

ネットワーク型演習支援システムにおける 教材の利用について

福島 学* 浮貝 雅裕* 木下哲男** 菅原 研次*

*千葉工業大学 **沖電気工業(株)総合システム研究所

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。これに対応して集団教育を効果的に行うための一手法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。実用的な教育支援システムとして、集団教育における多様な学生の学習意欲を維持し、学習効果を向上させるためには、学習者の支援のみでなく教員に対する支援も必要となる。このような教育支援システムを実現するためには、高度な支援機能が要求され、さらに大量の教材や学習履歴データなどを扱わなければならない。したがってスタンドアロン型のパーソナルコンピュータでは機能的に実現が困難であり、コンピュータネットワークを利用した分散協調処理型の教育支援システムの開発が必要となる。

千葉工業大学情報工学科の情報系科目の演習を支援することを目的として現在設計、開発が進められているネットワーク型演習支援システムNEBLASS (NETwork-Based LAboratory Support System) はイーサネットワークで接続されたUNIXワークステーション上に実現される。NEBLASSにおいて演習授業支援、教材作成支援、教材管理などの機能を実現するための各機能モジュールは、ワークステーション上に分散的に配置される。本論文ではNEBLASSにおける学習教材の利用について具体的に述べる。

Usage of Teaching Materials in The Network-based Laboratory Support System

Manabu Fukushima* Masahiro Ukigai* Tetsuo Kinoshita** Kenji Sugawara*

*Chiba Institute of Technology **OKI Electric Industry Co., Ltd.

*2-17-1, Tsudanuma, Narashino-shi, Chiba, 275 JAPAN

**11-22 Shibaura 4-Chome, Minato-ku, Tokyo, 108 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. To develop a practical computer assisted learning system, many kinds of supporting functions are required for effective learning. Therefore, it is necessary to develop such a large scale software in a distributed processing environment.

NEBLASS (NETwork-Based LAboratory Support System) has designed to support programming laboratories for undergraduates at Department of Computer Science, Chiba Institute of Technology. The designed NEBLASS is supported to be implemented in the distributed processing environment using UNIX workstations which are interconnected by ether network. Under the NEBLASS framework, CAI functions can be distributed as cooperative software modules in the UNIX workstations. Support functions for running laboratory management, teaching materials development and teaching materials management, learning assist are implemented in workstations, distributedly.

This paper discusses the usage of teaching materials which is managed by Laboratory Server workstations.

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。これに対応して集団教育を効果的に行うための一手法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。以下本文では教育支援システムとは演習授業などを含めた集団教育を支援するシステムを指すものとする。効果的な教育支援システムを実現するためには、教材表示機能、コースナビゲーション機能、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豊富な支援機能が必要となる。

これらの支援機能を実現するための機能モジュールは、学習教材、学習履歴、成績データ、アドバイス知識などを利用しながら集団における多様な学生の学習意欲を維持し、学習効果を向上させるための、学習支援だけでなく教員の演習授業の運営および教材作成をも支援する。このような大規模なデータを扱い、高度な演習授業支援機能を実装するための計算機システムとして、スタンドアロン型のパーソナルコンピュータでは記憶容量や処理能力の面で不十分である。したがってコンピュータネットワークを利用した統合型教育支援システムを実現するための方法論を確立する必要がある。

このようなシステムを実現するためには、分散処理環境での実現を前提として

- (a)学習教材の情報構造の形式化
- (b)学習教材の利用モデルの定式化

を行う必要がある。このモデルに基づいて開発された学習教材や教育支援システムに必要な機能モジュールをネットワーク環境に分散配置して、これらの協調処理により統合型支援システムが実現可能となる。

千葉工業大学情報工学科の情報系科目の演習を支援することを目的として現在設計、開発が進められているネットワーク型演習支援システムNEBLASS (NEtwork-Based LAboratory Support System)では、演習授業支援、教材作成支援、教材管理を提供する機能モジュールがイーサネットで接続されたUNIXワークステーション上に分散的に配置され、演習におけるさまざまなタスクに対応して協調的にそれらのタスクを実現する[1][2]。現在までに分散処理環境に対応した教材の情報構造

の形式化が行われており[3]、本論文ではこの教材の利用について述べる。

2. NEBLASSの概要

ネットワーク型演習支援システムNEBLASSは図1に示すような学生用ワークステーション、教員用ワークステーション、演習支援用ワークステーションから構成されるLANで稼働するソフトウェアシステムである。

演習支援用ワークステーション

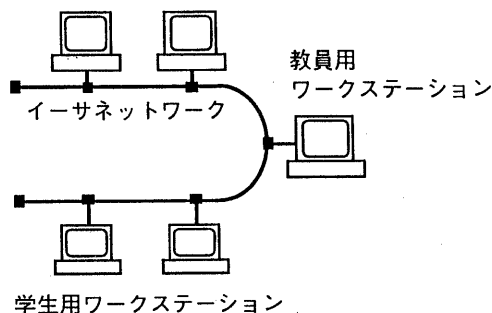


図1 NEBLASSのハードウェア構成

NEBLASSは次の演習支援機能を分散処理環境内に実現する事を目的として設計されている。

- (1)学習者主導型演習
- (2)学習者コースナビゲーション
- (3)教材管理
- (4)教材作成支援
- (5)学習履歴/成績管理
- (6)質疑応答/アドバイス

現在(1)(2)(3)の機能がプロトタイプシステムで実現されており、他の機能についても設計が進められている[4]。

NEBLASSは複数の演習授業科目を管理し、各科目に対して上記の機能を実現するため、管理しなければならない学習教材をはじめとする演習支援のためのデータは膨大なものになる。また上記(1)から(6)の機能を実現するためのソフトウェアシステムは大規模になる。この大規模なソフトウェアとデータを分散処理環境にバランス良く分散させるためにNEBLASSでは図2のようなソフトウェア構成をとる[1]。上記の各機能を実現するために学生支援モジュール、教員支援モジュール、演習支援/管理モジュールに含まれる機能要素モジュールが互いに協調しながら学生や教員を支援していく。機能要素モジュールは定められた

プロトコルに基づいて通信支援モジュール[2]を利用して協調処理を行う。

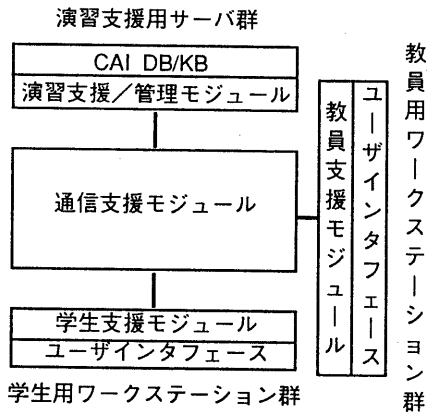


図2 NEBLASSのソフトウェア構成

3. NEBLASSにおける教材の情報構造

NEBLASSでは、コースウェアを識別するためのCID、コースアウトライン集合COおよび単元教材集合UMから構成される教材をコースウェアと定義する。コースアウトライン集合COは、コースアウトラインcoの集合であり、coには、どのように学習を進めて行くかがグラフモデルで記述されている。コースアウトラインcoは、コースアウトラインを識別するためのCOID、コースノードCN、コースノード間のリレーションCR、そのコースアウトラインがどのレベルの学習者を想定して作られたかを表すLによって表現される。

単元教材集合UMは、単元教材umの集合であり、単元教材を識別するためのumid、原理説明及び例題(Document)、練習問題(Exercise)、理解度を確認するためのテスト(Test)および分散処理環境に対応するための仮想インデックス(V_Index)によって表現される。単元教材は構造化されており、その構成要素である用語、文章、図表、問題間に対して教育上必要なリンクを定義する事ができる。したがって教員は、教材作成時に任意の教材データにインデックスと呼ばれる参照点を設定し、リンクを定義する事になる。

単元教材内のあるインデックスから他の単元教材のインデックスやコースノードにリンクが張られている場合、NEBLASSが分散環境で設計されているため、リンク先のインデックスが同一ワークステーション内に存在しない場合がある。

このため、単元教材には仮想インデックス(V_Index)が提供されている。仮想インデックスが参照された場合、リンク先が同一ワークステーション内にあるかどうかを調べ、無ければ通信支援モジュールを介して教材サーバから対象データを取り込む。これを実現するための手続きおよびデータが仮想インデックスに記述されている。特に、学生用ワークステーションとして使用しているマシンの制約から、利用不可能なタイプのメディアを教材として利用したい場合、対象がサーバで利用可能であれば、この仮想インデックスを用いる事によりサーバの機能を用いて教材として利用が可能となる。これは、現在使用しているマシンでは、動画画像および音が使用できないという制約があるが、教育的視点から教材として使用したいという要求に対応するためのものである。[3]

```

CW = <CID,CO,UM>
CO = { co | co = <COID, CN, CR, L> }
CN = { cn | cn = <cnid,attribute,unit,s_number,
title,abstract,test> }
CR = { cr | cr = <snode,dnode,label>, snode,
dnode ∈ CN }
UM = { um | um = <umid,Document, Exercise,
Test, V_Index> }
link = <s_ind, d_ind, label>
Document = <{COL},D_Link>
COL = { col | col = <colID, C_Attribute,
PARAGRAPH,FIGURE,PROGRAM> }
Exercise = <ExercisID, EXE>
EXE = { exe | exe = <num,problem,attribute,
procedure,hint,solution,E_LINK> }
Test = <TestID, TES, Interval>
TES = { test | test = <num, problem,attribute,
procedure,solution,score,T_LINK> }
CR = { cr | cr = <snode,dnode,label>, snode,
dnode ∈ CN }
label = {part-of, includes, enter_to, exit_to,next,
previous, grade_up,grade_down,
same_grade, basic, application}
  
```

4. 教材の利用モデル

4. 1. ステージの概念

教材の利用は、その目的によって、1) コースウェア選択および利用者特定、2) コースアウトライン参照および学習ノード選択[1]、3) 単元教材による学習[5]、の3つに分類する事ができる。本稿ではそれらをステージとして捉え、それぞれ1) 科目選択ステージ、2) コースアウトライン

ステージ、3) 単元学習ステージと呼ぶ事にする。

科目選択ステージでは、複数のコースウェアの中から学習するコースウェアを選択する事を目的としている。コースウェアの選択と学習者ID入力による学習者の特定により、システムは、学習者の学習履歴に基づき、必要なコースウェアの設定を行う。

コースアウトラインステージでは、設定されたコースウェアのうち、学習するコースノードを選択する事を目的としている。このため、コースアウトラインステージでは、設定されたコースウェアのうち、コースアウトラインに関する情報の参照が中心となる。

単元学習ステージでは、選択したコースノードに対応して作成された単元教材のより学習する事を目的としている。

4. 2. ステージ間推移

利用者は、これらのステージ間を推移する事により、学習を進めていく。ステージ間の推移は、科目選択ステージとコースアウトラインステージ、コースアウトラインステージと単元学習ステージ間の推移が定義されており、それぞれのステージの目的が達成された段階でステージ間の推移を行う。したがって、科目選択ステージでは、このステージの目的であるコースウェアの選択が行われた段階で、コースアウトラインステージへと推移し、さらにコースアウトラインステージにおいて、コースノードが選択された段階で、単元学習ステージへと推移していく。また、単元学習ステージにおいて、所定の学習を行うという目的が達成された段階で、単元学習ステージからコースアウトラインステージへと推移し、所定のコースノードの学習が終了した段階で、科目選択ステージへと推移する。特に、システムの起動時には、科目選択ステージにいるものとし、また演習の終了は科目選択ステージにおいて行われるものとする(図3)。

ステージ間の推移は、基本的に学習者主導型であり、学習者の意思で自由に推移するものとする。しかし、ステージ間の推移を行う場合には、演習を行う上で制約が必要な場合がある。この制約は以下の通りである。

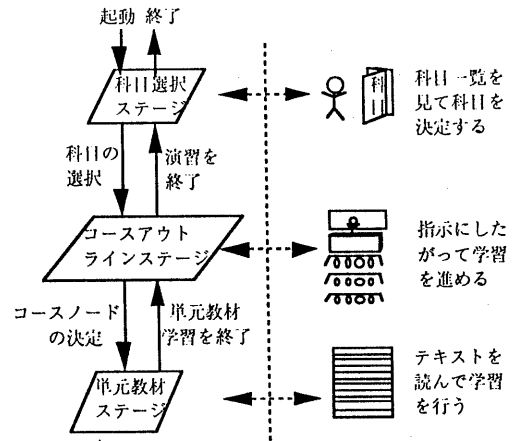


図3 ステージとステージ間推移

- a) 科目選択ステージからのコースアウトラインステージへの推移の場合
 - ・コースウェアが選択されている事
 - ・利用者が特定されている事
- b) コースアウトラインステージから科目選択ステージへの推移の場合
 - ・所定のコースノードを終了している事
 - ・連絡事項がある場合、これを確認している事
- c) コースアウトラインステージから単元学習ステージへの推移の場合
 - ・コースノードが選択されている事
 - ・選択したコースノードを学習可能である事
- d) 単元学習ステージからコースアウトラインステージへの推移の場合
 - ・所定の学習を終了している事

単元教材の学習を終了したかは、テストを行う事により判定し、教員により演習時に指定された所定のコースノードを終了したかは、学習履歴と、演習授業の運用上の制約として設定されたコースノードにより判定し、選択したコースノードの情報を参照して学習可能であるかの判定は、学習履歴とそのノード学習に必要な条件を参照して判定する。テストは、単元教材を作成する際に作成され、ノード学習に必要な条件は、コースアウトラインを作成する際に設定される。また演習時間に係わる制約としては、授業時間制や演習室利用などの関係から演習終了時間を設定することが可能である。これらの制約は、変更可能であり、演習授業を運用する担当教員に任される。

学習者がどのように学習を進めていくかの制約は、主にコースアウトラインとして記述される。

5. ステージ間推移の際の制約

5. 1. 科目選択とコースアウトラインステージ間推移の制約

科目選択ステージとコースアウトラインステージ間の推移の制約は、科目選択ステージからコースアウトラインステージへの推移と、コースアウトラインステージから科目選択ステージへの推移の2通りがある。前者は、選択したコースウェアを学習可能かどうかを判断するためのものである。このため、カリキュラムとしての制約として、履修年度および選択された科目を学習するために履修が必要となる条件が制約として記述される。これにより、履修不可能な科目を選択してしまうことを回避する事が可能である。これは、特に複数コースウェアがある場合には、必要な制約である。

後者は、演習授業として、所定の学習を終了しているかを判断するためのものである。このため、終了しなければならないコースノードおよび課題を制約として設定する。これは、集団教育として開講される演習授業において、所定の学習を行わせるために必要な制約である。

前者に対応して、システムは、科目選択可能か否かの判定を行い、可能であれば、科目に対応したコースアウトラインを設定し、コースアウトラインステージへの推移を許可する。この際に課題等の伝達事項を学習者に対して提示する。後者に対応して、システムは、終了条件の確認を行い、条件を満たしていれば、学習記録を演習サーバに登録し、伝達事項があれば掲示し、科目選択ステージへの推移を許可する。

5. 2. コースアウトラインステージと単元学習ステージ間推移の制約

コースアウトラインステージと単元学習ステージ間の制約には、コースアウトラインステージから単元学習ステージへの推移と、単元学習ステージからコースアウトラインステージへの推移の2種類がある。前者は、選択したコースノードを学習可能かどうかを判断するものである。基本的に学習者がどのように学習を進めていくかはコースアウトラインのノード間の移動によって表現される。これ以外に演習時間制および、演習室利用上

の制約から、時間の設定が必要となる。NEBLASSはこの設定により、教員の演習授業の宇尿支援を行う。ただし、これは、担当教員に延長等の操作が任されている。後者は、コースアウトラインで選択したコースノードにおいて所定の単元教材の学習を終了したかを判断するためのものであり、このための制約として単元教材終了時にテストを実施する。これにより、単元学習ステージからコースアウトラインステージへの推移が可能であるかを判断する。ここで行われるテストの答案およびテスト結果は、学習履歴として記録される。テストの終了には、GiveUp、Return、OKがある。GiveUpはテストに対する回答を作成しきれない事を意味し、システムに対して基礎学習等のコースナビゲーションを必要とするという意思表示である。Returnは、推移を断念し、再度同じ単元教材を用いて学習を行いたいという意思表示である。OKは、答案作成終了を意味する。したがって、Returnを選択した場合以外、単元学習ステージからコースアウトラインステージへの推移が許可される。推移の際、伝達事項等がある場合、それを表示する。

前者に対応して、システムは、その単元学習が可能か否かの判定を行い、可能であれば、選択されたコースノードに対応した単元教材を設定し、単元学習ステージへの推移を許可する。後者に対応して、システムは、テスト問題および答案用紙を表示し、答案の作成を行わせる。テストの終了としてGiveUpまたはOKが選択された場合、コースアウトラインステージへの推移が許可される。

6. NEBLASSの学習用ユーザインタフェース

演習支援システムでの教材の利用は、科目選択ステージ、コースアウトラインステージ、単元学習ステージの3つのステージ、ステージ間の推移および推移の際の制約によって表現される。科目選択ステージでは、科目を選択/参照するための科目ブラウザを提供する(図4)。このステージで選択可能な操作として、学生ID入力、概要説明、学習履歴要求、科目決定、終了が提供されており、これらのオペレーションは図4に示すように、ボタンとして画面上に提供されている。

コースアウトラインステージでは、コースアウトラインを見て、コースノードを決定するための

コースアウトラインブラウザが提供されており、操作として、コースノードの概要説明、学習履歴要求、学習要求、アドバイス要求、基礎要求、応用要求、レベル変更要求、位置表示要求、終了が提供されており、これらのオペレーションは図5に示すように、ボタンとして画面上に提供されている。

単元学習ステージでは、学習を行うために必要な操作として、説明要求、例題要求、練習問題要求、応用要求、基礎要求、学習履歴要求、学習進捗表示要求、終了が提供されており、図6のように、ボタンとして画面上に提供されている。

学習進捗表示要求は、コースアウトライン上、現在自分がどこを学習しているか、またあといくつノードの学習を終了すれば所定の学習を終了できるかの表示を要求するものであり、全体における位置付けを確認するための操作である。

7. おわりに

情報系科目の演習授業の支援を目的としたネットワーク型演習支援システムNEBLASSにおける教材の利用について述べた。NEBLASSで使用する教材は、それぞれの目的に対応した3つのステージによって利用される。学習者は、各ステージに設定された制約条件を満足する事により学習を進めていく。一般に演習授業に必要な教材は膨大な情報を包含し、さらに複数の科目で演習用計算機システムを利用する場合、十分な支援機構がない限り円滑な授業の運営が困難である。

現在NEBLASSにおける演習授業の支援に必要な機能要素の設計を進めており、学生ワークステーションのユーザインタフェースなどの一部の機能モジュールの実装、評価を進めている。NEBLASSはイーサネットワークで接続された45台のUNIXワークステーション上の、Smalltalk-80を利用し、プロトタイプシステムとして開発を行っている。

[参考文献]

- [1]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
- [2]M.Fukushima, M.Ukigai, K.Sugawara, Y.Miida, J.Oizumi: A Distributed Support System for Programming Laboratories, 5th Joint Workshop on Computer Communications, 1990
- [3]福島、浮貝、菅原:ネットワーク型演習支援システムにおける教材サーバについて、情報処理学会第49回マルチメディア通信と分散処理研究会(49-8)、1991
- [4]福島、浮貝、菅原: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、情報処理学会第47回マルチメディア通信と分散処理研究会(47-4)、1990
- [5]鈴木、福島、菅原:ネットワーク型演習支援システムの構築(4)、情報処理学会第42回全国大会、9月、1990

| 科目ブラウザ | | | | |
|------------------------------------------------------------------------------------------------------------------------|------|------|------|----|
| 学生ID入力 | 概要説明 | 習履歴要 | 科目決定 | 終了 |
| <p>UNIX演習</p> <p>Smalltalk演習 C言語演習 ソフトウェア演習 Fortran演習</p> | | | | |
| <p>UNIX演習では..... マシンを利用して、実際にUNIXを使う事を目的としており、最終的には、ProcessおよびFileSystemについて学ぶ事を目的としている。 [流れ] マシンの起動操作</p> | | | | |

図4 科目ブラウザ

| コースアウトラインブラウザ | | | | | | | |
|---------------|--------|---|---|----|---|---|------------|
| 要説 | 履歴 | ト | 道 | ダイ | 確 | 要 | 用 |
| 要 | 要 | 要 | 要 | 要 | 要 | 要 | 要 |
| の歴史 | 基礎演習 | | | | | | 応用 |
| Shellとは | 利用者を見る | | | | | | Processを見る |

必要なコースノードの学習が終了しているので、選択したコースノードを学習することが可能です。
今回の課題は、「UNIXの基礎演習」を終了する事です。

図5 コースアウトラインブラウザ

| 説明要求 | 例題要求 | 習問題要 | 応用要求 | 基礎要求 | 習履歴要 | 置表示要 | 終了 |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|------|------|------|------|------|----|
| <p>ログインしている人の確認</p> <p>現在そのマシンにログインしている利用者を確認するためのコマンドとして、「who」が提供されています。</p> <p>このコマンドを実行する事により、現在利用しているマシンに誰が入っているか/利用しているかを確認する事ができます。</p> <p>(使用例) who</p> <p>(結果の説明) 一行に一人分が表示されており、</p> | | | | | | | |
| <p>実行/確認用Window</p> <pre>\$who c056 console Apr 20 00:09 c056 tty0 Apr 20 00:08 \$</pre> | | | | | | | |

図6 単元学習ステージ