

# 意思決定に基づくプログラム設計指導システムの 知識ベースの形式化と具体化について

2R-3

鈴木 彦文 関本 理佳 海尻 賢二

信州大学 工学部

E-mail:csuzuki@cs.shinshu-u.ac.jp

## 1 はじめに

プログラムを作成する段階は(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。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、プログラム設計はどのように行なえば良いのか正確に把握し難く、またどのように考えてゆけば良いのか理解し難い。そこで設計知識に関して、抽象的なプログラム設計及び設計プロセスである意思決定・詳細化プロセスに注目し、これらを形式的に表現する方法を考案する。そしてこれらの知識を基にどのような指導環境が構築できるのかについての考察を行なう。そして具体化に必要な情報を、ゴール/プラン[1]を利用し蓄積する。

これによりプログラムの処理についての理解が深まり、かつプログラムの設計技術向上に役立てることができると期待される。

## 2 抽象プログラム構造

一般的にチャートを用いた設計を考える場合、その方法は言語の構造のみに注目したものが多く、しかしここで用いる抽象プログラム構造は、抽象度の高い状態でのプログラム設計を行なうためのパーツを用意する。この抽象的なプログラム構造は次の6つの構造がある。

**Serial:** 複数の処理の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成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処理を表す。

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)[2]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。

## 3 設計知識

本研究で提案する知識ベースでは、意思決定・詳細化というプロセスを表現する。これは非常に粒の細かい単位での設計プロセスの記述が可能となる。これらの情報を蓄積し、設計と関連付を行なうことにより、高度な設計知識の蓄積と再利用が可能となる。本研究はこの知識を意思決定木(Decision tree)[2]を用いて表現する。

意思決定木は基本的にAND/OR木で構成されている。ノードは問題を表し、子ノードは親ノードの問題を解くために詳細化を行なったものである。

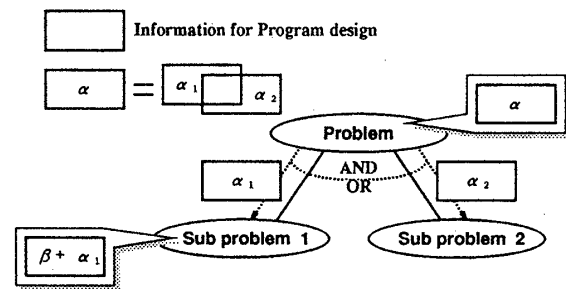


図1: 意思決定木におけるノードと問題分割

## 4 指導とシステム

プログラム設計中には様々な局面が存在する。この局面にはどのようなものがあるのかを捉え、これをどのようにシステムで処理することにより解決するのかを考える必要がある。局面はエディタを中心に学習者から見て能動的な局面と受動的な局面に分けて考える。

Formalization of knowledge based on Decision Making for Program Design Tutor System

Hikofumi SUZUKI, Rika SEKIMOTO, Kenji KAIJIRI  
Faculty of Engineering, Shinshu University

500 wakasato,nagano 380, Japan

#### 4.1 能動的局面

学習者が積極的に行なう動作をまとめると次の局面があると考えられる。

**設計 (設計ホルダの組合せ)** エディタを用いて、設計を抽象的に表したパーツを組み合わせるにより設計を構築してゆく。

**意思決定・詳細化** 指導システムより問題の解法に対するヒント (ガイド) が提供され、学習者が意思決定・詳細化を行なう。その結果の情報を提供されることにより、学習者が現在どの問題を解けばいいのかに集中できる。

**意思決定の変更** 過去に行なった意思決定を変更する。

以上に加え、意思決定・詳細化や診断によって設計情報がもたらされた場合、これをそのまま利用できる。

#### 4.2 受動的局面

学習者に対して指導システムの方から積極的に指導がもたらされる局面は次の様になる。

**ヒントの提供** 意思決定木を基に部分問題の解法に対してのヒント (ガイド) を提供する。問題に対してヒントの出し方により、細かい指導が可能となる。

**設計情報の提供** 部分問題を解く際に必要なホルダや、利用できるホルダを提供する。

**トップ・ダウン設計** 1つの問題に対して、ヒントの提供・設計情報の提供を行ない、学習者に意思決定・詳細化をさせるといった作業を連続的に行なうことにより、1つの問題 (設計) をガイドしながら完成させてゆく。

**診断** 設計が正しいかどうかを意思決定木を基に判定し、診断結果を学習者に返す。

ヒントの提供と設計情報の提供、及びトップ・ダウン設計は、意思決定木上のトップから必ず始めなくてはならないというものではない。また前者二つの機能と診断を用いることによって、特別な処理をしなくともボトムアップ設計に対しての指導も可能となる。

以上の局面に対応するための指導システムを考案すると次のモジュールが必要となる (図 2)。

**エディタ部** エディタはスマートリンクが可能なグラフィックエディタをベースとする。これは構文に則した形でのグラフィックエディタである。

**診断部** エディタ部からの情報と知識ベースを基に学習者の状態を把握し、その結果を意思決定木の状態を操作することにより反映させる。

**指導部** 診断部が操作した意思決定木を元に、実際に指導に必要な情報をエディタ部・GUI部に提供する。

**知識ベース** 設計知識を表現した意思決定木を静的に保管しておく。

**GUI** 指導部と学習者との間の情報伝達を行なう。学習者の入力によって得られた情報のうち、必要なものは指導部を介してエディタ部に伝達される。

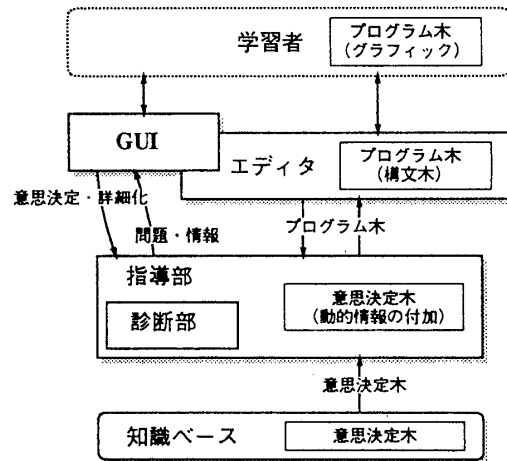


図 2: システムと情報

## 5 まとめ

意思決定や詳細化といったプロセスに注目した知識ベースの構築について述べた。このように設計プロセスを知識化することで、プログラムの意図が明確になる。従来用いられているチャートは基本的には言語に依存したものが多く、これに対して抽象的な意味を持たせた設計であるプログラム木と、設計プロセス自体を詳細に記録できる意思決定木を用いることで、プログラムの設計や構造についての理解を深めることができる。また、この2つのレベルのどちらかでも残しておくことにより、有用なドキュメントとして活用することが可能となる。設計指導システムでは、このような情報の扱いについての教育を促すものである。

## 参考文献

- [1] W.Lewis Johnson, etc : *Understanding and Debugging Novice Programs*, Artificial Intelligence 42 (1990)
- [2] 鈴木 彦文: プログラム設計指導システムにおけるデータベース, 電子情報通信学会技術研究報告 KBSE93-12(1993-07).