

# チュータリングシステムにおける意思決定と 詳細化に基づくプログラム設計知識の形式化

2 S - 4

鈴木 彦文<sup>1</sup> 関本 理佳<sup>2</sup> 海尻 賢二<sup>2</sup>

<sup>1</sup>長野高専 電子情報 <sup>2</sup>信州大学 工学部 情報工学科

## 1 はじめに

学生に対するプログラム演習を考えると、プログラム作成の段階は (1) モデル化 (2) アルゴリズム設計 (3) プログラム設計 (4) プログラム化 からなると本研究では考える。学習者に対してこのような段階を踏まえるプログラミングを指導する時、問題となるのが設計から具体的なプログラムを作成する場合の指導をいかに行うかである。特に、設計を行っている際に生じた意思決定をふまえた上での段階的詳細化の指導を行うことが望ましい。しかし、意思決定がプログラムにどのように反映されるのかを確かめるながら指導することは困難である。したがって、このような指導を行うための情報をどのように表現し蓄積するのかが重要となってくる。

もちろん、開発に利用するモデルで、特に意思決定やプログラム設計に関するモデルは幾つか存在する。IBIS[1, 2] や DRL といったモデルでは、意思決定やその案といった情報を表現することができる。しかしながら、各々の情報が具体的に結び付いた結果、設計がどのようになり、最終的にプログラムにどのように反映されたかに至る過程の情報は盛り込みにくい。また、プロセスプログラミングも、設計から具体的な成果物であるプログラムがどのように作成されたのかの情報を十分に表現できない。これはこれらのモデルが、どちらかと言えば上流工程に重きを置いているためである

PAD などのチャート形式のプログラム設計 (表現法) では、プログラムの構造は明確になる。しかし、どのような決定から作成されたものなのかがはっきりしない。これはチャート形式で表現されたものが、ほぼプログラムそのものであるということが原因である。したがって上記の指導に用いるには無理がある。

Formalization of Program Design Knowledge based on Decision Making and Decompose for Tutoring System Tutor System

Hikofumi SUZUKI<sup>1</sup>, Rika SEKIMOTO<sup>2</sup>, Kenji KAIJIRI<sup>2</sup>

<sup>1</sup>Department of Electronics and Computer Science,

Nagano National College of Technology

<sup>2</sup>Faculty of Engineering, Shinshu University

このようなことから、本研究ではプログラム設計・作成指導を効果的に行うためのモデルを考案した。このモデルは2種類の木構造から成るもので、1つは意思決定と詳細化に着目した意思決定木 (Decision tree)[4] で、これは AND/OR 木がベースとなっている。もう1つはプログラム木 (Program Tree)[4] で、こちらはプログラムを表現する。意思決定木をベースに行った意思決定・詳細化の情報を基に、それが反映したプログラム木を生成できる。これにより、意思決定が具体的にどのようにプログラムに反映されたのかを表現できる。

## 2 意思決定木

本研究で提案する意思決定木では、プログラムの設計から作成段階における意思決定と詳細化の情報を表現する。意思決定木は基本的には AND/OR 木である。図1はその様子をあらわしている。ノードには問題 (Problem) が配置されている。問題はさらに幾つかの部分問題 (Sub-Problem) に分割されている。

設計に関する情報

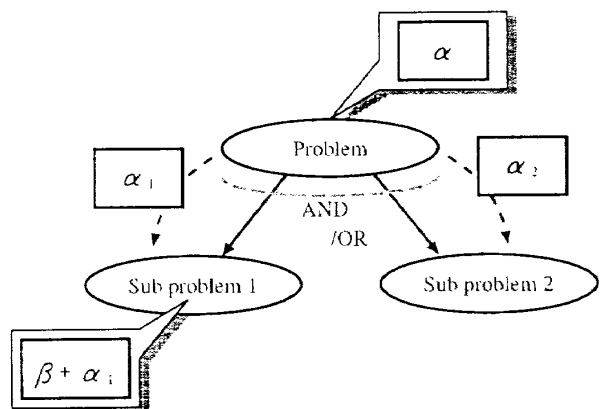
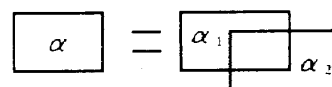


図 1: 意思決定木におけるノードと問題分割

問題はそれぞれが設計情報を保持している。図1では、Problemが $\alpha$ という設計情報を保持している。この $\alpha$ はそれぞれのSub problemに対する設計情報の和( $\alpha_1 + \alpha_2$ )である。Sub problemへ分割されたときに、必要な情報がそれぞれに伝達し利用されることとなる。これにより上位で行った意思決定詳細化の情報が伝播することになる。

このように、問題を部分問題に分割することにより、意思決定と詳細化を表現する。ノードには設計やプログラムに関する情報が保持されており、決定や詳細化に伴い部分問題の情報と合成されてゆく。したがって、意思決定や詳細化が反映したプログラムが作成できる。この時、プログラムに関する情報は主としてプログラム木の形で保持しておく。このため、ある程度意思決定・詳細化を行うことによりプログラムの構造が確定する。*clichés*[3]の形で具体的なプログラムを保持しておくことにより、具体的なプログラムを完成することができる。

また、途中で意思決定を変更した場合に、プログラムがどのように変化するかだけでなく、過去に行った意思決定・詳細化の情報がどこまで再利用できるのか、また、具体的にコード化を行っていた場合、それがどこまで再利用できるのかを示すことができる。

### 3 プログラム木

プログラム木は具体的なプログラムの構造を表現する。

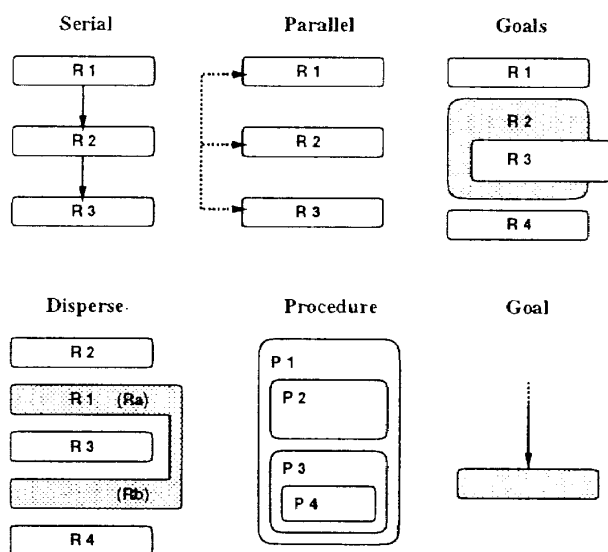


図2: プログラム木の構造

図2がその種類をあらわしている。基本的には次の6つの構造から成っている。

**Serial:** 複数の処理順序が固定している構成を表現する

**Parallel:** 複数の処理の順不同な構成を表現する

**Goals:** ステートメント中にステートメントを定義する構造を表すためのもので、主にループ処理に対応する

**Disperse:** 場所的には分散しているが、実際にはひとまとまりの処理であることを表す

**Procedure:** 手続きを利用する場合に用いる

**Goal:** 具体的なプログラム断片の情報を持つ事を表す

Goalを葉とした抽象プログラム構造から構築される木を、プログラム木(Program tree)[4]と呼ぶ。実際にはユーザーは抽象プログラム構造と知識ベースを基に、具体的なプログラムを構築することになる。

### 4 まとめ

以上の様な2種類の木構造により、より詳細なプログラム作成過程における情報を蓄積できる。このモデルでは、従来の上流工程のモデルでは補えなかった非常にきめ細かい情報まで保持できる。これにより、プログラム作成の指導に有効であると考えられる。

もちろん、モデルでは、プログラム設計からプログラム化までを重点に置いているため、システム全体を通じての開発などには向いていない。しかし、上流工程では補えない部分をカバーするプログラミング能力の指導に効果が期待できる。また上流工程を抜きにしても、プログラム設計とプログラム作成を行うためのシステム作成に貢献できることが期待できる。

### 参考文献

- [1] Conklin, J and Begeman, M.L.: *gIBIS: A Hypertext Tool for Exploratory Policy Discussion*, CSCW '88 Proceedings, ACM, pp.140-152(1988).
- [2] 古宮 誠一: ソフトウェア設計上の意思決定と判断根拠を記録し再利用するためのモデル, 電子情報通信学会技術研究報告 KBSE97-29(1998-01).
- [3] Charles, Rich and Richard, C. Waters: *A Research Overview*, COMPUTER, Vol 21, No.1 (1988).
- [4] 鈴木 彦文: 初心者プログラマーに対するプログラム設計支援と設計知識, 電子情報通信学会技術研究報告 KBSE96-26(1997-01).