

## 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

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

CAIシステム機能の高度化に伴うソフトウェアシステムの大規模化に対応するためには、分散環境上にCAI機能を効果的に分散して構築する、ネットワーク型CAIシステムの構成法を確立する必要がある。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)は学生用、教員用及び演習サーバの3種類のワークステーションから構成されており、演習授業支援機能、教材作成支援機能、教材管理機能、成績管理機能などのさまざまな機能が各ワークステーション上に分散配置される。本稿ではNEBLASSの教材サーバで管理される、演習授業に用いる教材の情報構造について述べる。

## A Material Server in Network-based Laboratory Support System

Manabu Fukushima\* Masahiro Ukigai\*\* Kenji Sugawara\*\*  
\*Fuji Xerox \*\*Chiba Institute of Technology

To develop a high-level CAI system, the software should be large and complicated. However, due to the large software size, stand-alone personal computers can hardly implement it. Therefore, it is very important to devise methodology for designing a network-based CAI system, which is to be realized within a distributed processing environment. NEBLASS(NETwork-Based LABORatory Support System) has designed to support programming education for undergraduates at Department of Computer Science, Chiba Institute of Technology. The designed NEBLASS consists of three kinds of workstations: they are named as Student Workstation, Authoring Workstation and Laboratory Server Workstation, respectively. Under the NEBLASS framework, CAI functions can be distributed as cooperative software modules in these workstations. Support tasks for programming education are realized by cooperative work of these modules.

This paper discusses the data structure of teaching materials which is managed by Server.

## 1.はじめに

CAIシステムの機能が高度になるに伴い、そのためのソフトウェアシステムは大規模化してくる。これに対応して分散環境上にCAI機能を効果的に分散して構築するネットワーク型のCAIシステムの構成法を確立する必要がある[1][2]。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 (N**e**twork-B**a**sed L**a**boratory S**u**pport S**y**stem)は学生用、教員用及び演習サーバの3種類のワークステーションから構成されており、演習授業支援機能、教材作成支援機能、教材管理機能、成績管理機能などのさまざまな機能が各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これらの機能モジュールが協調的にそれらのタスクを実現する[3][4]。

NEBLASSにおいては、演習教材は演習サーバワークステーションにおける教材サーバで一括して管理され、演習授業の中で必要に応じて学生用ワークステーションから利用される。教材は教員用ワークステーションのオーサリング支援環境において、教材部品を作成し、それらの教材部品を利用して一演習科目のコースウェアを編集する形式で作成されていく。コースウェアは単元教材という教材構成の基本単位と、授業の流れや階層構造を表現するコースアウトラインから構成される。単元教材はドキュメント、練習問題、テストの3種類の要素から構成され各要素はテキスト、図式や画像、プログラムなどのマルチメディアにより記述され、それらが学習のガイド方法に基づいて複雑な関係が定義されているハイパーメディアとして構成されている。

本稿においては、NEBLASSにおける教材の情報構造と、ネットワーク内の教材の管理について述べる。

## 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LANで稼働するソフトウェアシステムである。NEBLASSは図2に示す4種類のソフトウェアモジュールの集合から構成される。

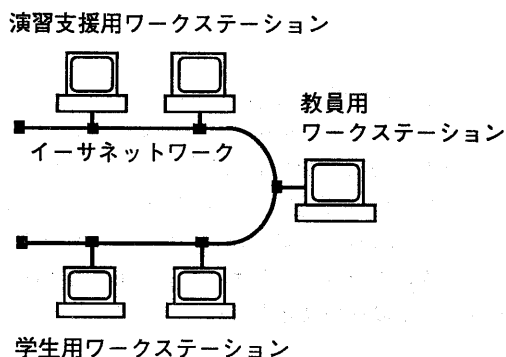


図1 演習授業のためのLAN

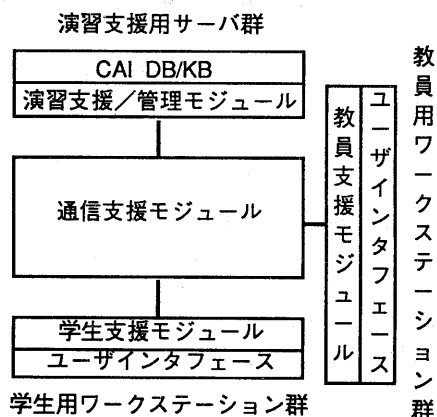


図2 NEBLASSのソフトウェア構成

学生用ワークステーションには学生が学習を行うための演習支援ソフトウェア、教員用ワークステーションには教材の作成や成績の分析を行うための演習管理ソフトウェア、演習サーバワークステーションには教材を管理するための教材サーバ、成績管理サーバ、学生モデルサーバ、学習履歴サーバなど演習を行うために学生や教員を支援するためのさまざまなソフトウェアシステムが必要である。これらのネットワーク上に分散されているソフ

トウェアモジュールの間の通信を効率的に支援するために通信支援モジュールが用意されている。

NEBLASSでは演習実行、コースナビゲーション、成績処理などの演習の実施に必要な機能を、分散したソフトウェアモジュールが通信支援機構を利用して協調的に実現して行く。これらの分散モジュールは演習のための教材モデル、学生モデル、アドバイスマodelなどの知識を共有し、これを基に協調的に学生及び教員の演習行為を支援していく。

### 3.NEBLASSにおける教材の情報構造

#### 3.1 コースウェアとコースアウトライン

一科目の演習授業に用いる教材の全体をコースウェアという。コースウェアCWを次の三組で記述する。

$$CW = \langle CID, CO, UM \rangle$$

CIDはコースウェアに関するカリキュラム上の情報の記述要素で、科目名、開講学年、単位数、担当教員名などの記述を含む。UMは単元教材umの集合でumはモジュール化された教材の最小構成要素であり、次の3.2節で詳述する。

COはコースアウトラインcoの集合である。coは単元教材の間の関係を記述したグラフモデルであり、単元教材の学習順序などに使われる情報である。

$$CO = \{ co \mid co = \langle COID, CN, CR, L \rangle \}$$

COIDはコースアウトライン識別子である。CNは次に示すコースノードcnの集合である。

$$CN = \{ cn \mid cn = \langle cnid, attribute, unit, Section, Title, Abstract, Test \rangle \}$$

cnidはコースノード識別子で、attributeはコースノード属性である。コースノード属性attributeは内部ノードINとリーフノードLNのうちのいずれかの値を持つ。attributeの値がLNの場合はそのコースノードに学生が学習を行う単元教材が割り当てられていることを意味し、unitはその単元教材へのポインタである。

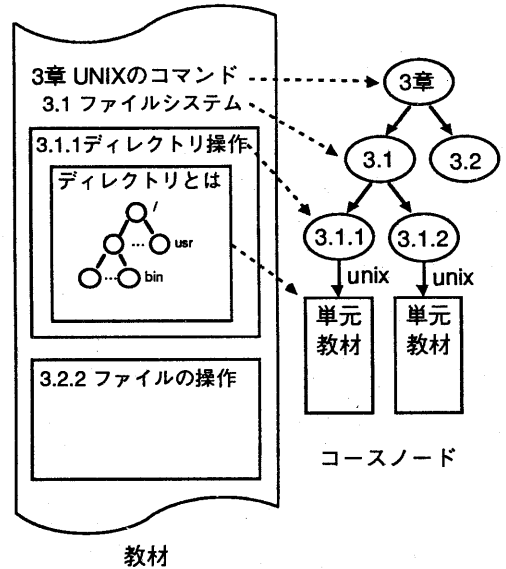


図3 コースノード属性

attributeの値がINの場合はそのコースノードは教材の構成概念を表すためのノードであることを意味する。たとえば図3に示すような章や節など内部にさらにいくつかの教材要素を含む概念に対応する。またattributeがINTestであるときは、そのコースノードには教材は割り当てられていないが、そのコースノードを通過するときはテストを受けなければならない事を意味する。そのテストの情報はTestに記述されているがその内容は次の節で述べる。Titleはそのコースノードに割り当てられた題目、Sectionはその章あるいは節の番号、Abstractはそのコースノードで学習する目的や概要を記述した文書である。

CRはコースノードリレーションと呼ばれるラベル付きの弧の集合である。

$$CR = \{ cr \mid cr = \langle snode, dnode, label \rangle, snode, dnode \in CN \}$$

snodeは弧crのソースノードでdnodeはcrのデスティネーションノードである。labelはcrのラベルである。ラベルは教材の構造や学習のナビゲーションなどの学習制御に用いるものなどさまざまにリレーションが定義される。この関係については3.3節で詳しく述べる。

コースアウトライン集合COでは学習者の理解度に対応して図4に示すようないくつかの学習レベルのコースアウトラインが用意されている。Lはコースアウトラインレベルを表す値である。レベルが異なるコースアウトライン間の遷移は特定のコースノードの間に張られたGradeUp,GradeDownのリレーションをもちいて行われる。

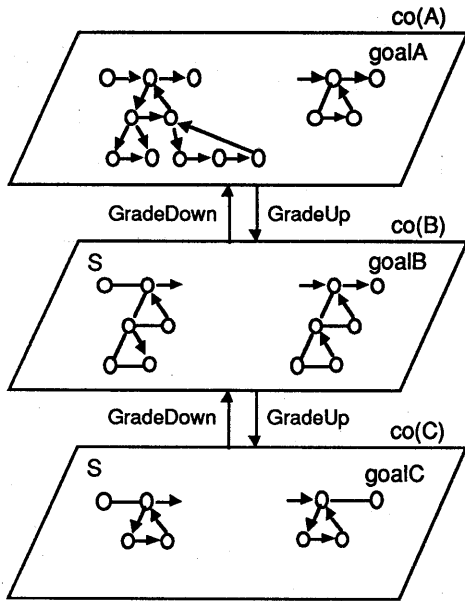


図4 コースアウトラインレベル

### 3.2 単元教材

単元教材  $u_m$  は学習対象の実体を表す情報であり、次の単元教材要素

Document, Exercise, Test から構成される。

$$UM = \{um \mid um = \langle umid, Document, Exercise, Test, Index \rangle\}$$

umid は単元教材識別子で、Document は単元の説明文、図表、例題などからなる教材データである。単元教材はハイパーメディアであり、教材作成時に任意の教材データにアンカーポイント(ap)[5]を設定し、その間にリンクを定義する事ができる。リンクは図5に示すように単元教材要素内リンク、単元教材要素間リンク、単元教材間リンクに類別される。これらを一括して教材間リンクと呼び、コースノード間リレーションと区別する。単元教材の

あるアンカーポイントが他の単元教材のアンカーポイントやコースノードにリンクが張られている場合はNEBLASSは分散環境で動作する事を考慮にいと、相手のアンカーポイントが同一ワークステーション内に存在しない事がある。従って単元教材間リンクは他の種類のリンクと区別して扱う。すなわちIndexという仮想アンカーポイントを単元教材内に用意しておき、仮想アンカーポイントが参照されたとき、それが同一ワークステーション内にあるかを調べ、無ければ通信支援モジュールを利用して教材サーバから対象データを取り込む。Indexにはそのためのデータ及び手続きが記述されている。

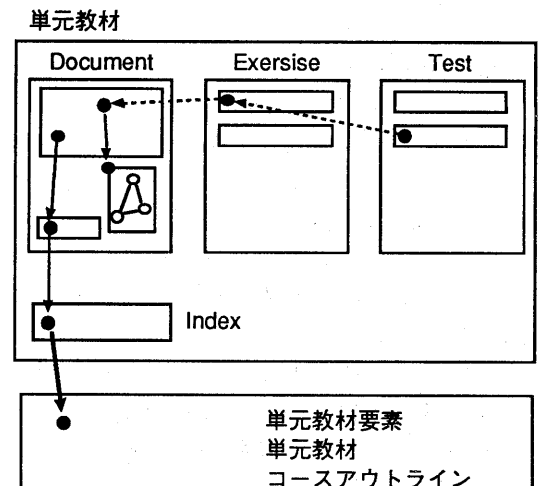


図5 単元教材のリンク

Documentは演習授業に関する概念、原理、用語、方法などの説明文書であり、次の構造を持つ。

$$\begin{aligned} \text{Document} &= \langle \text{MainCol}, \text{COL} \rangle \\ \text{COL} &= \{ \text{col} \mid \text{col} = \langle \text{colID}, \text{PARAGRAPH}, \\ &\quad \text{FIGURE}, \text{PROGRAM}, \text{LINK} \rangle \} \\ \text{LINK} &= \{ \text{link} \mid \text{link} = \langle \text{ap}, \text{dest}, \text{label} \rangle \text{ dest} \in \\ &\quad \text{COL} \cup \text{EXE} \cup \text{TES} \cup \{ \text{ap} \} \} \end{aligned}$$

Documentはコラムcolの集合からなる。コラムは説明文書の単位で、文書、図面、プログラムから構成され、コラム内、コラム間にリンクが定義されている。このリンクを利用して、学生の学習状況に応じて説明のしかたが違うコラムや補足説明のための文章、図面、例題が書かれているコラムを参照できる。

colIDはcolの識別子であり、この他、文字列からなるPARAGRAPH、図面を表すFIGURE、演習対象の言語により記述されたPROGRAMである。ただし中心となる教材文書はMainColに書かれており、演習時のDocumentの初期表示画面はこれが表示される。教材間リンクはLINKで記述する。colの内容は表示オブジェクトとして管理されているが、本稿ではsmalltalk-80の表示オブジェクトをこれに当てている。

Exerciseは問題とその解答やヒントに関するデータである。

Exercise = <ExerciseID, EXE>

EXE = {exe | exe = <num,problem,attribute,procedure, hint,solution,LINK>}

numは演習問題識別子で、problemは文、図表を含む問題文書で、学生は演習インタフェース[6]で与えられる演習解答ウィンドウに回答を書き込む。このとき演習問題の種別によりいくつかの回答確認手続きprocedureが定義される。この手続きを起動する方法（デモンタイプ、サーバントタイプ）がattributeに記述されている。hint,solutionは必要なときに学生が参照する事ができる。また前に定義した教材間リンクはLINKで記述する。

Testは教材単元の終了時に理解度を確認するための試験データである。

Test = <TestID, TES>

TES = {test | test = <num, problem,attribute,procedure,solution,LINK>}

TestIDは識別子で、TESはテストの問題testの集合である。num,problem,solutionはEXEと同じであるが学生はsolutionを参照する事は許されない。attributeは回答方法種類の記述であり、選択式と記述式の両方がある。

procedureは回答するための手段を提供するた

めの手続きである。

### 3.3 コースノード間関係

コースアウトラインでは、演習を進めるためのコースに関する記述がされる。この記述は、コースノードリレーションによって表現される。このリレーションには、部分関係を表わすためのpart-of、includes、内部ノード(IN)とリーフノード(LN)の関係を記述するenter\_to、exit\_to、コースとしての前後を現わすnext、previous、コースアウトライン間のレベル関係を表わすgrade\_up、grade\_down、さらに同一レベルの関係を表わすsame\_gradeを定義した。

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}

upper\_gradeおよびlower\_gradeのリレーションをもつコースノードは、そのコースノードにおいてレベルの変更が可能である事を意味する。特に下位レベルから上位レベルへ移動する場合、上位レベルにおける学習に必要であろうと思われる概説（サマリ）が提供されており、これにより下位レベルから上位レベルに移った場合のギャップを埋める。

テストの結果等により、現在のレベルでの学習が不適切であると判断される場合、まず同一レベルを表わすsame\_asのリレーションが張られているコースノードの学習を行い、それで理解できれば、そのままのレベルで次のコースノードへと学習を進めていくが、いくつかの同一レベルのコースノードを学習しても理解できない場合、previous関係を辿り、もっとも近いレベル変更可能なコースノードへ移動し、下位レベルに移る。そこでの学習結果によって、再度上位レベルに戻るか、または、そのレベルで学習を進めるかが決定される（図6）。

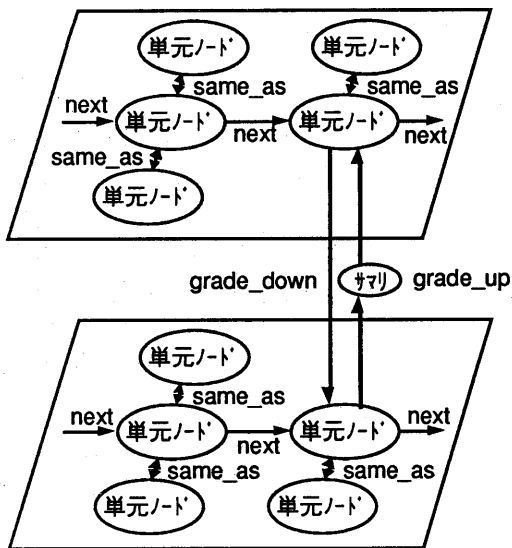
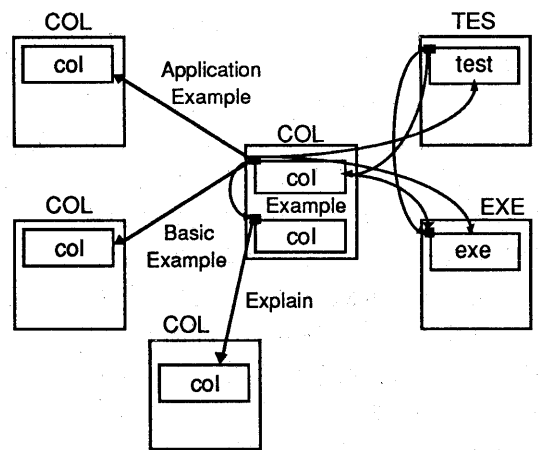


図6 コースアウトライン間のグレード



単元教材要素内リンク 単元教材要素間リンク

図7 単元要素間、単元教材要素内リンク

### 3.4 教材間リンク

教材間リンクとは、単元教材要素間リンク、単元教材要素内リンクおよび単元教材間リンクの総称である。

単元教材要素間リンクとは、Document、Exercise、Testの間のリンクを記述するものであり、アンカーポイントによって、説明に対応する練習問題およびテスト間のリンクが記述される。

単元教材要素内リンクは、単元教材を構成する要素に対して付加情報のためのリンクを記述するためのものであり、アンカーポイントによって記述される。この付加説明のための教材の間の関係はリンクとして表現される。このリンクとして、原理説明と例題間のリンクを記述するためのExample、補足説明のためのComment、原理説明とそれを用いた応用例のリンクを記述するためApplicationExample、基礎的な例とのリンクを記述するためのBasicExample、用語の説明または詳細な説明のためのExplainを提供する(図7)。

単元教材間リンクは単元教材要素内リンクのうち、他の単元教材内の要素に対してリンクが張られているものを表す。

単元教材要素内リンクのうち他の単元教材へのリンクは、index内の仮想ノードに対して単元教材要素内リンク、仮想ノードから目的の要素を含む単元教材のindex内の仮想ノード、相手の仮想ノードから目的の要素に対して単元教材要素内リンク、によって表現される。単元教材間リンクはこれのうち、仮想ノード間のリンクである。このため、indexには、複数の仮想ノードの定義が必要であり、このためのテーブルを持っている(図8)。

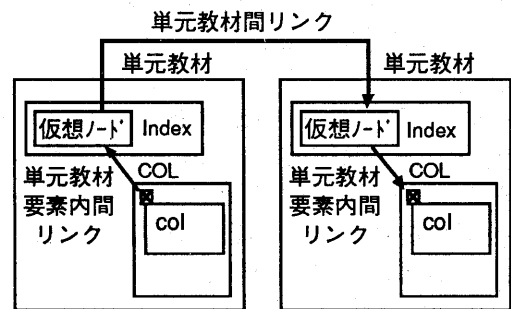


図8 単元教材間リンク

教員は、これらの関係を一つ一つ定義する必要はなく、教材作成支援環境において単元教材間のリンクを定義する事により、教材作成支援モジュールが必要なリンクへの分割および仮想ノードの設定を行う。

これらの教材間関係により、単元教材として必要なリンクの記述が行える。特に、indexにおいて仮想ノードを提供することにより、単元教材間でのリンク記述を行う事により、分散処理環境上での利用に対応する事が可能である。

#### 4.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の教材の情報構造について述べた。ネットワーク型演習支援システムNEBLASSにおける教材は単元教材と、授業の流れに基づいたそれらの間の関係であるコースアウトラインにより構造化されており、教材サーバで一括して管理される。一般に教育機関において半期あるいは通年の授業に必要な教材は膨大な情報を包含し、また複数の科目で演習用コンピュータシステムを利用する場合、強力な支援機構が無いかぎり、教材や授業結果の管理が困難であり、スタンドアロン型の演習システムでは円滑な演習授業の運営に問題を生じる。

NEBLASSでは教材及びコースアウトラインはハイパーメディアとして作成され、そのハイパーリレーションは分散処理環境上で定義されている。従って授業の進度に従って自由に必要な教材をアクセスできるため、半期のカリキュラムコースに必要な大量の演習教材を管理、実行することができる。

現在NEBLASSは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で稼働し、smalltalk-80を利用してソフトウェアのプロトタイプの開発を行っている。

#### [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、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
- [4]M.Fukushima, M.Ukigai, K.Sugawara, Y.Miida, J.Oizumi: A Distributed Support System for Programming Laboratories, 5th Joint Workshop on Computer Communications, 1990
- [5]Special Issue: "Hypertext", COMMUNICATIONS OF THE ACM, Vol.31, No.7, 1988
- [6]浮貝、福島、菅原、三井田: 情報工学科新入生に対する導入教育のための演習授業について、情報処理学会、コンピュータと教育(12-4)、1990