K-032

プログラミング教育のための語句選択を用いた構造化プログラム設計

Structured Program Design with Combination of Phrases for Programming Education

稲葉 大祐 † 原田 史子 ‡ 島川 博光 ‡
Daisuke Inaba Fumiko Harada Hiromitsu Shimakawa

1. はじめに

大学のプログラミング教育では，講義において，学習者はプログラミング言語やアルゴリズムといったプログラムを作成する上で必要となる内容の学習をする．これらの内容は密接な係わり合いがあり，一般的に両者は並行して学習が進められる．演習において，学習者はプログラミング言語やアルゴリズムを理解しながらソースコードを作成する．教員はそれを評価することにより学習者の理解度を把握している．このように，現在の教育ではソースコードを作成する演習に重点が置かれている．

現在のプログラミング教育における問題点として 2 点挙げられる．第 1 はソースコードを作成することのみに重点が置かれている点である．特定のふるまいをするプログラムを実現するためには，プログラミング言語の知識に加えてプログラムの制御構造への変換に関する知識が必要である．第 2 は段階的詳細化に対する指針を示せていない点である．明瞭な構造をもち，かつ正しく動作するプログラムを作成するためには，プログラムのふるまいを大まかな機能に分割し，さらにその中を詳細化していく必要がある．そこで本研究では，プログラムのふるまいを手続きとして記述し，構造化プログラム設計をする手法を提案する．

2. プログラムの設計

2.1 プログラミング言語に依存しない設計

目標とするプログラムを作成するためには，プログラムのふるまいを手続きとして組み立てる必要がある．手続きとして組み立てずにソースコードの作成をすると，処理の流れや構造の把握をしにくいスパゲッティコードになる恐れがある．プログラムのふるまいを手続きとして組み立てることができなければ，思い通りにプログラムを作成することは難しい．そのため，教員はプログラムのふるまいを手続きとして組み立てる方法を学習者に教える必要がある．

プログラムを設計するとき，その設計方法はプログラミング言語と切り離されるべきである．その理由は，プログラミング言語の理解によって設計の結果が左右されてしまうため，学習者がプログラムの設計に集中できないからである．そこで，プログラミング言語の細かな文法の整合性を気にすることなく，プログラムのふるまいを手続きとして論理的に組み立てることにより設計する演習が必要になる．したがって，プログラムのふるまいをプログラミング言語とは別の表現方法を用いて組み立てる必要がある．

2.2 段階的詳細化

プログラムを設計するために，まず大まかな機能を設計し，それらを段階的に機能分割をすることにより詳細化していく必要がある．プログラムのふるまいは，順次・選択・反復の 3 つの制御構造を用いて組み立てられる．これらの制御構造を用いてプログラムを段階的に詳細化して設計することにより，正しく分かりやすいプログラムを作成することができる．しかし，プログラミングの初心者にとって，機能分割をしてモジュールを作成し，モジュール間の値の授受を理解することは容易でない．

プログラムを詳細化するために，教員は学習者に事例を用いてプログラムのふるまいの設計方法について説明し，その演習を実施する必要がある．まず，教員は機能分割をして制御構造を用いて組み立てる方法を学習者に提示する．そして，処理のまとまりを部品化したものであるモジュール間を結合するための手続きを考える必要がある．これらを教員が例を示すことにより学習者に理解させた後，学習者がそれらを理解しているかを確認する手段が必要になる．

3. 構造化プログラム設計

3.1 語句選択による設計

語句選択を用いてプログラムを段階的詳細化に沿って設計する手法を提案する．本手法では，模範的な設計をするために用意された課題について，学習者が構造化プログラム設計をする．教員が学習者にプログラムのふるまいを手続きとして組み立てる方法を教えるために，設計段階は基本設計と詳細設計の 2 つに分割される．基本設計ではプログラムの全体像を設計する．詳細設計ではモジュール間の関連を考慮し，各モジュールについて詳細な手続きを設計する．本手法により，3 つの制御構造を用いて論理的にプログラムを設計する方法を教えることができる．さらに，模範となる抽象度の落とし方を教えることにより，プログラムの設計方法に関する 1 つの指針を示すことができる．

学習者は，フローチャートや PAD といったアルゴリズムの記法を使用し，プログラムのふるまいを手続きとして記述する．本論文ではフローチャートを例に挙げ，それを用いて学習者が構造化プログラム設計をする方法

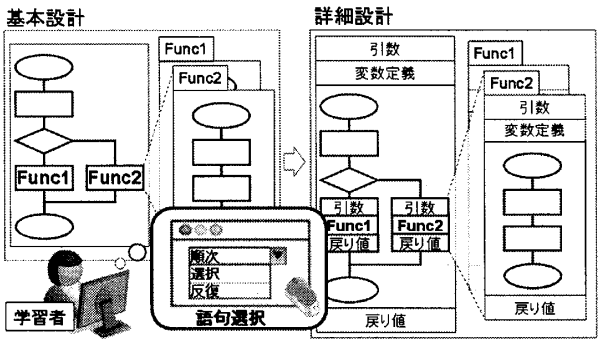

図 1: フローチャートを用いた構造化設計

† 立命館大学大学院 理工学研究科 情報理工学専攻
‡ 立命館大学 情報理工学部 情報システム学科

について述べる．図 1 にフローチャートを用いた設計方法の概要を示す．学習者は，順次・選択・反復の 3 つの制御構造のうち 1 つを選択し，その制御の内容を示す文を語句選択により作成する．これにより，プログラムのふるまいが手続きとして記述される．語句選択により学習者は，基本設計では処理の流れを作成し，詳細設計では処理の流れに加えて，変数とその役割を作成する．同様に繰り返し処理を追加することにより，学習者は完成したフローチャートを作成する．したがって，学習者はプログラミング言語の細かな記法を気にすることなく，論理的にプログラムのふるまいを設計することができる．

3.2 基本設計

基本設計は，語句選択を用いて処理の流れを記述することにより，プログラムの全体像を設計する段階である．この段階では，Start と End のみで構成されたフローチャートが与えられる．学習者はそのフローチャートに処理を追加することにより，フローチャートを完成させる．メインとなる処理を組み立てた後，学習者は処理を分割し，細かい機能を組み立てる．処理内容は，重要な語句が穴抜きされた文であるモデル文とその空欄を埋める語句の候補群である解答語句の選択により記述される．

手続きの詳細化にはモデル文を用いる．モデル文は詳細化の段階ごと，制御構造ごとに用意され，学習者はモデル文を用いて段階的に設計を詳細化していく．モデル文と解答語句を選択することにより作成された手続きは，階層的に管理される．階層的に処理内容を管理することにより，詳細化の段階を学習者に明示することができる．基本設計により，教員は学習者にプログラムのふるまいを階層的に考えることを強制できる．

3.3 詳細設計

詳細設計は，処理の流れに加えて変数とその役割を記述することにより，プログラムのふるまいを手続きとして設計する段階である．この段階ではモジュール間の値の授受を考慮し，基本設計で作成された各々の手続きを詳細に記述する．モジュールを呼び出す側では，引数と戻り値を明らかにし，呼び出される側は，引数を受け取った後に戻り値を返すまでに必要な手続きを明らかにする．

図 2 に基本設計と詳細設計の関係を示す．詳細にプログラムのふるまいを設計するために，基本設計で作成された手続きを実現するために必要となる，変数とその役割を記述する．変数の役割は 11 個に分類される [1]．本手法では，変数の役割を学習者に選択させることにより，変数の使用目的を明確にする．

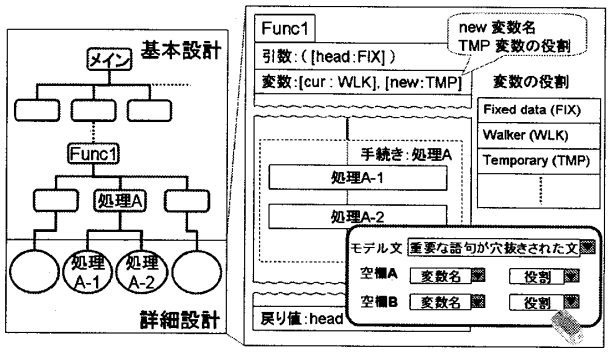

図 2: 基本設計と詳細設計の関連

手続きはモデル文，変数とその役割を選択することにより作成される．引数や戻り値は変数を用いて表現される．たとえば図 2 において，モジュール Func1 を実現するために，引数 head と変数 cur, new が定義されている．head, cur, new の役割はそれぞれ，Fixed data, Walker, Temporary である．また，変数を用いて処理 A の手続きを記述する場合，処理 A-1 と処理 A-2 が必要になる．それらの処理をモデル文，その空欄の穴を埋める変数とその役割の選択により作成する．

変数の概念を用いることにより，値の変化の様子やモジュール間の値のやり取りを意識して手続きを記述することを強制できる．教員は学習者に基本設計で作成した手続きを詳細化する方法と，各機能間の関連をまとめる方法を教えることができる．

4. 関連研究

プログラムのふるまいを手続きとして記述し，設計を支援するシステムとして，SICAS[2] と SFC[3] を挙げる．

SICAS はフローチャートを用いて，プログラムのふるまいを設計し，そのシミュレーションをすることができる．SICAS は入出力や関数といったプログラミング言語に特化した内容を記述することができる．そのため，学習者は設計したアルゴリズムのふるまいを視覚的に確認できる．しかし，プログラムを大枠から設計し詳細化することができないため，学習者は段階的詳細化に沿って設計する方法を学習することができない．

SFC はフローチャートを用いてプログラムのふるまいを詳細に組み立てることができる．SFC を用いてフローチャートを作成すると，自動的に擬似コードが生成される．関数ごとに設計ができるため，関数間の関連を考えつつ設計をすることができる．しかし，設計における自由度が高いため，プログラミング初心者が設計方法を学習するために利用することには不向きである．

5. おわりに

本論文では，学習者に段階的詳細化に沿ってプログラムのふるまいを組み立てる方法を教えること目的に，語句選択を用いて構造化プログラム設計をする手法を提案した．今後は本手法の有効性を検証するために，実験・評価をする予定である．

参考文献

[1] Byckling, P. and Sajaniemi, J.: A Study on Applying Roles of Variables in Introductory Programming, *VLHCC '07*, Washington, DC, USA, IEEE Computer Society, pp. 61–68 (2007).

[2] Mendes, A. J., Gomes, A., Esteves, M., Marcelino, M. J., Bravo, C. and Redondo, M. A.: Using Simulation and Collaboration in CS1 and CS2, *SIGCSE Bull.*, Vol. 37, No. 3, pp. 193–197 (2005).

[3] Watts, T.: The SFC editor a graphical tool for algorithm development, *J. Comput. Small Coll.*, Vol. 20, No. 2, pp. 73–85 (2004).