

部品の利用と合成を指向した プログラミング学習支援システムの開発

Development of Programming Learning Support System for Utilization and Synthesization of Components

座間 出実¹ 古池 謙人² 東本 崇仁¹
Izumi ZAMA¹, Kento KOIKE², Takahito TOMOTO¹

¹ 東京工芸大学工学部

¹ Faculty of Engineering, Tokyo Polytechnic University

² 東京工芸大学大学院工学研究科

² Graduate School of Engineering, Tokyo Polytechnic University

Abstract: One of the difficulties in system development is the use of components, such as a single syntax, functions and synthesis activities. In this study, we developed a system to support the use of components and the acquisition of synthesis ability. This system presents two goals: a goal for learning the behavior of individual components and one for synthesizing the learned components to create new components. The goal is to help learners acquire the ability to use and synthesize components by asking them to think about the behavior resulting from the use of the components.

1 はじめに

ソフトウェア開発では、言語固有の組み込み関数や、外部ライブラリの関数を利用することが要求される。そのためプログラミングにおいては、このようなブラックボックス化した要素を部品として組み合わせる能力が重要である。

しかし初学者にとって、こういったブラックボックスの部品を扱うことは困難である。その原因に、変数や代入、If文やFor文などの今まで理解してきた基礎構文と接地していないために、動作が予測できないことが挙げられる。したがって、部品の利用および合成能力を身につけるための学習支援が必要である。

プログラミングにおける部品を取り扱った研究として、古池らの一連の取り組みが挙げられる [1, 2]。古池らはこれまで、基礎構文レベルから部品を構築し、さらにその部品を再利用しながら段階的に大きな部品へと発展させていく、といった学習支援を行ってきた [1]。また古池らは、部品を用いた問題解決過程のモデルも提案している [2]。このモデル中では、部品の理解において、機能（何を達成するか）と構造（ソースコード）の間の関係性だけでなく、暗黙的に振舞い（ソースコードがどのような動作を生成するか）が介在していることを指摘している。すなわち、機能の達成に直接関わっているのはその部品が振舞う動作であり、構造そのものではないという指摘である。したがって、ブラック

ボックス化した部品であっても、振舞いの理解を伴えば機能的に活用できることを示唆している。

しかし、古池らによる支援はプログラミングにおける基礎構文に接地させた部品構築および拡張を指向しており、先に述べたような、ブラックボックス化した部品の利用と合成については支援されていない。ブラックボックス化した部品を学習者が理解し、扱えるようにするためには、古池らが指摘するように振舞いの理解が必要であると考えられる。

そこで本研究では、プログラミングにおいてブラックボックス化した部品の利用と合成の能力獲得を支援するシステムを開発する。

2 プログラミングにおける部品

2.1 部品の定義

プログラムを構築する中で、同じ処理を記述することはしばしば存在する。一方で、同じ処理を複数記述してしまうと、可読性や変更容易性が下がることが知られている。そこで、特定の処理に機能的なラベルを付けることで、再利用可能にしたものが一般に言われている関数である。

古池らは、このような一定の機能を果たすソースコードの有意味な塊を部品と定義している [1, 2]。また、1

章で述べたように、部品をプログラム中で扱うことは、達成される機能と構造（ソースコード）の関係を記憶するだけでなく、機能と振舞い（その機能がどのように達成されているか）の関係性を理解することでも実現できると考えられる。

2.2 部品の利用と合成に関する考察

先行研究では、部品の段階的拡張手法という学習法を用いたプログラミング学習支援システムの開発が行われてきた [1]。部品の段階的拡張手法とは、学習者自身が条件や代入、繰返し処理などの十分に習得している部品を組み合わせて部品を構築し、さらにその部品を再利用しながら段階的に大きな部品へと発展させていく手法である。よって先行研究では、この手法を用いて部品の再利用性や構造的理解を支援している。

しかし、実際のソフトウェア開発において全ての部品を学習者自身が作成することは少なく、組み込み関数やライブラリの関数などの部品を利用したり、それらの部品を組み合わせて新たな部品を作る機会が多いと考えられる。そこで本研究では、学習者自身が作成していない部品を利用すること、学習者自身が作成していない部品らを合成して新たな部品を作成する能力獲得を支援対象とする。

先行研究と本研究における支援対象の差異は図 1 のように示すことができる。1 章で述べたように、先行研究ではプログラミングにおける基礎構文に接地させた機能構築および拡張を支援している。一方で本研究では、ブラックボックス化した部品の利用と合成の支援を対象としているといった違いがある。

したがって古池らの部品の定義に基づき、本研究で取り扱う部品は「一定の機能を果たす暗黙的な構造を持ったステートメント」とする。例えば、プログラミング言語が提供するような組み込み関数や自分自身で定義し作成した関数も部品であるといえる。このような部品を利用することは、プログラムの記述量や可読性、変更容易性においてメリットがあると考えられる。したがって、部品の利用と合成を能力として身につけることは、プログラミングを学ぶ上で重要な要素の 1 つと考えている。

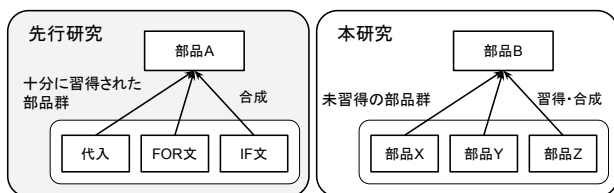


図 1: 先行研究と本研究の支援対象の差異

3 提案手法

これまで述べてきたように、組み込み関数やライブラリの関数などのブラックボックス化した部品を学習者が利用したり、それらの部品を組み合わせて新たな部品を作るためには、振舞いを理解することが必要であると考えられる。よって本章では、システムから与えられた部品に対して、出力される振舞いを観察しながら入力と試行錯誤することで、部品の利用と合成の能力獲得を支援する手法を提案する。

提案手法は、単体目標とする単一の部品を習得する問題と、合成目標とする習得した複数の部品を合成する問題から構成される。また、前者の問題解決過程を単体目標フェーズ、後者の問題解決過程を合成目標フェーズとし、前者で習得する部品を単体部品、後者で習得する部品を合成部品とする。

学習者が部品を合成するためには、合成に用いる単体部品の振舞いを理解している必要がある。よって、まず単体目標フェーズで合成させる単体部品を習得する。次に、合成目標フェーズに移る。合成目標フェーズでは、既に習得した単体部品を組み合わせて要求を満たす機能を構築する。

3.1 単体目標フェーズ

単体目標フェーズは、それぞれ独立した複数の単体目標から構成され、単体目標 1 つにつき 1 つの部品の習得を行う。図 2 には、単体目標フェーズ全体における部品習得の流れを示している。

初めにシステムは、単体目標に設定されている部品の実行例を学習者に提示する（図 2 の 1 に相当）。学習者は、システムから提示された部品の実行例を実行することで、システムは実行結果とフィードバックを与え、学習者は振舞いを観察する（図 2 の 2 と 3 に相当）。

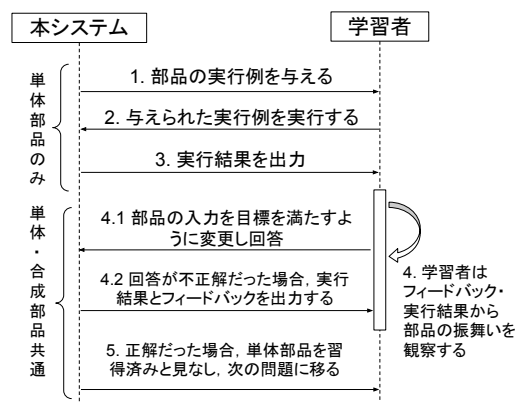


図 2: 部品習得の流れ

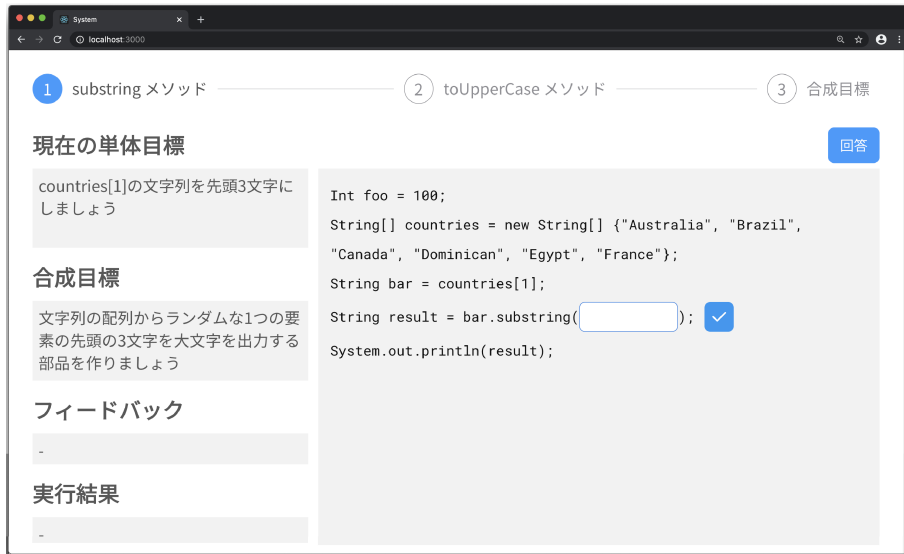


図 3: システム画面（単体目標フェーズ）

次に、学習者は、実行結果とフィードバックをもとにして要求を満たすまで部品を入力を変更し実行することで解答を行う（図2の4に相当）。このように学習者が観察した振舞いから帰納的に入力の内容を変え、さらに、繰り返し振舞いを観察することで部品の振舞いを習得できると考える。解答が正しければ、システムは学習者がその部品を習得したと認識し、次の問題を提示する。

3.2 合成目標フェーズ

合成目標フェーズには、単体目標フェーズで提示される全ての部品が習得状態になった場合にのみ移行する。このフェーズで学習者は、習得した単体部品を組み合わせ、合成目標と同じ出力結果をもつソースコードの構築を目指す。

合成目標は、単体目標と異なり例が与えられない（図1中の4~5のみの流れ）。したがって学習者は、それぞれの単体部品の振舞いと、部品間の入出力の関係を思考する必要がある。

本フェーズで学習者は、出力結果が目標を達成するまで解答を行う。ここでは正解の場合は実行結果が、不正解の場合は加えてフィードバックが表示される。

4 システム実装

本章では、提案手法を実現するシステムの実装および学習の流れについて説明する。本システムが提示するプログラミング言語として、Javaをベースとした擬似言語を使用する。また本システムは言語処理系を内

包するように開発している。これにより、解答の実行、学習者の解答内容を分析することや、未定義変数の使用などのエラーを含んだフィードバックを柔軟に行うことが可能となる。

本システムの対象者はプログラミング初学者、特に基礎構文に関する知識は有しているが、関数の呼び出しや組み合わせが意図した通りに行えない学習者である。

4.1 インタフェース

図3および図4に、単体目標および合成目標フェーズのシステム画面を示す。画面上部には、全体の進捗状態を表すプログレスバーが表示され、学習者が進捗状態を確認できる。画面左側には、上から順に、現在学習中の単体目標または合成目標、実行後のフィードバック、実行結果の欄がある。画面右側には、ソースコードを編集するためのエディタが表示される。学習者は、エディタを用いてソースコードの編集を行い、ソースコードを実行する。エディタは、構文エラーなど学習の本質的ではないミスを減らすため、部品の学習に関する特定の場所のみを編集可能としている。

4.2 フィードバック

本システムでは、学習者の構築したソースコードの実行結果をフィードバックとして返す。

実行結果は、プログラムの返す出力とエラーの2つに分けられる。学習者が解答した内容にエラーがない場合、プログラムの返す出力のフィードバックでは、予めシステムが与えたプログラムに記述された出力部品

現在の単体目標

文字列の配列からランダムな1つの要素の先頭の3文字を大文字を出力する部品を作りましょう

合成目標

文字列の配列からランダムな1つの要素の先頭の3文字を大文字を出力する部品を作りましょう

```
String[] countries = new String[]{"Australia", "Brazil", "Canada",
    "Dominican", "Egypt", "France"};
String subStr = countries[1].substring(0,3);
String result = countries[0].toUpperCase();
System.out.println(result);
```

回答

図 4: システム画面 (合成目標フェーズ)

に与えられた変数の中身が出力される。構文エラーが存在する場合には実行エラーを出力する。例えば学習者が未定義変数を使用して解答した場合、システムは実行エラーとして「未定義の変数を利用することはできません」といった内容をフィードバックする。

4.3 学習の流れ

3章で述べたように、本システムは単体目標フェーズと合成目標フェーズから構成される。本節では具体例を交えながら、各フェーズについての説明を行う。

まず、学習者はシステムから提示される1つ目の単体目標と、合成目標の内容を確認する。例えば、文字列結合の部品と現在の日付を出力する部品の2つの単体目標と、名前の後に現在の日付の文字列を出力する合成目標が提示される。

1つ目に提示される単体目標は文字列結合部品であり、問題文として「文字列'hello' と文字列'world' から'hello world'を出力しなさい」が提示される。ここで学習者は、システムが例として与えたconcat('hello', 'there')を実行し、システムは'hello there'を実行結果欄に表示する。学習者が例を実行して振舞いを観察し、与えた文字列が結合される振舞いを観察することで、未知の部品の利用方法を学習させる。最後に、'there'の部分を'world'に変更し正解することができた場合は、次の単体目標に移る。

2つ目の単体目標としては、現在の日時を出力する部品が提示される。この際の問題文は、「10月19日のような〇〇月〇〇日のフォーマットになるように現在の日付を出力しなさい」である。先ほどと同様に、システムは例としてDate('YYYY-MM-DD')を与える。この例を実行すると、'2020-04-01'と出力される。このような振舞いを観察した上で、学習者がDate('MM月DD日')と解答できた場合に正解となる。

これらの単体目標フェーズで不正解になった場合も実行結果を出力することで、フィードバックを行い、学習者に振舞いを学習させる狙いがある。

単体目標の習得が終わると、合成目標フェーズに入る。単体目標で習得した文字列結合部品と現在時刻を出力する部品から、名前と日付を'John20200401'結合した文字列を出力する部品を作成する合成目標が与えられる。エディタには、次のような編集箇所があり、String today = Date(空欄); String result = concat(空欄, 空欄); 空欄に変数名や値を入力し解答する。解答時に実行結果が表示される。

5 おわりに

本稿では、部品の利用と合成について能力の獲得を支援する学習手法を提案し、そのシステムを実装した。実装したシステムでは、部品に対して入力を与えて実行させ、部品の振舞いを観察させるといった試行錯誤を学習者に繰り返させることで、部品の利用と合成能力の獲得支援を目指した。

今後の課題としては、本システムを用いた評価実験が挙げられる。

謝辞

本研究の一部は科研費・基盤研究(C)(18K11586)の助成による。

参考文献

- [1] 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.
- [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-C, pp. 1-17, 2020.