

## エージェントを利用したプログラミング診断機構について

2N-7

鷹岡 亮

岡本 敏雄

電気通信大学大学院 情報システム学研究所

## 1 はじめに

本研究の目的は、マルチエージェント指向の計画認識・学習システムを設計／構築するための方法論を提案することである。特に、エージェントを利用した診断機構と学習者のプログラム作成計画及びプログラミング修正過程における支援機構に焦点がおかれている。具体的なシステムとして、UNIXのCシェルプログラミングを対象領域としたプログラミング支援環境を構築している。本稿では、学習者のCシェルプログラムをエージェントを利用して診断する機構について報告する。

## 2 エージェントを利用したプログラム診断機構

プログラミング過程は問題理解、プログラム設計、コーディング、デバッグの4つの段階から構成されると考えられる [1]。一般に、経験の浅いプログラマは、与えられた課題から十分に分析を行わず場当たりにコーディング作業に入ると言われている。このため、コーディング時の支援も重要であるが、その作業の計画段階に相当するプログラム設計仕様書（以下、設計仕様書）を作成する過程も重要であると考えられる。

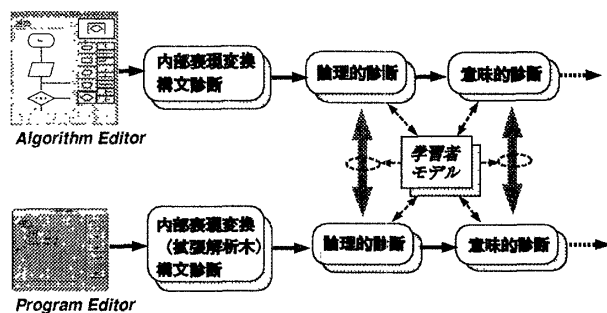


図 1: プログラム診断に関する模式図

そこで本研究では、設計仕様書、ソースプログラム（以下、プログラム）を診断することによって学習者の理解状態を認識する（図 1）。ここで論理的診断とは、機能を考慮しない論理的整合性を診断することである。また、意味的診断とは要求された課題の機能をどの程度満足しているかを

A Study of Program Diagnosis Mechanism with Agents  
by Ryo Takaoka and Toshio Okamoto, Graduate School of Information Systems, the University of Electro-Communications, 1-5-1 Chofugaoka, Chofu-shi, Tokyo, 182 Japan,  
E-mail: {ryo,okamoto}@ai.is.uec.ac.jp

診断することである。このプログラム診断では、‘対象領域に関する診断’と‘プログラム診断’の二つの診断系が必要である。本章では、この二つの診断系について述べる。

## 2.1 対象領域に関する診断

対象領域に関する診断では、UNIX コマンドの使用方法や入力されたCシェルプログラミングの構文を診断する。また、プログラムは命令コードや処理対象の定数、変数をノードに、ノード間の関係（順次処理、条件成立、条件式など）をリンクとした拡張解析木に変換される。これらの診断に伴い、UNIX の概念、UNIX コマンド、C シェルプログラミングの構文規則に関する学習者モデルを構築する。これらの診断を行うために、UNIX の概念知識、コマンドの使用方法に関する知識、C シェルプログラミングの構文規則と各々に対するバグ知識が用意されている。

## 2.2 プログラム診断

プログラム診断では、学習者の設計仕様書とプログラムに対して論理的診断と意味的診断を行う。そして、設計仕様書とプログラムに対する学習者モデルを構築する。学習者のプログラムを診断するために、問題解決概念知識（課題解決で要求される目標構造を理解するための知識）、アルゴリズム処理知識（処理技法知識の系列で表現された知識。1つのアルゴリズムに対して処理技法の選択、処理順序の入れ換えが存在する）、処理技法知識（基本操作知識と構造を持った命令の組み合わせられた知識）、基本操作知識（1行の命令コードに対応する詳細なパターンの知識）、バグ知識（論理的バグ、意味的バグの知識）を具備している。

学習者によって記述されたプログラムの1行は、拡張解析木の表現でワーキングメモリに入力される。論理的診断は学習者のプログラムは基本操作知識と処理技法知識を利用してボトムアップに実行される。具体的には、拡張解析木で表現された知識と学習者のプログラムとのパターンマッチングを行う。結果として、上位レイヤに対して複数の説明可能な仮説が生成される。論理的診断において、学習者のプログラムの制御構造を認識する。また、意味的診断がトップダウンに行われ、結果として各々の仮説が検証される。この時、論理的バグ知識と意味的バグ知識を利用して学習者の誤りを診断し、バグ誤概念ネットワークを用いて学習者が有しているであろう誤概念を診断する。

本研究では、上述した二つの診断系をエージェントモデ

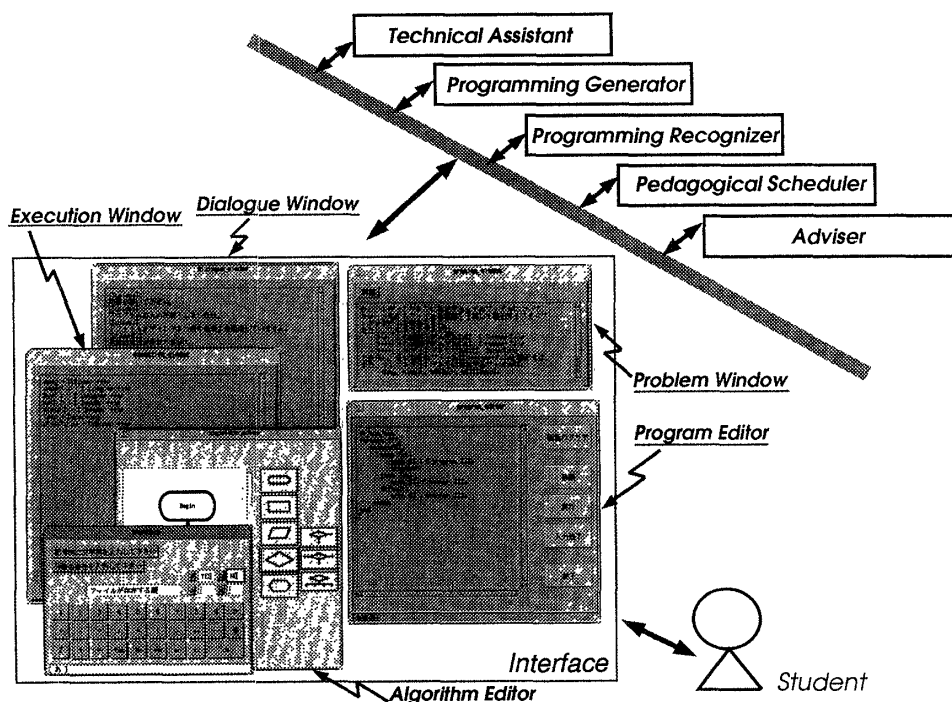


図 2: プログラミング支援環境の構成

ルを用いて実装する [2, 3]。つまり、対象領域に関する診断系では、UNIX の概念、UNIX コマンド、C シェルプログラミングのシンタックスなどに関する学習者の理解状態を信念としてエージェントに所有させる。また、プログラム診断系では、プログラム診断中の仮説や結果、学習者の誤概念などを信念としてエージェントに所有させる。

### 3 プログラミング支援環境

本研究では、教育的な機能を有するエージェントが組織化されることによって、1つの学習環境が作成され、その学習環境の中で教授・学習過程が展開される。現在の環境では、前章で述べた二つのエージェントを含んだ五つのエージェントによって組織化される (Technical Assistant(TA): 学習環境で使用されるツール、UNIX や C シェルプログラミングの概念や構文規則に関する専門家、Programming Recognizer(PR): プログラムの論理的/意味的診断、学習者モデル構築の専門家、Programming Generator(PG): プログラム作成の専門家、Pedagogical Scheduler(PS): 教育学(教授)に関する専門家、AdViser(AV): 円滑な相互作用を実現する専門家)。これらのエージェントは、他のエージェントとコミットメントを結ぶことによって協調した振る舞いを示す。そして、この振る舞いの結果がプログラミング支援環境の機能を果たすことになる。

学習者に対しては、図 2 に示されているように、プログラムエディタ、アルゴリズムエディタ、実行ウィンドウ、問題ウィンドウ、対話ウィンドウの五種類のウィンドウが与えられる。学習者とシステムとの対話は対話ウィンドウ上

で展開され、学習中の対話履歴を見ることが可能になっている。対話ウィンドウにおける学習者の入力には語彙テンプレートから選択することにより行われる。問題ウィンドウ上に問題が提示されると、学習者はアルゴリズムエディタを利用することにより設計仕様書を作成し、プログラムエディタを利用することによりプログラムを記述する。プログラムエディタ上では、診断ボタンをクリックすることによってシステムの診断が行われ、実行ボタンをクリックすることにより学習者のプログラムが実行される。この機能は学習者の理解状態に応じて、利用可能/不可能が決定する。

### 4 おわりに

本稿では、学習者の C シェルプログラムをエージェントを利用して診断する機構について述べた。また、この診断機構を有するプログラミング支援環境の構成を説明した。今後の課題として、要求仕様理解レベルの診断機構を設計・実装、複数のプログラム診断を行うエージェントが実装されている環境で協調することによりプログラム診断を展開していく機構の実現等があげられる。

### 参考文献

- [1] 河村一樹, “ソフトウェア工学入門”, 近代科学社, 1995.
- [2] Y. Shoham, “Agent-oriented Programming”, *Artificial Intelligence*, vol.60, pp.51-92, 1993.
- [3] 鷹岡亮, 岡本敏雄, “知的プログラミング支援環境におけるエージェントモデルの構成について”, 1996 年度人工知能学会全国大会, pp.469-472, 1996.