JSiSE Research Report
vol.37,no.7(2023-3)

プログラミングの文章問題を対象とした定式化による解法の理解を促す学習支援システムの開発・評価

白髭　虹輝[*1], 松爲　泰生[*2], 東本　崇仁[*1]
*1 東京工芸大学工学部,　*2 東京工芸大学大学院工学研究科

Development and Evaluation of a Learning Support System That Encourage the Understanding of the Solution by Formularization for Programming Sentences

Koki Shirahige[*1], Taiki Matsui[*2], Takahito Tomoto[*1]
*1 Faculty of Engineering, Tokyo Polytechnic University
*2 Graduate School of Engineering, Tokyo Polytechnic University

プログラミング課題の文章問題において解が導けない学習者は，問題文中の情報を整理できず，解の導き方を理解できていない状態であると考えられる．学習者に解の導き方を理解させるためには，一連の問題解決過程に沿った思考をさせることが望ましい．そこで本研究では，プログラミング課題を対象とした文章問題の定式化を提案する．定式化では問題文中に存在する情報を整理し，その情報をもとに属性の関係式を求め，関係式から解を導く一連の流れを作成する．これらの活動を学習者に行わせることで，解法の理解を促すことが期待される．本稿では提案手法に基づいてシステムを開発・評価した．

キーワード: 外在化支援，定式化，プログラミング支援，学習支援システム

1. はじめに

プログラミング課題において文章問題を解く際に解を導けない学習者が存在する．解が導けない学習者は問題文中の情報を整理できず，解を導くための一連の流れを理解できない状態であると考えられる．このような学習者に解の導き方を理解させるためには，問題解決過程に沿った思考をさせることが望ましい．平嶋ら[1]は算数や数学，力学などの文章問題の領域で問題の定式化が重要であることを指摘し，問題解決過程を 3 つの段階に分けてモデル化した．著者らはこの問題解決過程をプログラミング領域に活用することで，プログラミング課題の文章問題において学習者が問題文から解を導くことが可能になると考える．

そこで本研究では，プログラミング課題を対象とした文章問題の定式化を行わせる学習支援を提案する．具体的には，平嶋らが算数や数学あるいは力学などの文章問題の領域で提案した定式化構造，制約構造，解法構造の三つの構造を活用する．学習者は問題に対応したそれぞれの構造を作成することにより，プログラミング課題における文章問題の解法の理解を促すことが期待される．開発したシステムは学習者に，問題文中の情報を整理させ（定式化構造），その情報をもとに関係式と属性を求めさせる（制約構造）．そして，求めさせた関係式と属性を用いて解を導く一連の流れを組み立てさせる（解法構造）．このような演習を学習者に行わせることで，学習者にプログラミング課題の解法の理解を促すことが期待される．

2. 先行研究

2.1 補助問題の定式化

2.1.1 問題解決過程

平嶋ら[1]は算数や数学，力学などの文章問題におけ

る問題解決過程を3段階に分けてモデル化した（図1）．問題解決過程は表層構造作成過程，定式化過程，解導出過程の三つの段階に分かれている．表層構造作成過程は，問題文に記されているオブジェクトと属性の関係を表現する表層構造を問題文から生成する過程である．定式化過程は，表層構造を数量関係が適用可能にした構造である定式化構造を生成する過程である．解導出過程は，定式化構造から数量関係を用いた変換をしていき，最終的に求めたい解を含んだ目標構造を生成する過程である．さらに，解導出過程において目標構造を導くまでの一連の数量関係を解法構造と呼ぶ．また，解法構造に含まれている数量関係は，その問題で対象としている状況の一部にすぎず，問題が対象としている状況に存在する数量関係のすべてを表現したものを制約構造と呼ぶ．

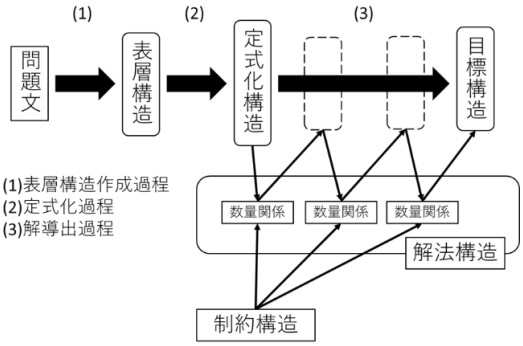

図 1 問題解決過程

2.1.2 表層構造と定式化構造

図 2 に表層構造と定式化構造を示す．力学の問題などでは，「ボールを静かに離す」といった明示的に含まれている「ボール」のようなオブジェクトと，「静かに離す」のような属性がある．この二つの関係を表層構造として表現することができる．また，表層構造で表現した「静かに離す」という属性を「速度 v0(T=0)」のように公式などに数量関係として適用可能な表現にした構造が，定式化構造である．

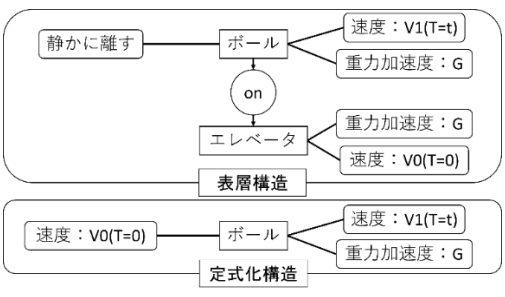

図 2 表層構造と定式化構造

2.1.3 制約構造

図 3 に制約構造を示す．制約構造は属性と数量関係で構成される．数量関係に対応する入出力属性を線で結び，問題解決に関係のないものも含めて問題の背景に存在する数量関係をすべて記述したものが制約構造である．例えば「ベクトル分解」は「斜面傾角」，「ブロック鉛直重力」，「ブロック斜面方向重力成分」の三つの属性が必要な制約である．そのため，制約構造では制約である三つの属性と「ベクトル分解」を線で結び，問題の中に存在する属性間の数量関係を表現する．

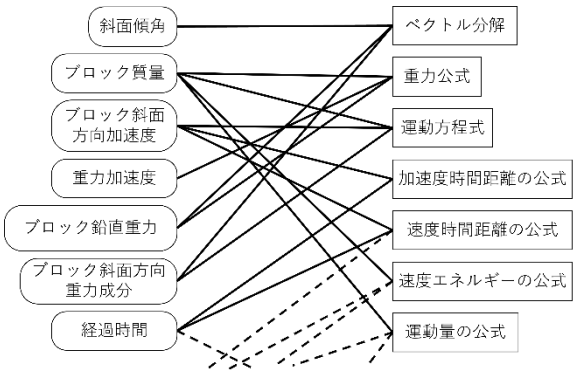

図 3 制約構造

2.1.4 解法構造

図 4 に解法構造を示す．解法構造は，属性と数量関係を使用し，数量関係を介して入力属性から出力属性を矢印で結ぶことで作成する．例えば「ブロック初速度」，「ブロック斜面方向加速度」，「移動距離」が入力属性であるとき，出力属性である「ブロックの斜面方向加速度」を導くことができる．そして入力属性と出力属性を結ぶ際に「$v1^2$-$v0^2$=2AS」のような数量関係を入出力属性の間に記す．このように解を導く一連の数量関係を表現した構造を解法構造としている．

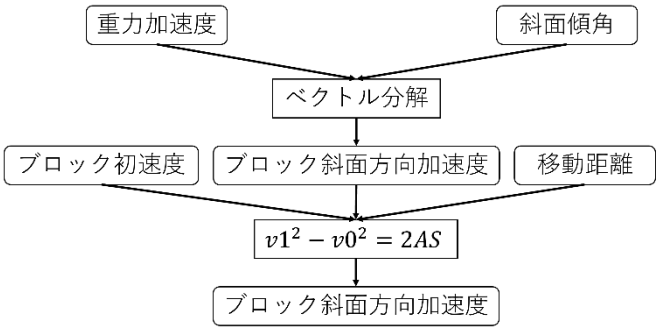

図 4 解法構造

2.2 プログラミング領域への活用

著者らはプログラミング課題の文章問題が解けない

学習者に対して，問題解決過程に沿った思考をさせることが重要であると考える．そこで本研究では，平嶋らが提案した問題解決過程をプログラミング領域に活用する．これにより，プログラミング課題を対象とした文章問題における解法の理解を学習者に促すことが期待される．先行研究の問題解決過程で対象としている数学や力学の領域では，問題解決に用いられる知識そのものが領域に依存した知識であり，数学の知識や力学の知識を構造として表現することで問題解決過程を表現できる．一方，プログラミングの領域においては，if 文や for 文に関する言語的な知識，カプセル化などの概念的な知識，あるいはソートなどの特定のアルゴリズムに関する操作を行うための知識などプログラミング特有の知識と，プログラミングを用いて解決しようとしている問題領域に関する知識の双方を必要とする．たとえば，微分積分に関するプログラミングを作るためには微分積分の知識を必要とし，物流システムに関するプログラムを開発するためには物流システムに関する知識が必要となる．そのため，先行研究の問題解決過程をそのままプログラミングに適用させることは一般的に困難であると考えられる．

そこで，本研究ではまずはプログラミングにおける問題で対象としている領域知識を制約構造として表現した上で，プログラミングの文章問題に対する定式化を通した問題解決過程を学習させる方法を提案する．なお，問題文で対象としている領域知識については，プログラミングの教師など専門家があらかじめ構造を提供してくれることを前提とし，制約構造そのものの正誤や質は本稿の議論の範囲外とする．

3. 提案手法

本研究では文章問題の定式化を学習者に行わせることで，解法の理解を促す手法を提案する．具体的には，平嶋ら[1]の定式化構造，制約構造，解法構造を活用し，学習者に問題に対応した構造を作成させることで解の導き方を段階的に思考させる学習支援を行う．先述したように，ここで表現されるべき知識は，プログラミング特有の知識と，プログラミングにより解決しようとしている対象の領域の知識の双方であるが，本研究では対象の領域の知識に焦点を当てた方法を提案する．

本研究において，定式化構造は問題文中のオブジェクトと属性の関係性を表現した構造とし，制約構造は問題に存在する属性と属性を構成する関係式を表現する構造とした．ただし，プログラミングにおいては，「３つの変数の最大値」などの複雑な関係式も許容する．そして，この定式化構造と制約構造から，解を求める一連の流れを属性と関係式を用いて表現した構造を解法構造としている．通常，プログラミングにおいては，入力された値やあらかじめプログラム上で規定された値を元に，最終的な値を求める（あるいは最終的な状態にする）ことが求められる．さらに，この値を変化させるものが関数などのコンポーネント単位の操作であると考えられる．したがって，プログラミングにおける解法構造は，データと操作を組み合わせたものにより生成されるデータフローダイアグラムに近い形となると考える．平嶋らの問題解決過程において，制約構造は問題の背景に存在するすべての数量関係であり，解法導出過程で参照されるものだった．しかし，物理などの領域において制約構造は「その領域の知識の構造」であったのに対し，プログラミングにおいてはプログラムが対象とする領域によって制約構造は異なると考えられる．そのため，本研究では制約構造を学習者に作らせることは，プログラミングの問題文が対象とする領域の知識の獲得の促進につながると考えている．そのため，本稿では制約構造も学習者自身に明示的に構築させる．以下にそれぞれの構造の作成において，学習者が行う活動とその目的について例示する．

まず，定式化構造を作成する活動について説明する．解法構造を作成するには，学習者が問題文中に登場するオブジェクトと属性の関係性を理解する必要がある．よって，学習者には問題文中の単語について，その単語をオブジェクトと属性に振り分けさせ，その関係性を記述させることで定式化構造を作成させる．例えば「入浴券１枚の価格とプール券１枚の価格，A さんが持っているお金が入力として与えられたとき，入浴券とプール券を１枚ずつ買った後に A さんが持っているお金を出力するプログラムを作成してください.」という問題文にはオブジェクトである「入浴券」や「プール券」という単語と，「入浴券」の属性である「（入浴券１枚の）価格」という単語がある．学習者にはこ

の「入浴券」と「(入浴券 1 枚の) 価格」のように関係している単語同士を接続する活動を要求する．この活動により，学習者は問題文中のオブジェクトと属性の関係性を理解することができると考える．なお，この活動はオブジェクト指向プログラミングにおける設計と近い活動であるが，オブジェクト指向プログラミングでなくてもこのような構造化は重要である．

次に，制約構造を作成する活動では，学習者は定式化構造で求めた属性を使用し，属性間の関係式を求める活動を要求される．例えば，「(入浴券の) 1 枚の価格と (入浴券の) 枚数と (入浴券の) 総額」の関係式を求めるとする．この場合，「(入浴券の) 1 枚の価格」という属性と「(入浴券の) 枚数」という属性を使用し，積を求めることで「(入浴券の) 1 枚の価格×(入浴券の) 枚数＝(入浴券の) 総額」という関係式を求めることができる．このように学習者は属性を使用した関係式を作成して制約構造を作成していく．なお，この活動においてはある属性がどのオブジェクトの属性であるかを意識する必要がある．上記の関係式はオブジェクトの入浴券をプール券に変更しても成立する関係であると同時に，プール券の 1 枚の価格×入浴券の枚数などオブジェクトを意識しない関係式を構築してはいけないことを意味している．この活動により，学習者は属性がどのように構成されているか考え，属性を構成するための関係式をオブジェクトと関連付けながら理解することができると考える．

最後に，解法構造を作成する活動では，学習者は制約構造で求めた関係式を使用し，入力から解を求める一連の流れを作成する活動を行う．例えば，「入浴券の総額」を求めるには「(入浴券の) 1 枚の価格」と「(入浴券の) 枚数」という属性の積をとることで求めることができる．さらに，「(プール券の) 総額」と和をとることで，「全体の総額」を求めることができる．

このように問題文から定式化構造を作成したのち，問題文が対象とする領域についての知識の構造である制約構造を作成し，解法構造を構築する一連の流れをすべて学習者は体験することになる．この活動により，問題文のどの要素が定式化されるか，対象としている領域において必要とされる知識は何であるか，定式化構造と制約構造を用いてどのように解法を導くプロセスを生成できるかを学ぶことになる．

4. 提案システム

提案システムは定式化構造作成画面，制約構造作成画面，解法構造作成画面の三つの画面で構成されている．ここでは三つの画面についての説明を行う．

4.1 定式化構造作成画面

図 5 に定式化構造作成画面を示す．この画面では，システムから問題文が提示される．本来は問題文から単語を抽出させる過程も存在するが，現時点のシステムでは抽出された後の単語をあらかじめ提示する．この単語は教授者があらかじめ設定した正解の構造から自動で作成される．学習者は提示された単語についてオブジェクトと属性のどちらかをボタンにより決定する．さらに，属性は関連するオブジェクトとリンク (関係を表す線) で接続させる．この一連の過程により学習者に定式化構造を作成させる．

例えば，図 6 では，「入浴券」と「入浴券の購入枚数」がある．学習者は「入浴券」をオブジェクトブロックとして画面に生成し，「入浴券の購入枚数」を属性ブロックとして生成している (以降，これを「入浴券/オブジェクト」「入浴券の購入枚数/属性」と表記する)．そして，学習者は「入浴券/オブジェクト」「入浴券の購入枚数/属性」のブロックには関係があるとして，二つのブロックをリンクしている．このようにブロックの生成とリンクの接続を繰り返し行い，学習者は定式化構造を作成していく．

次に，フィードバックについて説明を行う．ここでは教授者があらかじめ作成した正しい構造と比較することで差分を提示する．具体的には「オブジェクトブロックの正誤」,「属性ブロックの正誤」，そして「ブロックの接続関係の正誤」の三つについて診断を行う．例えば，学習者が誤った単語をオブジェクトブロックとして生成した場合，「オブジェクトで生成してはいけないブロックが○個あります」とフィードバックされる．また，別の例として，学習者が「入浴券の割引料金/属性」を「入浴券/オブジェクト」にリンクで接続しておらず，「プール券/オブジェクト」に接続している場合，入浴券の割引料金と入浴券は接続する必要があり，入浴券の割引料金とプール券は接続してはな

らない．この場合，システムは「リンクで接続してはいけない箇所が一つ，リンクで接続しなければいけない箇所が一つあります」とフィードバックを行う．

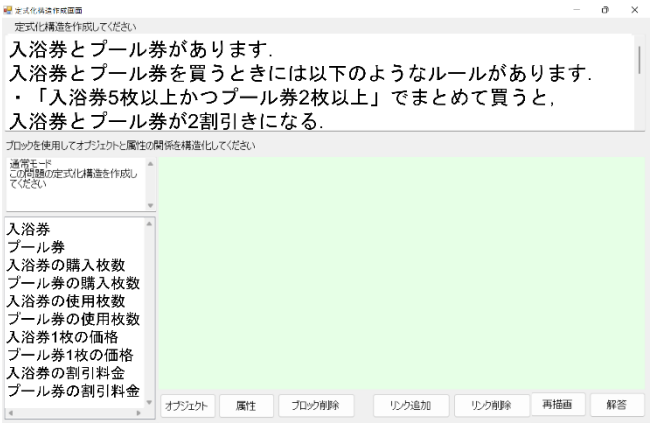

図 5 定式化構造作成画面

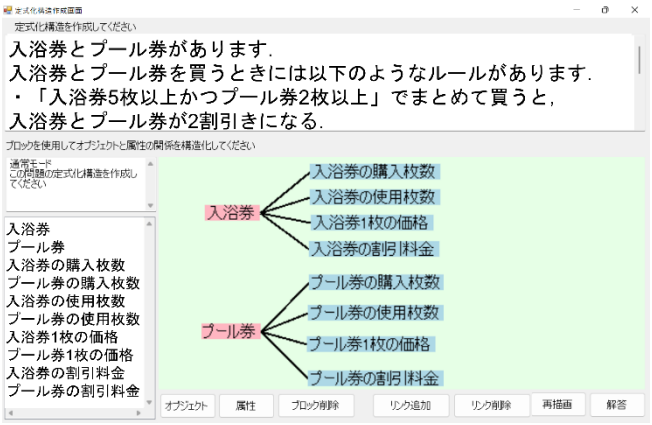

図 6 学習者が定式化構造を作成した状態

4.2 制約構造作成画面

図 7 に制約構造作成画面を示す．学習者は複数の属性を用いて問題の背景となる関係式を構築し，制約構造を作成する．具体的には，システムが問題の属性の一覧と「=」のような関係式の雛形を提示し，学習者は提示された関係式の雛形に属性を当てはめることで，制約構造を作成する．この雛形や属性の一覧は教授者があらかじめ設定した正解の構造から自動で作成される．なお，二つの値の最大値（Max）の値を入れるなどの関係式も扱っている．

例えば，図 8 では学習者が「A（入浴券の購入枚数）」と「B（入浴券の使用枚数）」を選択し，「A=B」という関係式を作成する．このように属性を選択し，関係式を作成することで制約構造を作成している．

次に，フィードバックについて説明を行う．制約構造作成画面では，関係式の左辺を「求める属性」，右辺を「左辺を求めるために必要な属性の式」として，診断を行う．左辺が誤っている場合は誤っている個数を表示し，誤っている箇所の色が変更される．右辺が誤っている場合，誤っている関係式を表示する．例えば，学習者が「H（割引されていない入浴券にかかる合計料金）」，「B（入浴券の使用枚数）」，「G（入浴券 1 枚の価格）」を「H=B*G」と入力した場合，正答は「A（入浴券の購入枚数）」を使用した「H=A*G」であるため右辺の関係式が誤っている．システムは「以下の式の右辺が間違っています　H=B*G」とフィードバックを行う．

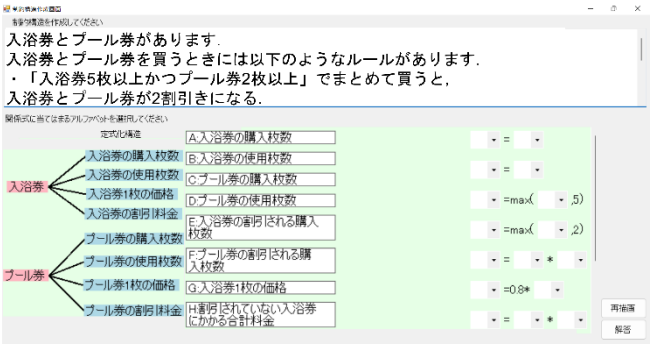

図 7 制約構造作成画面

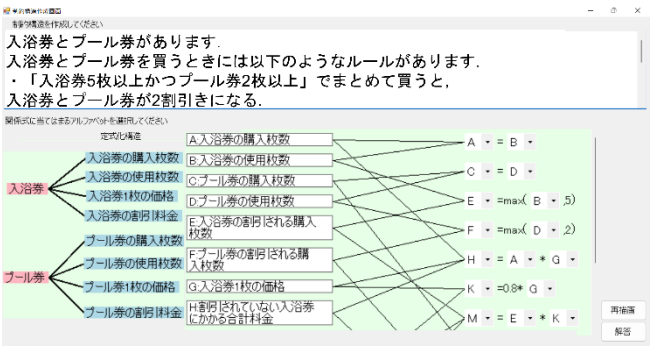

図 8 学習者が制約構造を作成した状態

4.3 解法構造作成画面

図 9 に解法構造作成画面を示す．この画面では，これまでに作成した制約構造をシステムが提示し，学習者が制約構造の制約を入力から出力までの一連の流れに当てはめることで解法構造を作成する．具体的には，システムが解法構造の雛形と，これまでに作成した制約構造および属性を提示する．この雛形も教授者があらかじめ設定した正解に基づいて自動で生成される．学習者は提示された制約構造と属性をみて，雛形にそれらを当てはめる．

例えば，図 10 では学習者はまず求めるべき属性と，問題文によって与えられる属性を考える．今回の問題

文では，「最も安いプール券と入浴券の総額」を求める問題であるため，「Q（最も安い総額）」が構造の終端（ルート）に来ることを決定できる（同様に与えられる属性はリーフとなる）．そのため，終端に「Q」を入力する．次に，学習者はこのルートを求めることができる関係式を制約構造から考え，「Q」につながる関係式に「Q=min(O,P)」を入力する．その後同様に繰り返すことでこの解法構造を作成していくこととなる．

次に，フィードバックについて説明を行う．解法構造作成画面では，「求める属性」，「属性を求めるための関係式」，「関係式に使用する属性」の正誤についてあらかじめ教授者に設定された構造を元に診断する．例えば，学習者が「Q=min(O,P)」に使用する属性を二つとも「O（割引されていない総額）」と解答したとする．しかし，ここでの正答は「O」と「P（割引された総額）」を選択することであり，関係式に使用する属性が誤っている．この場合，システムは「関係式を求めるための属性が不正解です」とフィードバックを表示し，誤っているコンボボックスの色を赤色に変更する．

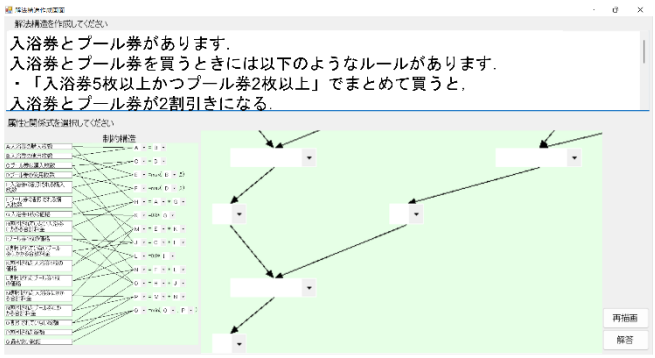

図 9 解法構造作成画面

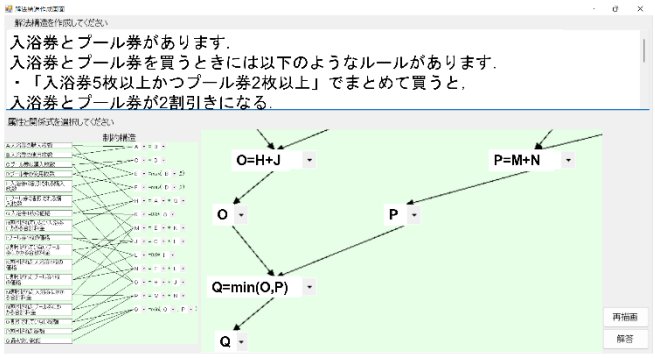

図 10 学習者が解法構造を作成した状態

5. 評価実験

5.1 実験概要

本システムの学習効果の検証を行うため，工学部の学生 16 名を対象に実験群 8 名，統制群 8 名に分けて評価実験を行った．実験は事前テストを 40 分，実験群と統制群に分かれての学習活動を 60 分，事後テストを 40 分，アンケートの順で行った．事前・事後テストと学習活動の際に提示する問題は同じ問題とした．

事前・事後テストは，プログラミング課題を提示し，定式化構造，制約構造，解法構造の三つの構造を作成するテストとなっている．学習活動では，実験群には提案システムを使用してもらい，統制群には実際にプログラムを構築する学習を行ってもらった．なお，実験群，統制群ともに扱った問題は事前テストで扱ったプログラミング課題となっている．アンケートでは，プログラミング課題を解く際に提案手法に基づいた思考を意識しているかといった質問をした．

5.2 実験結果

5.2.1 事前・事後テスト

事前・事後テストのスコアを表 1 に示す．テストは全 3 問存在し，問題ごとに，①定式化構造作成，②制約構造作成，③解法構造作成の 3 つの小問に分かれており，9 点満点となっている．総得点については，実験群の事前テストの平均点は 6.63 点で，事後テストの平均点は 8.00 点となった．一方で統制群の事前テストの平均点は 5.88 点で，事後テストの平均点は 6.25 点となった．テストの平均点と標準偏差から効果量を算出した結果，実験群では 1.60 で統制群では 0.33 という結果となった．これにより実験群は効果量が大きく，統制群は効果量が小さいという結果となった．このことから，提案システムは通常のプログラミングの作成を行う活動に比べて，問題の定式化を通した解法構造の構築に有効であることが分かった．

さらに，表 2，表 3，表 4 に小問ごとの平均点，標準偏差，効果量を記述する．定式化構造については実験群の効果量は中，統制群の効果量は小となり，実験群の方がやや効果が見られた．制約構造，解法構造については，いずれも実験群の効果量が大，統制群の効果量が大という結果になった．

以上より，問題文から定式化を行い，解法構造を構築する一連の過程において提案システムの方が単にプログラミングを構築するより効果があることが確認できたが，特に制約構造と解法構造の構築過程において顕著な差が出ることが分かった．今回，プログラミング課題の対象となる領域を制約構造として定義し，その構築を学習者に要求する手法を提案したが，その手法による効果により対象となる領域の知識構造が適切に構築された可能性が示唆された．

表 1 事前・事後テストの総合得点

	平均(標準偏差)		効果量	目安
	事前	事後		
実験群	6.63(0.70)	8.00(1.00)	1.60	大
統制群	5.88(1.27)	6.25(0.97)	0.38	小

表 2 定式化構造の得点

	平均(標準偏差)		効果量	目安
	事前	事後		
実験群	2.86(0.35)	3.00(0.00)	0.5	中
統制群	2.00(1.22)	2.33(1.00)	0.3	小

表 3 制約構造の得点

	平均(標準偏差)		効果量	目安
	事前	事後		
実験群	1.88(0.35)	2.50(0.53)	1.38	大
統制群	1.56(0.73)	1.67(0.71)	0.15	無

表 4 解法構造の得点

	平均(標準偏差)		効果量	目安
	事前	事後		
実験群	1.88(0.35)	2.50(0.53)	1.38	大
統制群	1.67(0.71)	1.56(0.73)	0.15	無

5.2.2 アンケート

アンケート結果の一部を表 5 に示す．アンケートは 6 件法で行い，6 が「非常にそう思う」，5 が「そう思う」，4 が「どちらかといえばそう思う」，3 が「どちらかといえばそう思わない」2 が「そう思わない」，1 が「全くそう思わない」として平均を集計した．表 5 より全項目において高い評価を得ることができた．質問 1，質問 2，質問 3 は実験群，統制群の両方にアンケートをとり，それ以外の質問は実験群にのみアンケートをとった．アンケート結果より，本研究で提案したプログラミングの問題を解くために問題の構造を作成し，処理の流れを考えることによって解を導くことが重要であると考えていることが確認できた．

表 5 アンケート結果

	実験群	統制群
1.プログラム問題で問題文を理解することは重要だと思いますか？	5.33	5.13
2.プログラミングの問題を解く際に，処理の流れを考えることは大切だと思いますか？	5.67	5.88
3.プログラミングの問題を解く際に，データの流れを考えることは大切だと思いますか？	5.67	5.50
4.本システムを用いた学習は，プログラミングの問題を解く際に，データの流れを考える能力の向上につながると思いますか？	5.33	-
5.本システムを用いた学習は，プログラミングの問題を解く際に，問題文を理解するために構造を段階的に作成する能力の向上につながると思いますか？	5.33	-
6.本システムを用いた学習はプログラミングの学習として有効だと思いますか？	5.33	-

6. おわりに

プログラミング課題において，文章問題の解を導くことができない学習者に対しては，問題解決過程に沿った思考をさせることが望ましい．そこで著者らは平嶋ら[1]がモデル化した問題解決過程をプログラミング領域に活用することでプログラミング課題の文章問題において学習者が問題文から解を導くことが可能になると考えた．

本研究では，プログラミングの問題解決において構造化するべき知識として，プログラミング特有の知識とプログラミング課題が対象とする領域についての知識を規定し，対象とする領域についての知識を主とした構造化を行った．本稿では，学習者に構造を整理させるための手法とその支援システムを提案した．

評価実験の結果から，本システムを利用せずに通常のようにプログラミングの構築を通して学習した学習者に比べ，システムを使って学習した学習者の点数が上昇していたことがわかった．よって，本システムを利用することでプログラミング課題から定式化を行い，解法構造を作成する能力の向上が促進されることが示唆された．

今後の課題として，本研究ではプログラミング課題そのものが対象とする領域についての構造化のみを対象としたが，本来はプログラムと構築するための知識と組み合わせることで初めてプログラミングを行える

はずである．たとえば，ある 3 つの値と最大値の間に関係式を引けたとしても，3 つの値から最大値を求める操作を行うコンポーネントを構築できるとは限らない．そのため，コンポーネントの構築に関する知識を今後本構造に付与する必要がある．著者らはこれまでにプログラミング特有のコンポーネントに関する知識の構築に関する研究[2][3]を推進しており，この研究と組み合わせることを予定している．さらに，評価実験においては，構築活動そのものの体験による差であるのか，構築に関する能力が向上したのかを厳密には評価できていないため，今後より精査された評価実験が必要である．

謝辞

本研究の一部は JSPS 科研費 JP22K12322，JP21H03565，JP20H01730 の助成による．

参 考 文 献

(1) 平嶋宗, 東正造, 柏原昭博, 豊田順一: “補助問題の定式化”, 人工知能学会誌, Vol.8, No. 8, pp. 211-218 (2017)

(2) 古池謙人, 東本崇仁, 堀口知也, 平嶋宗: “プログラミング学習における再利用性を指向した知識組織化のための知的支援: 機能・振舞い・構造の観点に基づく問題解決過程のモデル”, 人工知能学会論文誌, Vol.35, No. 5, pp. C-J82_1-17, (2020)

(3) 古池謙人, 東本崇仁, 堀口知也, 平嶋宗: “プログラミングの構造的理解を指向した部品の段階的拡張手法の提案と支援システムの開発・評価”, 教育システム情報学会誌, Vol.36, No. 3, pp. 190-202, (2019)