

ネットワーク型演習支援システムにおけるコースウェア アドバイスの生成について

福島 学^{*} 浮貝 雅裕^{**} 菅原 研次^{**}
^{*}富士ゼロックス ^{**}千葉工業大学

演習授業において学習者が効果的な演習を行うためには、演習支援システムが学習者に対して十分なアドバイスを提供することが重要である。しかし、アドバイス生成のためには学生の学習履歴、教材内容及びコースウェアガイドに関する膨大な情報が必要であり、またアドバイスを生成する機構も複雑で大きなものとなる。このため、従来のスタンドアロン型のCAI環境では、情報管理の面からもマシンパフォーマンスの面からも負担が大きく、十分なアドバイスの提示が行なえない。こうした点からもネットワーク環境を用いてアドバイス生成に関する機能を効果的に分散させ、それらの機能モジュールの協調作業により学習者に対する確かなアドバイスを生成することが必要である。本報告ではネットワーク型演習支援システムにおけるアドバイス生成において必要な機能のうち、特にコースアウトラインに関するアドバイス生成に焦点を絞って述べる。

A Courseware Advice Mechanism in Network Based CAI

Manabu Fukushima^{*} Mashiro Ukigai^{**} Kenji Sugawara^{**}
^{*}Fuji Xerox ^{**}Chiba Institute of Technology

^{*}KSP/R&D BUSINESS PARK BLDG. 100-1, Sakado, Takatsu, Kawasaki-shi, Kanagawa-ken, 213 JAPAN
^{**}2-17-1, Tsudanuma, Narashino-shi, Chiba-ken, 275 JAPAN

The technological advances have increased instructional loads and brought difficulties in teaching a large class. Therefore computer assisted learning systems are receiving considerable attention in computer engineering education. In order to guide students through the learning course of programming laboratories, many pieces of competent teacher's advice according to degree of individual students are required. However various kinds of and huge information, such as student's learning history, content of materials and learning course, and large scale software system is necessary for providing such advice. It will be too heavy for conventional stand alone type CAI systems to satisfy these requirements of large class size education. Consequently, network-based supporting system for programming laboratory is required. In the network environment, many modules are distributed and work cooperatively to guide student to right goal of the learning course.

This paper presents a providing course outline advice mechanism in network based CAI.

1. はじめに

科学技術や社会システムの急速な進展に伴い大学や企業などにおける高度教育や生涯教育の重要性が増してきている。このための効果的な教育システムを実現する手法の一つとして計算機支援型の教育システムの研究が進められてきた。効果的な教育環境を実現するためには、教材表示機能、コースアウトラインガイド機能、学習履歴管理機能、アドバイス生成機能、成績管理機能、オーサリング機能などの複雑な支援機能が必要になる。更にこれらの機能モジュールが相互に協調して、学生の個性に合わせて学習意欲を維持し、学習効果を向上させる仕組みが必要になる。この要請に応えるため教育支援の計算機システムとしては、スタンドアロン型のパーソナルコンピュータからコンピュータネットワーク上の統合型教育支援環境への移行の必要性が指摘されている。[1][2][3]

演習授業において効果的な演習を行うために必要な機能の一つとして、演習支援システムが学習者に対して十分なアドバイスの提供を行う事により学習者が効率よく学習を進める事を支援する機能がある。[4]

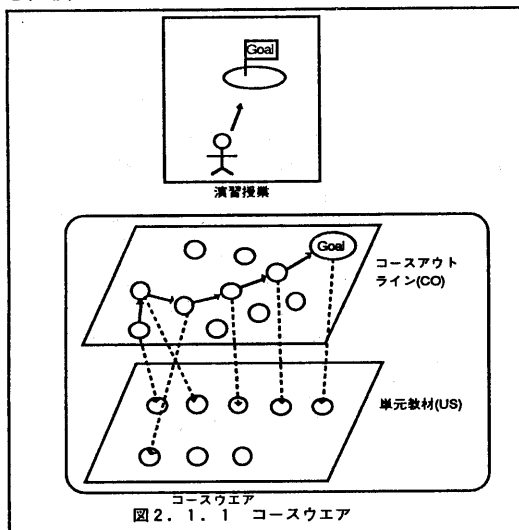
しかし、十分なアドバイスを提供するためにはアドバイス生成のために膨大な情報が必要であり、またアドバイスを生成する機構も大きなものとなる。

このため、スタンドアロン型のマシンだけでは、情報管理の面からもマシンパフォーマンスの面からも負担が大きい。こうした点からもネットワーク環境を用いて支援モジュールの分散協調処理を実現することにより、演習用計算機システムの負担を分散させる事が重要である。[5][6]

本報告ではネットワーク型演習支援システムにおけるアドバイス生成において必要な機能について述べる。本稿では特にコースアウトラインモードに於ける協調的アドバイス生成について述べる。

2. コースウェア

2. 1. コースウェアモデル



コースウェアは、どのように学習を進めて行くかが記述されたコースアウトライン(CO)と、実際の学習に用いられるテキストに関して記述された単元教材 (US) によって構成されている。(図2.1.1)

コースウェアモデル: CWM

$CWM = \langle CO, US \rangle$

2. 2. コースアウトラインモデル

演習授業では、ある事柄を習得する事を目的とした学習がなされる。この際、何をどのように学習させるかを設定したものがコースアウトラインである[7]。このコースアウトラインによって学習者は目的とした事柄を学習するために必要な習得を進めて行く事ができる。

しかし、学習者はそれぞれ学習能力が異なるため、学習進度や理解に必要な説明なども異なる。このため、コースアウトラインは各学習者に対応したものが提供される必要がある。

CO: コースアウトライン

$CO = \langle CN, REL, PROP \rangle$

CN: コースノード集合

REL: ラベル付きアーク集合

PROP: ノードのラベル集合

$REL = \{ (s, d, l) \mid \begin{array}{l} s \in CN \text{ where } s \text{ is a source node,} \\ d \in CN \text{ where } d \text{ is a destination node,} \\ l \in L \text{ where } l \text{ is a label} \end{array} \}$

$L = \{ \text{part-of, includes, entry_to, exit_to, next, previous, basic, grade_up, grade_down, application, basic_next, basic_previous, application_next, application_previous} \}$

$PROP = \{ \text{node_id, course_id, node_class, title, abstract, grade, unit_pointer} \}$

ラベルのpart-ofおよびincludesは、ノード間の階層を表し、ノード間の推移では、下位ノードへまたは上位ノードへの推移を表すentry_toおよびexit_toがある。同一レベルでの推移としては、次のノードへの推移を表すnext、前のノードへの推移を表すpreviousがあり、さらに、基礎学習として学習時に用いられるbasic_nextおよびbasic_previous、応用学習として学習時に用いられるapplication_nextおよびapplication_previousがある。また、grade_upおよびgrade_downはレベル間の変更を表す。

PROPでは、コースノード特有の情報を表す、node_id, course_id, node_class, title, abstractがある。

node_classはノードの特性を表しており、リーフノードであるかまたは下位ノードを持っているかを示す。unit_pointerは単元教材であるUnitNodeへのポインタである。

2. 3. 単元教材 (ユニット)

US: ユニットセット

$US = \{ un \mid un \text{ は単元教材のノード} \}$

$un = \langle \text{unit_id, TEXT, EXEMPLAR, EXERCISE, TEST} \rangle$

TEXT = {text_id | text_idはテキストフレームへのポインタ}

EXEMPLAR = {example_id | example_idは例題フレームへのポインタ}

EXERCISE = {exercise_id | exercise_idは練習問題フレームへのポインタ}

ンタ]

TEST={(test_id, answersheet_id) | test_idはテストフレームへのポイント answersheet_idは回答用紙へのポイント}

3. 学習者個別情報

3. 1. 学習者フレーム

学習履歴は各学習者毎に設定される学習者フレームに獲得されていく。学習者フレームは、1) 学習者の個人情報、2) 学習結果情報、3) 学習履歴情報、4) 現在の学習状況情報、によって構成されている。

学習者フレームの学習履歴情報は学習者が学習を行うに従いシステムにより情報収集が行われる。ここで収集された情報は教員用ワークステーションのAnalysisAssistが教員と協動的に処理を進める事により分析を行い、学習結果情報を生成する。

SF=<PD, CD, RD, HD>

PD=<student_id, Name, Department, Year, Sex>

CD=<CurrentNode_id, PN_ID, MODE>

PN_ID=<PreviousNode_id>

MODE=<CourseOutLine, Unit, OffLine>

PDは各学習者の個人情報を表す。

CDは現在の学習状況を表し、現在学習しているノード、直前に学習していたノード、およびそのモードを表す。

RDは各学習者の学習結果に対する評価を表す。

HDは各学習者の学習履歴情報を表す。

3. 2. 学習履歴

学習履歴は学習者の学習行為に従ってシーケンシャルに記録されていく。この際記録されるものは、学習した順番を表すi、どのノードを学習したかを表すn、どのような理由で学習したかを表すr、学習結果である答案用紙、である。

HD={(i,n,r,a) | i ∈ 自然数, n ∈ CN, r ∈ REL, a ∈ ANSWER_SHEET}
ANSWER_SHEET={回答用紙}

rはコースアウトラインで定義されているRELによって表現され、どのような関係で前のノードから移ってきたかを表す。これにより、どのような理由によってそのノードを学習したかが記録される。

この学習履歴は、学習者の学習時に獲得され、学習者フレームに追加されていく。

3. 3. 学習結果評価

RD=<TR, GR, UR>

TR={TotalResult | TotalResult ∈ Result}

GR={({Group_ID, GroupResult) | GroupResult ∈ Result}

UR={({Unit_ID, UnitNodeResult) | UnitNodeResult ∈ Result}

Result={評価基準値の集合、例えば0から100までの整数}

RDは評価情報を表す。TotalResultは現在までの総合評価を示し、GroupResultは複数ノードによって評価されるある単位の評価を示し、UnitNodeResultは各単元ノードの評価を示す。

学習結果情報は学習履歴情報から分析、評価され生成される。(図3.3.1)

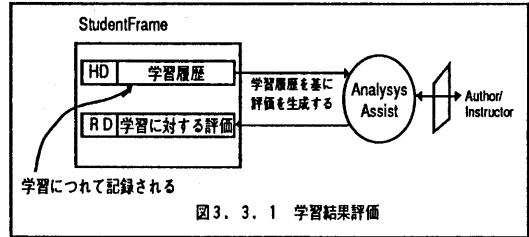


図3. 3. 1 学習結果評価

学習者フレームに記録される学習履歴および学習結果評価から、今まで学習してきたノードに対する、1) 何を理解しているか、2) 何を理解していないか、を判定し、コースウェアに対応した、1) 理解マップ、2) 誤答履歴、を生成し、アドバイス生成の際の情報とする。この各マップにおいては、ノード間の関係から、推定理解度および推定理解不足度を生成し、未履修ノードに対する評価を出しマップを補う。さらに、学習履歴および学習結果評価から、学習履歴詳細分析情報を生成する事により、各マップだけでは判断できない場合の判断情報とする。

4. 学習結果分析

4. 1. 理解マップ

理解マップ=<(node_id,node_understand),(group_id,group_understand)>

node_understand={ResultDegree, PredictDegree, Unknown}

ResultDegree={A, B, C, D}

PredictDegree={A, B, C, D}

group_understand={ResultDegree, PredictDegree, Unknown}

Unknownは判断するための情報が無いものであり、関係する全ノードに関する情報が無いものを示す。

PredictDegreeは未学習ノードに対する評価であり、関係する全ノードにおける評価とそのノードにおける新規学習事項により推定される評価である。

これにより、学習者が現在何を理解しており、今後学習するノードが理解できるかどうかを判断する材料となる。この情報は今後どのノードを学習するべきかに対するアドバイスを生成の際に用いられる。

4. 2. 誤答履歴

誤答履歴=<(node_id, node_miss),(group_id, group_miss)>

node_miss=<TypeOfMiss, Percentage>

TypeOfMiss={ResultMiss, PredictMiss}

ResultMiss={TypeMiss, SyntaxError, Blank, Unknown}

PredictMiss={TypeMiss, SyntaxError, Blank, Unknown}

group_miss=<TypeOfMiss, Percentage>

TypeMissはスペルミスを意味し、SyntaxErrorはプログラム言語におけるカッコの不一致等の文法ミスを意味し、Blankは答案欄が空白であるため、その理由が答案から判断できないものを意味し、Unknownは前述以外の誤答を意味する。

Percentageは全体に対してどのくらいの比率で誤答を出したかを示し、これにより、全体に対してどのタイプの誤答をどの程度出したかがわかる。

誤答はミスの種類によって比重がつけられており、致命的なミスから、致命的でないミスに分けられている。

PredictMissは未学習ノードに対する推定ミスであり、関係する全ノードにおけるミスとそのノードにおける新規学習事項により推定されるミスである。

これにより、学習者が現在何を理解しておらず、今後学習するノードが理解できるかどうかを判断する材料となる。この情報は今後どのノードを学習するべきかに対するアドバイスを生成する際に用いられる。

理解マップにおけるPredictDegreeおよび誤答履歴におけるPredictMissは、デフォルトでは、推定すべきノードの基礎ノードの情報により推定する。この際、既知情報と、未知情報の比率によっては、推定不能を表すUnknownとする。特に各ノードにおいて推定するための条件を記述する事により、デフォルトよりも細かい推定を行なう事が可能となる。

4. 3. 学習履歴詳細分析情報

学習履歴情報から、理解度マップ、誤答履歴が生成され、各学習者の理解度、理解不足度の判断のための情報となるが、さらに細かい判断を行なうための情報として学習履歴から学習履歴詳細分析情報を生成する。

学習履歴情報は学習行為の順に記録される。ノード間の推移を大別すると、1)コースアウトラインに沿ってノードを推移する、2)グレードを変更する、3)ノードの基礎ノードを学習する、4)ノードの応用ノードを学習する、5)既学習ノードを再学習する、となる。

学習履歴詳細分析情報では、これらのノード間の推移によって学習履歴情報を分析する(図4.3.1)。これにより、1)基礎ノードの学習を経て学習した場合、2)コースアウトラインに沿って学習した場合、3)応用ノードの学習を経て学習した場合、4)学習の回を重ねるごとに評価が、a)上がる場合、b)下がる場合、c)変化しない場合、d)安定しない場合の判断を行なう事ができ、同じ評価に対してもさらに細かく理解できているか、またはできていないか、理解できるか、理解できないかの判定を行う。

このように学習履歴詳細分析情報を生成する事により、答案以外の理解度、理解不足度を推定するのに有効な情報

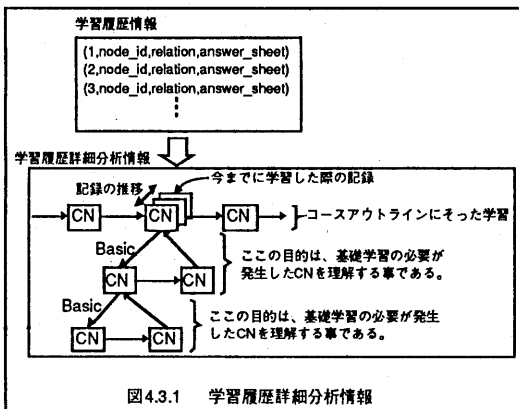


図4.3.1 学習履歴詳細分析情報

となり、より正確な推定を行う際の材料となる。

5. アドバイス

5. 1. アドバイス生成メカニズム

アドバイスには、コースアウトラインモードにおける対話の際に出されるものと、単元演習モードにおける対話の際に出されるものがある。コースアウトラインモードにおける対話の目的は次に学習する単元を決定するものであり、システムはアドバイスによって学習者の単元決定を支援する。単元演習モードにおける対話の目的は学習している単元における理解を促進する事であり、システムはアドバイスによって学習者の理解を支援する。

アドバイス生成の際に情報として利用できるものとして、1)総合評価、2)単位評価、3)単元ノード評価、4)理解マップ、5)誤答履歴、6)学習履歴詳細分析情報、がある。単位評価とは、ある複数ノードの学習によって学習される一つの学習対象に対する評価である。

これらの情報は、演習授業終了時に回収される更新された学習者フレームを分析する事により生成される。ここで、分析され生成された情報は演習開始時に必要な分だけ学生ワークステーションに配布され、学生ワークステーションでのアドバイス生成の際に利用される(図5.1.1)。

各マップはサーバのStudentFrameManagerと教員ワークステーションのAnalysisAssistおよび教員との協調処理により、記録の分析の結果として生成され、サーバのRecordManagerによって管理される(図5.1.2)。

アドバイスはサーバのAdviceManagerによって生成される。この際AdviceManagerは各マップおよび詳細情報を管理しているサーバのRecordManagerおよび教員ワークステーションのAnalysisAssistと協調処理を行う(図5.1.3)。

5. 2. コースアウトラインアドバイス

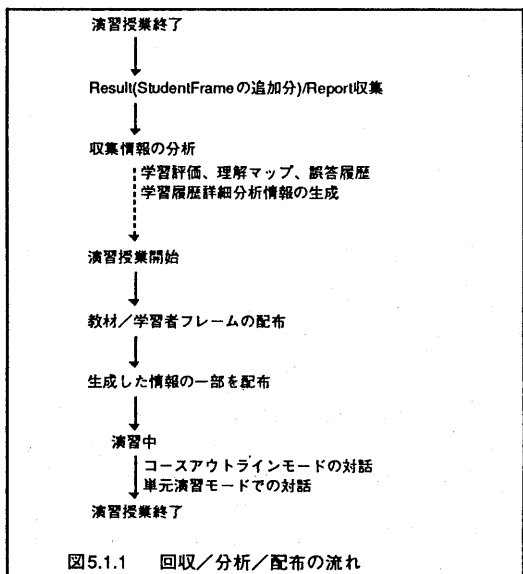


図5.1.1 回収/分析/配布の流れ

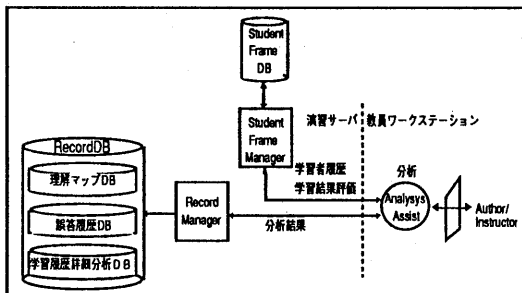


図5. 1. 2 各マップの生成 /更新

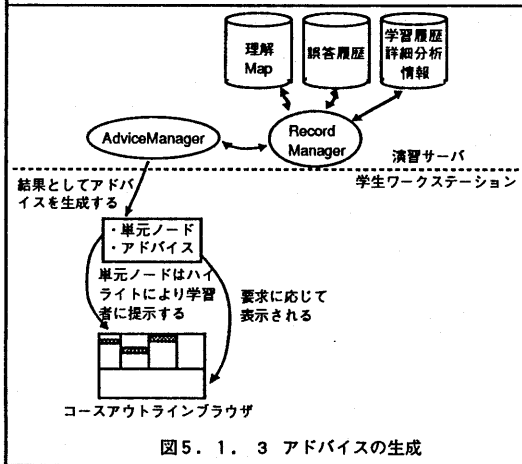


図5. 1. 3 アドバイスの生成

コースアウトラインモードにおけるアドバイスとして生成されるものは以下の通りである。

- 次に選択すべき単元
- 進度に関するアドバイス
- 学習対象に関するアドバイス
- 成績に関するアドバイス

次に選択すべき単元はシステムからコースアウトラインブラウザを通じて学習者に対して提示されるが、それ以外のアドバイスは学習者からの要求に対応して提示される。

学生用ワークステーションにはローカルにアドバイスを生成するための情報として、コースアウトラインマネージャにコースアウトライン、単元教材、理解マップ、誤答履歴、学習者フレームが保管される。これらは、演習開始時に演習サーバから配布される(図5.2.1)。

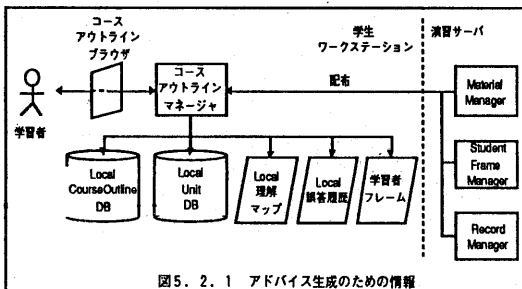


図5. 2. 1 アドバイス生成のための情報

LocalCourseOutlineDBには演習サーバのMaterialManagerが管理するコースアウトラインが配布される。

LocalUnitDBには、現在学習中の単元教材と、演習開始時に推測される演習時間内に進むであろう範囲の単元教材が演習授業開始時に配布される。LocalUnitDBの内容は演習実施時の進捗や理解度によって管理する内容が更新される。

Local理解マップおよびLocal誤答履歴には、LocalUnitDBで管理される単元教材に対応した部分のマップと、それぞれの単元ノードに対応した基礎ノードに対応した部分のマップが入る。これらは、LocalUnitDBの更新に対応して更新される。

さらに学習者に対応した学習者フレームが配布される。コースアウトラインモードでの対話時に必要なアドバイスは、まずこれらの、配布された情報を基に生成される。

5. 3. コースアウトラインアドバイス生成

コースアウトラインマネージャが保有している情報は、全情報に対して必要であると推定された部分だけである。したがって、コースアウトラインマネージャが保有している情報の範囲外に関するアドバイス要求が発生した場合、コースアウトラインマネージャはアドバイス生成に際して必要な情報が配布された情報に無いと判断し、演習サーバのアドバイスマネージャにアドバイスを要求する。アドバイス要求の際にコースアウトラインマネージャは何に対するアドバイスを要求するのかと現在配布を受けた情報が何であるかを送る。これを受けてアドバイスマネージャは配布した情報以外の情報を中心にアドバイスの生成を行なう。これは、配布された情報により生成可能なアドバイスより高次のアドバイスを生成するためであり、主に理解マップおよび誤答履歴のうち配布しなかった情報により配布情報を補い、さらに学習履歴詳細分析情報を用いてアドバイスの生成を行なう。

この際、学生ワークステーションのコースアウトラインマネージャからのアドバイス要求を受けて演習サーバのアドバイスマネージャは、1)コースアウトライン、単元教材を管理しているMaterialManager、2)学習者フレームを管理しているStudentManager、3)理解マップ、誤答履歴、学習履歴詳細分析情報を管理しているRecordManager、と協調処理を行ない、アドバイスを生成する(図5.3.1)。

協調処理の結果として出されるのはアドバイスだけでなく、アドバイスの生成に必要な理解マップおよび誤答履歴の一部、アドバイスによって選択される単元教材がある。これらは、学生ワークステーションで不足している不足情報である。

処理の結果としてアドバイスが生成された段階で、同じく生成された不足情報と共に学生ワークステーションに送付する。コースアウトラインマネージャはこの不足情報により理解マップをはじめとする諸情報を更新し、次回から同様のアドバイス要求が発生した場合、ローカルでアドバイス要求に対する対応を可能とする(図5.3.2)。

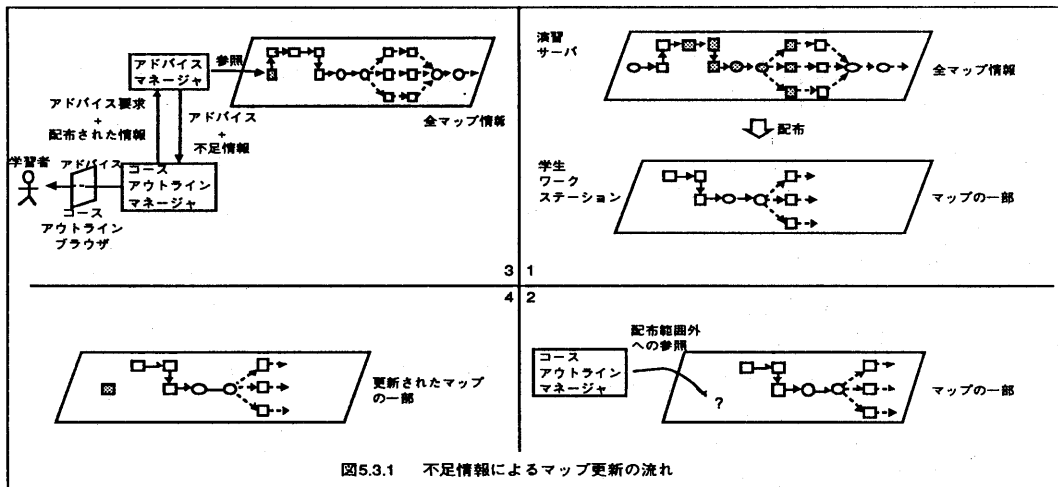


図5.3.1 不足情報によるマップ更新の流れ

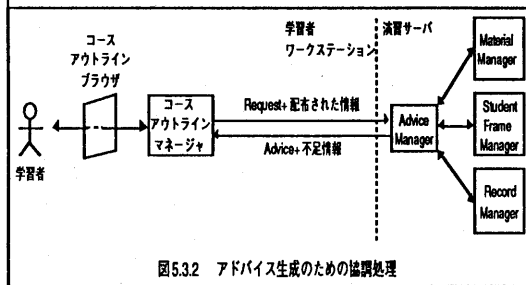


図5.3.2 アドバイス生成のための協議処理

6. アドバイス生成の例

ここでは次にどのノードを選択するべきかに関するアドバイスの生成例について述べる。

単元演習を終了すると、演習結果を基に理解マップおよび誤答履歴を更新する。これを受けて学生ワークステーション

のコースアウトラインマネージャは予め推定されていた推定結果と、実際の演習結果の比較を行なう。比較の結果が推定よりもよい場合は、応用ノードおよび上級ノードに対する推定を行ない、推定よりも悪かった場合は、基礎ノードおよび初級ノードに対する推定を行なう。推定は未学習ノードにおける推定評価と既学習ノードにおける評価結果を基に行なう。

この推定の結果として、理解マップおよび誤答履歴のそれぞれのマップにおいて次にどのノードを学習すればいいかの候補を選出する。候補の選出を受けて、両マップのうち、最も高い得点を得たノードを最終候補とし、学習者に対して選出したノードに進むようアドバイスを行なう(図6.1.1)。

この他にも、1)進度に関するアドバイス、2)学習対象に関するアドバイス、3)成績に関するアドバイス、が生成さ

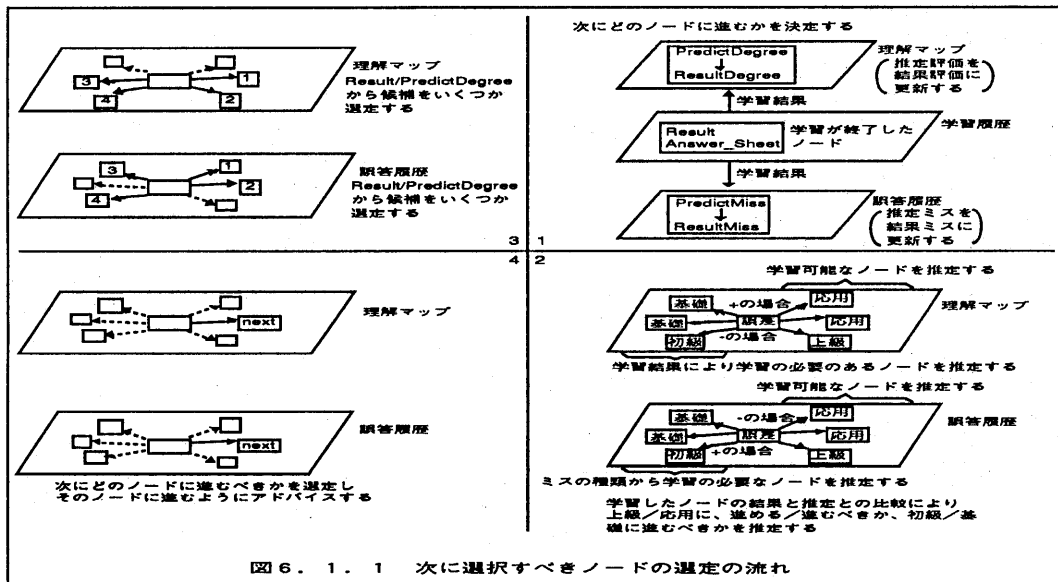


図6.1.1 次に選択すべきノードの選定の流れ

れる。

進度に関するアドバイスとしては、1)なぜそのノードを学習するよう指示されたのか、2)グレードの異なる学習をするためにはどうすればいいのか、3)早くに演習を終了するためにはどうすればいいのか、4)じっくり学習するためには何を選択すればいいのか、といったアドバイス要求に対応する。

学習対象に関するアドバイスとしては、1)このノードでは何を学習するのか、2)このノードは何のために学習するのか、3)このノードでは何を学習しておかなければならないのか、4)このノードを学習して何が理解できるのか、といったアドバイス要求に対応する。

成績に関するアドバイスとしては、1)成績を上げるためには何を選択すればいいのか、2)現在の学習者のレベルはどのくらいか、3)どうしてこの成績になったのか、というアドバイス要求に対応する。

これらのアドバイスを生成するための知識は、ルール言語によってそれぞれ記述されている。

システムと学習者との対話と演習結果を通じて、学習者の理解度、理解不足度、進度が補足/推定され、利用される可能性の低くなった情報を学習者ワークステーションから削除する。これにより学習者ワークステーションの資源の開放を行ない、マシンの負担の軽減を計る。

また、アドバイス生成に際して、アドバイス要求の返信として送付される不足情報により、今後利用される可能性のある新規情報の配布を受け、随時情報の更新を行なう。これにより、学習者からのアドバイス要求に対応する。

このような処理を行なう事により、学生ワークステーションにおけるアドバイス機能の充実とマシン負担の軽減を計っている(図6.1.2)。

7. おわりに

演習支援システムをネットワーク環境上に実現する事により、膨大な情報を必要とするアドバイス機能の実現と各

学生ワークステーションでの負担の軽減を実現した。

本システムでは、各学生ワークステーションに、全情報の一部が配布され、これによりローカルでのアドバイス生成が可能であり、さらに必要に応じて演習サーバへアドバイス要求を出す事により必要なアドバイスを得る事を可能としている。

また、各学生ワークステーションで管理される情報は、学習者の進捗、理解度等を基にサーバのアドバイスマネージャとの協調処理により、常に必要なだけの情報となるよう管理されている。これにより、高度アドバイス生成機能実現に伴う各学生ワークステーションの負担の軽減を実現している。

本稿では、特にコースアウトラインモードにおける対話時に出されるアドバイスの生成について述べた。

【参考文献】

- [1]Kearsley, G.:Training for Tomorrow, Addison-Wesley,1985
- [2]尾関他、情報処理ハンドブック、オーム社、1989
- [3]福島、伊奥田、浮貝：ネットワーク型演習支援システムの構築(1)、情報処理学会第41回全国大会、1990
- [4]中村、平島、上原、豊田：ITSのための説明機能に関する検討、電子情報通信学会論文誌D-II Vol.J73-D-IIV No.3、pp.399-407、1990
- [5]福島、浮貝、菅原：分散処理環境上の演習支援システムの一構成、情報処理学会第45回マルチメディア通信と分散処理研究会(45-19)、1990
- [6]Manabu Fukushima, Masahiro Ukigai, Kenji Sugawara, Yoshiro Miida, Juro Oizumi:A Distributed Support System for Programming Laboratories,5th International Joint Workshop on Computer Communication,1990
- [7]坂元他:CAIハンドブック、フジテクノシステム、1989

