

知的プログラミング支援環境におけるエージェントの協調について

7K-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科

1 はじめに

近年、Agent アーキテクチャの考え方を取り入れた様々なシステムが開発されている。そこでは、独立した各々の Agent がタスクを解決するために他の Agent と通信を介して協調することにより自分自身のゴールを解決していく [2]。従来、知的教育システムは機能分解されたモジュールから構成されていたため、その振舞いが暗黙の順序性に従っていた。しかし、上述した Agent を組み込むことによりこの問題点が解消されると考えられる。本研究では、Agent アーキテクチャに基づいた教育システム構築の方法論を提案する。具体的なシステムとして、UNIX のシェルプログラミングを対象領域とした、知的プログラミング支援環境を構築する。本稿では、知的プログラミング支援環境の構成、支援環境で生じる Agent 間の協調的な振る舞いを述べる。

2 知的プログラミング支援環境

2.1 協同的作業に必要な Agent の機能

人間の協調を階層的に捉えるという立場から「協調の次元階層モデル」が提案されている [1]。そこでは、Copresence, Awareness, Communication, Collaboration の 4 つの層で表現されている。我々はこのモデルを複数の人々で遂行されるタスクという観点から見直し、上位層に “Cooperative Task” と “Collaborative Task” を付加した。Agent 間でこのような階層構造を実現するために、各々の階層で対応する手段が必要となる。Copresence 層は Agent の立場からは必要とされない。Awareness 層は Agent の有する他 Agent の信念を心的状態として表現することにより実現される。Communication 層は Agent 間の通信を行うための通信プリミティブを用意することによって実現される。そして、Collaboration 層は Agent 間の通信プロトコルを設定することにより実現される。Agent 間の通信プロトコルは、目的に応じた通信プリミティブと各 Agent における処理の系列で表現できる。本研究では、教育タスクを Cooperative Task と考え、教育タスクの各々の機能をさらに Collaborative Task として行う Agent に分割したシステムを目指している。本稿では、前者の Cooperative Task に焦点をあてる。

Collaboration among agents in Intelligent Programming Supporting Environment 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

Problem

Write a script program to show what kind of files are contained in the current directory using the file name form. Do this for all files. In this script,
 + If the extension of file name is ".c", the result display should be "C Source File".
 + If the extension of file name is ".h", the result display should be "C Header File".
 + If the file is directory, the result display should be "Directory".
 + The result display about other files should be "Unknown file".

Student's Answer Program Including some errors.

```
1 #!/bin/csh
2 foreach i (*)
3   switch ($i)
4     case *.c :
5       echo $i : C program file
6       breaksw
7     case *.h :
8       echo $i : C header file
9     default:
10      echo $i : Unknown file
11   endsw
12 end
```

図 1: 問題と学習者の回答例

Cooperative Task において協調を表現するために次の 5 つの通信プリミティブを用意した; **inform**(情報通知)、**request**(動作, 処理, 提案の要求)、**propose**(提案)、**agree**(同意/非同意)、**negotiate**(交渉要求)。また、次の 3 つの通信プロトコルが組み込まれている; **Information Protocol**(情報通知のプロトコル)、**Request Protocol**(情報要求のプロトコル)、**Negotiation Protocol**(交渉のプロトコル)。

2.2 システムの構成

本研究では、知的プログラミング支援環境を実現するために 5 つの機能を Agent として組み込んでいる。**Technical Assitant(TA)** は、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に関する専門家である。そして、ツールや UNIX コマンドの使用方法を助言する機能、学習者の入力した UNIX コマンドや C シェルプログラミングの構文を診断する機能等が具備されている。**Programming Generator(PG)** は、与えられた課題(要求仕様)からプログラムを作成する。そのために要求仕様から課題アルリズムを生成する機能、課題アルリズムからソースプログラムを生成する機能等をもっている。**Student Recognizer(SR)** は、学習者プログラムの論理的診断と意味的診断を行い、学習者モデルを構築する。ここで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診断することであり、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。**Pedagogical Scheduler(PS)** は、学習者の理解状

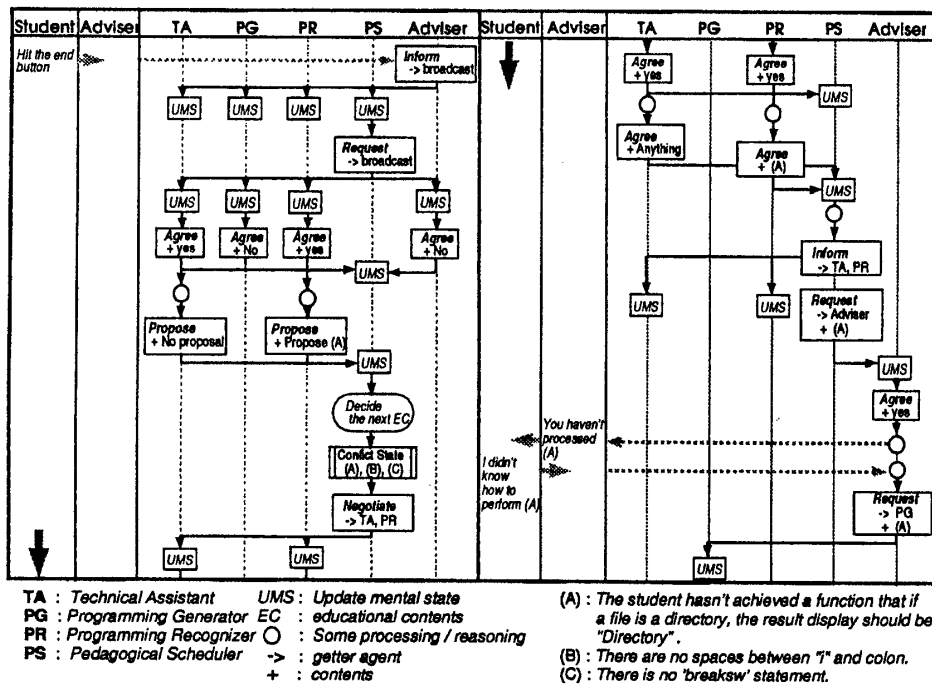


図 2: Agent 間の協調フロー例

態に対して適切な教授戦略モードと教授内容を決定する。Adviser は、学習者との円滑な相互作用を実現する。そのために、教授内容に対応した説明文を作成する機能、診断内容を提示する機能等を具備している。

2.3 教育タスクにおける Agent 間の協調例

本システムでは、各々の Agent が能動的に振る舞い、設定されている通信プロトコルの基で他の Agent とメッセージを通信し合うことにより、システムの振る舞いが決定される仕組みになっている。図 1 は C シェルプログラムの課題とある学習者の回答を示している。学習者がプログラム作成終了ボタンを押した時点で、学習者の誤りは次の 3 つであると推定される; (A) ファイルがディレクトリの時の処理が行われていない (意味的バグ), (B) 8 行目の i とセミコロンに空白がない (Syntax レベルのバグ), (C) 7 行目の case に対応する breaksw の宣言がない (Syntax レベルのバグ)。

図 2 は Agent 間の協調過程を示している。現在の教授戦略モードは「コーチ的支援」である。Adviser は「回答終了」を検知し、その情報を全ての Agent に送信する。全ての Agent は「学習者が回答を終了した」という信念をもつ。この時点で、PS は既に (B)、(C) の情報を TA から保持している。PS は全ての Agent から学習者の誤り、次の教授内容を要求する。TA と PR だけが Agree Message によって「同意」を示している。PR は (A) という意味的バグを Propose Message によって、TA は「他の誤りがない」ことを各々提案している。設定された時間が経過した時、PS は次の教授内容を決定しようと試みる。しかし妥当な教授内容を選択できないため、教授内容候補が競合状態を形成

する。PS は (A)、(B)、(C) の内容を該当 Agent に交渉してもらおう。TA と PR はこの Negotiate Message に対して同意する。TA は学習者がプログラムを 1 回も実行していないため、どの内容でもかまわないという意見を Agree Message によって送信する。SR は (A) に同意する。PS は SR の意見を受け入れ、教授内容を (A) に決定し、交渉に参加した人に対して決定通知を送付する。さらに、Adviser に対して (A) を助言するように要求する。Adviser は現在の教授戦略モードを確認後、「あなたは (A) に関して処理をしていない」という内容を提示する。それに対して、学習者は「(A) を成し遂げる方法が分からない」ことを Adviser に述べている。この時、Adviser は PG からその部分の正解プログラムを要求する。Adviser は PS との Request Protocol の中で新たな Request Protocol を用いて PG と相互作用を行っている。Adviser のコミュニケーションモジュールはこれらのプロトコルの進行状況を管理している。

3 おわりに

本稿では、Agent モデルを利用した知的プログラミング支援環境の構成について言及した。Agent モデルを組み込むことによって各々エージェントが能動的に振る舞い、学習者に対するシステムの振る舞いが決定された。今後、通信プロトコルや通信プリミティブの洗練化が必要である。

参考文献

[1] 岡田謙一, 松下温: 協調の次元階層モデルとグループウェアへの適用, 情報処理学会研究報告, pp.16-23 (1993).
 [2] Shoham, Y.: Agent-oriented Programming, Artificial Intelligence, Vol.60, pp.51-92 (1993).