目と比較すると，補足項目の充足率は高評価群においても相対的に低かった．また，主項目では高評価群の分布が比較的狭かったのに対し，補足項目については高評価群においても一定のばらつきがみられ，分布の広がりについて低評価群との有意な差は認められなかった．この結果は，高く評価されたスケッチであっても，すべての補足的な情報が一様に表出されているわけではないことを示している．

これらの結果から，参加者が課題解決に必要な情報をすべて書き出すのではなく，自身の理解や解決方針に応じて，スケッチ上に表出する情報を選択している可能性が考えられる．例えば，課題を理解する上で重要な処理や関係については明示する一方で，自身にとって自明である情報や，解決方針を構成する上で必ずしも明示する必要がないと判断した情報については，あえて記述しない場合が考えられる．つまり，要素を十分に捉えてスケッチに表出することは重要であるものの，要素充足率が高いほど常に望ましいとは限らず，すべての情報を書き出すことそのものを適切なスケッチの条件とすることには注意が必要である．

5.2 スケッチにおける処理の整理（H2）

4.3 節の結果から，S3「処理結果の利用関係」では，高評価群の表出率が 0.367 であったのに対し，低評価群では 0.147 であり，4 項目の中で最も大きな差がみられた．また，群間比較においても S3 のみ有意な差が認められた．S4「制御関係」についても，高評価群で 0.347，低評価群で 0.205 と，高評価群でより多く表出する傾向がみられた．S3 および S4 は，個々の処理そのものではなく，ある処理の結果が別の処理でどのように利用されるか，あるいは処理がどのような条件や繰り返しによって制御されるかといった，処理間の関係を表す項目である．つまり，適切な解決方針を構成するためには，必要な処理を断片的に表出するだけでなく，それらの処理が互いにどのように関係しながら実行されるのかを捉え，スケッチ上に表出することが重要である可能性が示唆された．

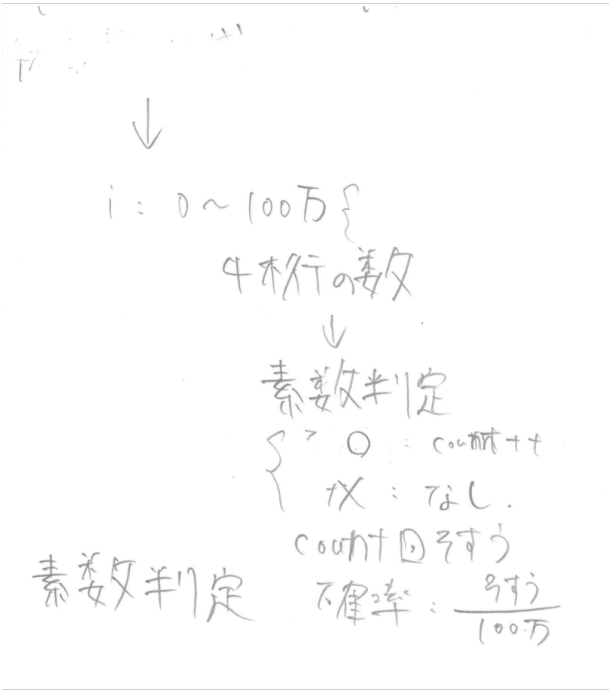

図 8 S2「実行順序」を書いているが低評価群に属すスケッチ

一方，S2「実行順序」に関して，高評価群では S1 から S4 の中で最も低い表出率であるのに対し，低評価群においては S4 と同程度で，S3 よりも高い表出率であった．このことから，処理を順番に並べて表現すること自体は，適切なスケッチに固有の特徴ではない可能性が考えられる．

この背景には，初学者が課題文に記述された処理の流れを順に書き出すことで，解決方針を表現したと捉えてしまっている可能性が考えられる．つまり，課題文に示された情報を順序に沿って並べることは比較的行いやすい一方で，それぞれの処理がどのようにつながるのか，ある処理の結果が後続の処理でどのように利用されるのかといった関係まで検討することは，より高度な課題理解を必要とすると考えられる．本研究においても，図 8 のように，実行順序は記載されているが課題理解には繋がりにくいスケッチがいくつかみられた．そのため，実行順序のみが表出されたスケッチでは，一見すると情報の整理が行われているように感じるが，解決方針として必要な処理間の関係が十分に構成されていない場合があると考えられる．

5.3 同一参加者の課題間にみられる共通点と相違点（H3）

4.4 節の結果から，高評価群では要素項目について 2 課題間で比較的近い充足率を示した一方で，スケッチ項目の正の一致率には項目によって違いがみられた．このことから，適切なスケッチを作成する参加者であっても，課題に含まれる情報を捉えるという点では比較的一貫している一方，それをどのように表現するかについては，必ずしも同じ方法を用いているわけではない可能性が考えられる．

© 2026 Information Processing Society of Japan

特に，S2「実行順序」は高評価群で高い一致率を示したのに対し，S3「処理結果の利用関係」などでは一致率が相対的に低かった．この違いは，参加者が複数のスケッチ方法を一様に用いているのではなく，比較的安定して用いる表現方法と，課題の構造や必要性に応じて選択的に用いる表現方法を使い分けている可能性を示している．

つまり，自身の中でのスケッチ方法を確立したうえで，課題に応じて必要な表現を選択することが重要であると考えられる．

5.4 スケッチを用いたプログラミング支援のための示唆

5.1 節〜5.4 節より，初学者のスケッチ作成を支援する際には，課題に含まれる要素をできるだけ多く書き出すことを促すだけでは十分ではないと考えられる．主要な処理やその実現方法を捉えることは重要である一方で，高く評価されたスケッチにおいてもすべての情報が一様に表出されているわけではなかった．そのため，支援においては記述量や要素充足率そのものを高めるのではなく，課題を解くうえでどの情報を明示する必要があるのかを判断できるようにすることが重要である．

また，スケッチ項目の分析からは，個々の処理を記述するだけでなく，S3「処理結果の利用関係」や S4「制御関係」のように，複数の処理がどのようにつながっているのかを捉えることが，適切な解決方針の構成に関係する可能性が示された．つまり，初学者に対して単に処理を順番に書かせるのではなく，「この処理の結果は次にどこで使われるのか」「この処理はどの条件によって実行されるのか」といった，処理間の関係に着目させる支援が有効である可能性がある．さらに，適切な表現方法は課題によって異なる可能性があるため，特定のスケッチ形式を一律に提示するのではなく，複数の表現方法を選択肢として与え，課題の構造に応じて必要な情報や関係を選択して表出できるようにすることが望ましい．

以上のことより，今後のスケッチ支援では，情報を網羅的に書き出させることよりも，課題に必要な情報を選択し，それらの関係を構造化して表現する過程を支援することが重要であると考えられる．

6. おわりに

本研究では，プログラミング学習者 127 名が 2 課題に対して作成した手書きスケッチを対象に，課題解決に必要な処理の記述と処理間の関係に着目して分析した．その結果，課題解決可能性が高く評価されたスケッチでは，主要な処理だけでなく，具体的な実現方法に関する記述も多いことが明らかとなった．また，スケッチ項目のうち，「処理結果の利用関係」について高評価群と低評価群の間に有意な差が認められた．これらの結果から，個々の処理の具体化と処理間の関係の整理を促す支援の可能性が示唆された．

今後は，今回明らかにしたスケッチの要素や必要要件を再整理し，初学者のスケッチを自動分析するとともに，そのスケッチをより良いものにするための支援手法を実現する．また，こうした支援がスケッチの記述内容や実際のプログラム作成に与える影響を検証する．

参考文献

[1] Lahtinen, E., Ala-Mutka, K. and Järvinen, H.-M.: A Study of the Difficulties of Novice Programmers, *Proceedings of the 10th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education*, ITiCSE '05, Association for Computing Machinery, p. 14–18 (2005).

[2] Scaife, M. and Rogers, Y.: External cognition: how do graphical representations work?, *International Journal of Human-Computer Studies*, Vol. 45, No. 2, pp. 185–213 (1996).

[3] Tversky, B.: What do sketches say about thinking, *2002 AAAI Spring Symposium, Sketch Understanding Workshop, Stanford University, AAAI Technical Report SS-02-08*, Vol. 148, p. 151 (2002).

[4] Corno, F., De Russis, L. and Pablo Sáenz, J.: TextCode: A Tool to Support Problem Solving Among Novice Programmers, *2021 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, pp. 1–5 (2021).

[5] Choi, K., Shin, H., Xia, M. and Kim, J.: AlgoSolve: Supporting Subgoal Learning in Algorithmic Problem-Solving with Learnersourced Microtasks, *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, CHI '22, Association for Computing Machinery (2022).

[6] Cherubini, M., Venolia, G., DeLine, R. and Ko, A. J.: Let's go to the whiteboard: how and why software developers use drawings, *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '07, Association for Computing Machinery, p. 557–566 (2007).

[7] Mangano, N., LaToza, T., Petre, M. and van der Hoek, A.: How Software Designers Interact with Sketches at the Whiteboard, *IEEE Transactions on Software Engineering*, Vol. 41, pp. 135–156 (2015).

[8] Self, J.: A Distributed Approach to Design Sketching, *DRS2024*, (online), DOI: 10.21606/drs.2024.205 (2024).

[9] Cunningham, K., Blanchard, S., Ericson, B. and Guzdial, M.: Using Tracing and Sketching to Solve Programming Problems: Replicating and Extending an Analysis of What Students Draw, *Proceedings of the 2017 ACM Conference on International Computing Education Research*, ICER '17, Association for Computing Machinery, p. 164–172 (2017).

[10] Kwon, K.: Novice Programmer's Misconception of Programming Reflected on Problem-solving Plans, *International Journal of Computer Science Education in Schools*, Vol. 1, pp. 14–24 (2017).

[11] Gonzalez, R., Klevgaard, T. and Raaen, K.: Students' Visualisations of Programming Concepts: An Exploratory Analysis, *Norsk IKT-konferanse for forskning og utdanning* (2024).

[12] Vadaparty, A., Hayatpur, D., Raj, A. G. S., Porter, L. and Zingaro, D.: Planning on Paper: Problem Decomposition with Diagrams in Introductory Computing (2026).

© 2026 Information Processing Society of Japan