

知的マルチメディア CAI における Web ページ統合モデルの提案

山北 隆典[†] 富士 隆[†] 三枝 武男^{††}

知的マルチメディア CAI の教材部品であるハイパーフレームを管理するために Dexter モデルに基づいたハイパーメディア型データベースを開発してきた。しかし、ハイパーフレームは固有のデータモデルに基づいたオブジェクトであり、その開発コストは高い。教材開発のコスト削減には、WWW 上の既存の Web ページの利用が有効である。Web ページを CAI の教材部品として利用するためには、Web ページをデータベース上でハイパーフレームと分け隔てなく統合的に管理する必要がある。本論文では、統合管理の要件として、(1) 利用する Web ページの限定、(2) Web ページの属性管理、(3) Web ページ間のハイパーリンク管理、(4) Web ページの部品化情報の管理、(5) クライアント・プログラム情報の管理、を取り上げる。さらに、これらの要件の実現には Web ページを提示する際のコンテンツの動的な変更機能、すなわちビュー機能の実現が鍵となることを指摘し、Dexter モデルにおけるビュー機能を司る presentation specifications を機能拡張することで、学習という文脈で Web ページを利用するために必要な情報を管理するモデルとして、Web ページ統合モデルを提案する。最後に、その妥当性を試作システムの開発例によって示す。

An Integration Model of Web Pages for the Intelligent Multimedia CAI

TAKANORI YAMAKITA,[†] TAKASHI FUJI[†] and TAKEO SAEGUSA^{††}

In order to manage hyper-frames which are instructional units for the intelligent multimedia CAI system, we have developed a hypermedia database based on the Dexter model. However, developing hyper-frames costs much time and labor, because of its peculiar data model. Therefore, the use of existing Web pages on the WWW is an effective approach for reducing the cost. For utilizing Web pages as instructional units in the CAI system, it is necessary to handle them together with hyper-frames uniformly in the database. In this paper, we point out some requirements for integrating Web pages into the database. In order to satisfy them, we propose to extend the presentation specifications which manage a view function in the Dexter model. Moreover, we describe an object-oriented framework for the integration of Web pages into the database. Finally, we show that our integration model is valid by constructing the prototype system.

1. はじめに

企業や学校、家庭にも多様な形でコンピュータが普及するにつれ、あらゆる分野でマルチメディアを取り込んだ様々な形式の情報が生成され、利用されるようになった。さらに、インターネットの普及によって、こうした情報を共有し統合管理するためのデータベース技術がますます重要になっている²¹⁾。このような状況はコンピュータによる学習システムの研究においても例外ではない。

我々はこれまで個々の学習者の学習状況に適應したマルチメディア教材を動的に提供できる知的マルチメディア CAI (Computer-Assisted Instruction) のフレームワークの開発^{5),6)}やその統合開発環境の研究⁷⁾を続けてきた。このような CAI システムでは教材を部品化し適切に属性付けしたうえでデータベース上に管理しておくことで、学習者の状況に応じた動的な部品の組合せとして、適應的な教材の提供を実現できるとともに、教材部品の再利用による教材開発コストの削減も期待できる。

我々は文献⁵⁾で教材部品のモデルとしてハイパーフレームを提案している。ハイパーフレームとは学習を進めるうえでの基本単位であり、教えるべき内容をマルチメディア・データとして保持するとともに、その内

[†] 株式会社学習情報通信システム研究所
Software Research Laboratory

^{††} 北海道情報大学

Hokkaido Information University

容に固有な教材知識もカプセル化したオブジェクトである。その属性として、学習項目、内容の難易度、主要な提示メディアの型などを付与しており、教材部品として適切にモデル化されている。また、ハイパーフレーム間には参照関係も設定されており、学習者が主体的に関連をたどることもできる。そこで、ハイパーフレームを管理するための CAI 教材データベースの基本機能として、属性値を利用した集成的検索機能と教材部品間の関連を利用したナビゲーション（構造的検索）機能を位置付け、ハイパーテキスト¹⁷⁾の参照モデルである Dexter Hypertext Reference Model¹¹⁾（以降、Dexter モデルと略す）に基づいたハイパーメディア型データベースの研究・開発を行ってきた。

ところで、教材作成者は自分の教えたい内容そのものについては自らが積極的に教材部品を開発するが、その周辺分野や関連知識についてはできるだけ開発の労力を省きたいという要求を持っている。この要求に応えるアプローチは 2 通りに大別できる。1 つは、他の分野向けに開発されたハイパーフレームを利用しようという立場で、教材部品としての再利用性を活用するアプローチである。その実現に向けた取組みとして、インターネット上に複数の CAI 教材データベースが存在する環境において、それぞれのハイパーテキスト型教材を相互利用できるようにハイパーリンク・ビューという概念を導入した Dexter モデルのリンクメカニズムの拡張をすでに提案した²²⁾。

2 つ目のアプローチはハイパーフレームという枠にとらわれずに、参照可能なデジタル文書を教材部品として利用しようというアプローチである。ハイパーフレームは固有のデータモデルに基づいており、その開発コストが高いために既存の資産はまだ少ない。しかし、近年デジタル文書として一般的となっている WWW 上の大量の Web ページの中には、学習教材として有効な内容を持った文書も少なくない。ハイパーフレームに比べ教材部品としての機能（マルチメディアの提示機能や学習者との対話機能など）は劣るが、ハイパーフレームの補足的な利用には有効であり、教材作成者の要求に応える手段として期待できる。

本論文では上述した第 2 のアプローチの実現、すなわち既存の Web ページを知的マルチメディア CAI の教材部品として利用するために、CAI 教材データベース上に Web ページをハイパーフレームと分け隔てなく管理し、利用するための統合モデルを提案し、コンピュータによる学習システムの研究を通じて新世代データベースの取り組むべき課題の 1 つの方向を提起する。

2. Web ページ利用の現状

既存の Web ページを学習教材として積極的に利用した例として K-12（幼稚園から高校）を対象とした研究⁸⁾がある。このシステムでは教材作成者が既存の Web ページにパス（paths）^{*}を定義することで、カリキュラムに基づいた学習を支援している。Web ページをカリキュラムでの文脈で利用できるように、学習者に提示する際にコンテンツの一部を隠蔽したり、導入文を追加したりでき、部品としての利用を可能にしている点も特徴である。ただし、Web ページ以外の教材は扱っていない。

また、WWW のデリバリー機能を利用した CAI システムが多数提案されている。その多くは、WWW サーバのバックエンドに ICAI（Intelligent CAI）もしくは ITS（Intelligent Tutoring System）と呼ばれるシステムを位置付け、その出力結果を HTML（HyperText Markup Language）化したのち WWW ブラウザへ転送するといったアーキテクチャを採用している^{2),14)}。つまり、教材自体は独自の形式で表現・管理されている。

こうした WWW ベースの CAI システムに既存の Web ページを利用している例として、CALAT¹⁶⁾がある。このシステムでは、システムが管理している HTML ファイルによって学習の進行を WWW ブラウザ上でを行い、ICAI の提供するマルチメディア教材の提示は外部の専用ビューアで実現している。HTML ファイルから既存の Web ページへのリンクが設定できる。しかし、利用可能な Web ページとしては独立性の高い辞書的な内容のものを想定するにとどまっており、また、それらの Web ページを ICAI の中で HTML ファイルや独自のマルチメディア教材と統一的な枠組みで管理しているわけではない。

教室内 LAN 上の独自の学習環境の中で、専用に設計されたハイパーメディア教材に加えて、Web ページも教材として扱えるシステム¹⁾も提案されている。Web ページは URL（Uniform Resource Locator）で管理しており、Web ページが呼び出されると WWW ブラウザを起動し、URL で示された Web ページを提示する。

注意すべきことは、いずれのシステムにおいても Web ページ上にあらかじめ埋め込まれているリンク（以降、Web ページの作成者によって HTML で記述されたリンクを embedded links と呼ぶ）によるナビ

^{*} 複数のリンクを連結してあらかじめ順序付けしたナビゲーションの道筋。

ゲーションを学習者が自由に行える点である。

既存の Web ページと他のアプリケーション文書との統合の研究として、open hypermedia system の研究がある。DHM¹⁰⁾は Dexter モデルに基づき、異種メディアの文書を統合的に扱えるハイパーメディアシステムのオブジェクト指向フレームワークであり、Web ページもこのフレームワークの中で扱える。しかしながら、Web ページ上の既存の embedded links についてシステムは関与しない。

データベース研究の側からは、WWW とデータベースの連携が多数報告されている^{13),18)}。しかし、それらはもともとデータベース上で定形的に構造化され管理されていたデータを HTML 化して WWW ブラウザを通じて利用するタイプの範疇にあるが、我々のアプローチは既存の Web ページを利用するための情報をデータベースで管理するという点で異なっている。

3. Web ページ統合モデルの概要

3.1 CAI 教材データベースへの統合の要件

学習者の状況に適応した教材を動的に提供するために、ハイパーフレームには、その学習項目、内容の難易度、提示メディアの型などの属性値を付与して管理している。Web ページも知的マルチメディア CAI の教材部品として利用するためには同様に属性管理の対象とならなければならない。ハイパーリンクによるナビゲーション（構造的検索）だけではなく、属性値による集成的検索によっても利用できなければ、教材部品として統合的に管理されたとはいえない。同様に、Web ページ間のハイパーリンクをハイパーフレーム間の関連と同一の形式で管理する必要もある。ハイパーフレームか Web ページかによらず、教材部品としてその間に関連が設定でき、関連を利用した処理が分け隔てなく実現できることもハイパーメディア型データベースでの統合的な管理の必要条件といえる。

ところで、明確な学習目標を持った CAI のための教材データベースでは、そこで利用する教材部品は教材作成者により厳選され、すべて管理の対象としておく必要がある^{*}。しかし、Web ページ上の embedded links に従った芋づる式のナビゲーションによって利用できる Web ページの中には、教材作成者から見ると学習者にとって不必要な情報、あるいは学習に悪影響を与える恐れのある情報しか含まないページが少な

くない^{**}。すなわち、学習の過程で Web ページを提示するときにはハイパーリンクを自由に使わせるのではなく、そういった Web ページへのハイパーリンクは隠蔽し、学習で利用できる Web ページを限定することが大前提となる。

次に、ハイパーフレームと Web ページとの差異を考慮すると、前者が学習者の状況に応じた教材の動的な提供を可能にするために適切に部品化されているのに対し、後者はその作者の文脈に依存した記述がなされている。したがって、教材部品として利用するためにはもともとの文脈に依存した記述を隠蔽したり、部品として利用するための補足的な記述を追加したり、また、他の部品との粒度を合わせるための分割・合成も必要になる。これらの処理は、その Web ページを CAI の中で利用する際に動的に行う必要があり、もともとの文書そのものに手を加えるものであってはならない。なぜなら、Web ページのコピーを作り内容を書き換えて利用すると、もともとの Web ページの内容が更新されても、それが反映されなくなるため、教材作成者がコピーに手を加える必要が生じるからである。このような手間は極力省きたい。また、ハイパーフレームは複数のマルチメディアの組合せで内容を提示するとともに、内容固有の教材知識を用いて学習者と対話するために、特別なクライアント・ソフトウェアが必要である。他方、Web ページは通常の WWW ブラウザで利用できる。したがって、管理上はいつでも教材部品として分け隔てなく扱うけれども、学習者への提示の際に利用するクライアント・ソフトウェアは使い分ける必要がある。教材の部品化という主旨からは、こうした情報は教材部品の属性としてデータベースの中で管理するのが望ましい。

以上、要件を整理すると Web ページの統合管理のためには単に URL を保持するだけではなく、次のような要件を満たさなければならないことが分かる。

- (1) 利用する Web ページを限定すること（大前提）。
- (2) Web ページをハイパーフレームと同一の形式で属性管理すること。
- (3) Embedded links によるハイパーリンクをハイパーフレーム間の関連と同一の形式で管理すること。
- (4) Web ページを部品化するための情報とそのアルゴリズムを管理すること。
- (5) 個々の教材部品が各々に適したクライアント・

^{*} 明確な学習目標に基づいた学習では、学習者の主体的なナビゲーションといえども、学習者があたかも自由に利用しているように感じられるように綿密に計算された教材空間の中で行わせるべきである。

^{**} たとえば、Web ページ作成者の用意したメニュー・ページや趣味・娯楽のページなど。

ソフトウェアに関する情報を管理すること。

3.2 Pspecs の機能拡張による Web ページ統合モデル

統合モデルは基本的に文献 22) で提案した拡張 Dexter モデルに基づくものである (図 1 参照)。

Dexter モデル自体は、3つの層と2つのインタフェースから構成されたハイパーテキストの参照モデルである。Storage layer でノードとリンクのネットワークを管理し、ノードのコンテンツを within-component layer で分離して管理するところに特徴がある。ノードのコンテンツに対するアンカーの指定は anchoring インタフェースで行う。一方、ハイパーテキストとユーザとの対話は run-time layer で扱う。このとき、ハイパーテキストの表示方式は storage layer と run-time layer とのインタフェースである presentation specifications (以降、pspecs と略す) で指定する。

しかし、Dexter モデルは複数のハイパーテキスト間にまたがるハイパーリンクを扱うには不十分であり、そのままではハイパーテキストが分散した環境での相互利用を実現するのは困難である。拡張 Dexter モデルはこの問題点を解決するために、storage layer と pspecs の間に link-context layer と context selecting インタフェースを挿入し拡張したモデルである。

さて、Dexter モデルに従うと、ハイパーフレームと Web ページとの違いは within-component layer によって覆い隠されるので、ノードはコンテンツの種別の違いを意識する必要がなくなる。すなわち、そのコンテンツがハイパーフレームであれ Web ページであれ、ノードとして抽象的に扱うことができるようになる。なお、ハイパーフレームを管理するための within-component layer のオブジェクトは、その実体へのポインタ情報を保持すればよい。また、Web ページを

管理するための within-component layer のオブジェクトは、その属性として URL を保持し、メソッドとしてその実体をダウンロードするための処理を用意すればよい。

さらに、Dexter モデルではノードに対し、属性名と属性値の対として任意の属性をインクリメンタルに設定することができる。これを利用して学習支援に必要な項目をノード属性として定義しておけば、それらの項目を利用した集成的な検索処理を双方の文書すべてを対象に実現できる。付与すべき属性はすでに文献 5) で検討されている。このように、Dexter モデルに基づくことで先に述べた要件 (2) は満たされる。

ところで、要件 (1)、すなわち Web ページ統合の大前提である Web ページの限定は、教材作成者が芽づる式に連結しているハイパーリンクをある範囲で抑制することを意味しており、任意の embedded links に対する隠蔽機能によってはじめて実現できる。しかも、もともとの文書上の embedded links を削除するのではなく、教材部品として提示するときのみに HTML による記述を動的に削除処理しなければならない。また、要件 (3) は HTML による embedded links を拡張 Dexter モデルのリンクメカニズムで置き換えることを意味している。この要件を満たすためには、後述するように embedded links のリンク先を拡張 Dexter モデルに基づくリンクメカニズムを実現するプロセスに変更するアプローチが有効であり、embedded links に関する HTML による記述内容を動的に変更する機能が必要である。要件 (4) も同様に HTML による記述内容の動的な操作を意味している。CAI 教材データベースへの要件を実現するには、Web ページを提示する際にもともとのコンテンツを加工する処理、つまりビューを実現するための機能が重要な鍵となる。要件 (5) はまさにビュー機能そのものに依存する。

そこで、Dexter モデルにおいてビューメカニズムの実現を司る pspecs の機能を拡張することで、上で述べた要件を満足するモデルを提案する。

この統合モデルは、データだけではなく実行時の処理をも管理できるようにオブジェクト指向方法論に従ってフレームワーク化されている。図 2 にフレームワークのオブジェクト図を OMT 法¹⁰⁾により示す。

また、表 1 に Web ページの利用に関して関連研究と我々のアプローチとの比較を示す。

以下で、pspecs の機能拡張の詳細とフレームワークに基づいた embedded links によるハイパーリンクの実現方法について述べる。

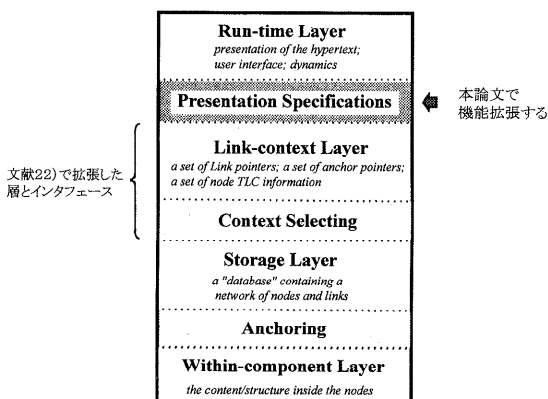


図 1 拡張 Dexter モデル

Fig.1 Extended Dexter model.

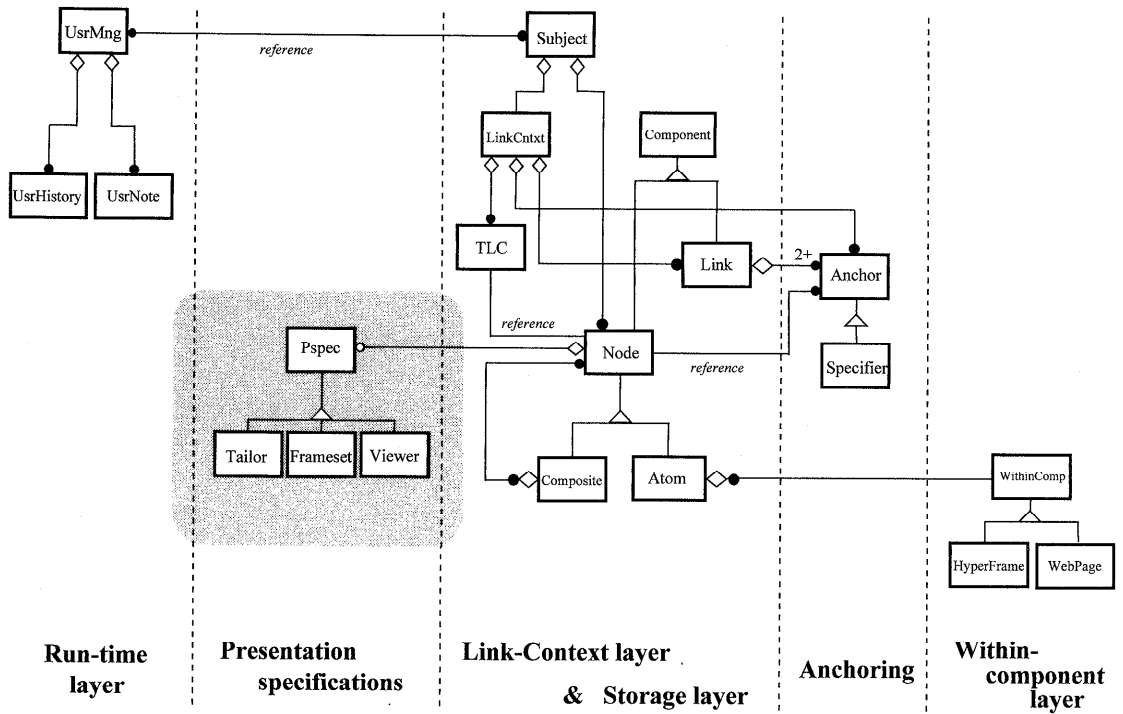


図2 統合モデルのオブジェクト図
Fig.2 Object diagram of the integration model.

表1 Web ページ利用に関する関連研究との比較
Table 1 Comparison with related systems.

関連研究	属性管理の形態	Embedded links の利用形態	部品化機能	他文書の提示	利用範囲の限定
Walden's Paths ⁸⁾	なし	手を加えず利用	追加・変更・部分削除	なし	なし(ただし、管理外のページからの復帰を支援)
CALAT ¹⁶⁾	なし	一部のリンク先を動的に書換え	なし	plug-in による専用ビューア	なし(ただし、管理外のページからの復帰を支援)
Didactic Network ¹⁾ DHM ¹⁰⁾	メディア型を付与 任意の数の属性名と属性値の対を付与	手を加えず利用 リンクの追加が可能	なし なし	専用ビューア(Webブラウザの制御を行う) メディアごとに専用ビューア	なし なし
Ours	任意の数の属性名と属性値の対を付与	リンクの追加, 削除およびリンク先変更が可能	追加・変更・部分削除, および合成	ハイパーフレームの種類ごとに専用の Java Applet	Embedded links の削除, リンク先変更により限定

3.2.1 Pspecs の機能拡張

不適切な embedded links の隠蔽も含め、Web ページを提示する際の embedded links に関する HTML 記述の変更やその他のコンテンツに対する追加・変更・削除、および Web ページ合成のための情報は、Dexter モデルに従うと pspecs として管理すべき情報である。Pspecs にはノードを提示する際に利用するプログラムの指定、window サイズや提示位置などを管理するものととらえられるが、Dexter モデルでは pspecs の

詳細について述べていない。また、AHM¹²⁾ではマルチメディア情報の時間的な関係や提示環境を管理できるように pspecs に詳細な機能拡張を施している。我々は、Web ページを統合管理するために図3のような pspecs の機能拡張を提案する。

(1) Pspec クラス

Pspecs を表現するための抽象クラスであり、ハイパーフレームと Web ページの統合管理を具体的に実現するクラスのスーパークラスである。教材部品提示用の

共通インタフェースを仮想関数として定義している。

(2) Tailor クラス

Web ページのコンテンツ (HTML による文字列) の追加, 変更, 削除のための情報を管理するとともに, それぞれの処理のアルゴリズムをメソッドとして実装する。Web ページへの導入文やコメントの追加を行うためには, 追加する文章と追加する場所 (Web ペー

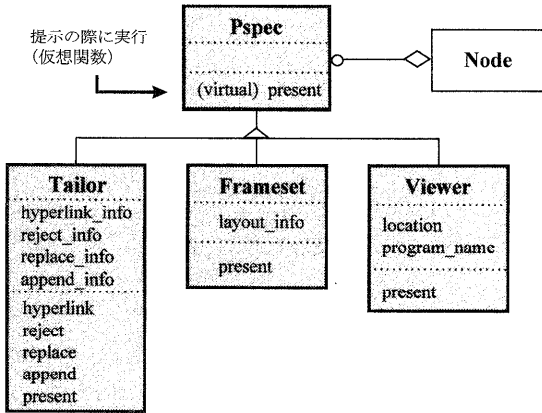


図3 Pspecs のクラス構成

Fig.3 Class hierarchy of pspecs.

ジの先頭もしくは最後) が管理される。用語の統一など, 文字列の一括変換を行うためには, 対象となる文字列と変換後の文字列が管理される。

また, HTML による embedded links 記述の変更, 隠蔽のための情報やその実現アルゴリズム (後述) も管理する。Embedded links の提示形態としては, 何ら変更を加えずに提示する場合 (ページ内へのリンクがこの形態として利用できる), リンク先を変更して提示する場合, ホットスポットとなる文字列のみ提示してリンク機能は隠蔽する場合, ホットスポット文字列も隠蔽する場合, の4通りある。Web ページ上のすべての embedded links についていずれかに対応した情報を管理する。この場合, Web ページ上の変更場所 (当該 embedded link の記述位置) を特定する情報としてもとの URL 文字列を保持するとともに, たとえばリンク先を変更して提示する形態の場合には, 変更後の URL も管理する。この変更後情報に対して, null ならば変更を加えない, 空白ならば隠蔽する等の意味を持たせることで他の3種類の指定についての情報も管理できる。図4に ODMG-93³⁾の ODL (Object Definition Language) に従ってオブジェク

```
interface HyperlinkInfo : persistent {
    attribute String old_url; //変更対象URL
    attribute String new_url; //置き換えURL
    relationship Tailor owner inverse Tailor::hyperlink_info;
};

interface RejectInfo : persistent {
    attribute String start_pos; //start_pos::=<開始基準文字列>,<出現順位>,<オフセット>
    attribute String end_pos; //end_pos::=<終了基準文字列>,<出現順位>,<オフセット>
    relationship Tailor owner inverse Tailor::reject_info;
};

interface ReplaceInfo : persistent {
    attribute String old_str; //変更対象文字列
    attribute String new_str; //置き換え文字列
    relationship Tailor owner inverse Tailor::replace_info;
};

interface AppendInfo : persistent {
    attribute String position; //position::=<基準文字列>,<出現順位>,<オフセット>
    attribute String str; //追加文字列
    relationship Tailor owner inverse Tailor::append_info;
};

interface Tailor : Pspec : persistent {
    relationship List<HyperlinkInfo> hyperlink_info
        inverse HyperlinkInfo::owner; // embedded links置換データ
    relationship List<RejectInfo> reject_info
        inverse RejectInfo::owner; // 部分削除データ
    relationship List<ReplaceInfo> replace_info
        inverse ReplaceInfo::owner; // 文字列置換データ
    relationship List<AppendInfo> append_info
        inverse AppendInfo::owner; // 文字列追加データ

    String hyperlink(inout String html); // embedded links置換処理
    String reject(inout String html); // 部分文字列削除処理
    String replace(inout String html); // 文字列置換処理
    String append(inout String html); // 文字列追加処理
    String present(inout String html); // 飾品化処理
};
```

図4 Tailor クラスのオブジェクト・モデル

Fig.4 Object model of Tailor class.

Web ページを対象にした属性値による集学的検索、およびナビゲーション（構造的検索）の統合利用の実現を検証するもので、学習者に適応的に教材を提供する機能は実現しない。しかしながら、知的マルチメディア CAI における適応的な教材提示機能は、学習者の特性や状態情報をもとに教授方略のアルゴリズムによって、適切な学習項目、内容の難易度、提示メディア型などの属性値を決定し、データベースから対応する教材部品を検索することで実現している⁵⁾ため、試作システムの範囲で知的マルチメディア CAI で利用する教材データベースの基本機能の実現に関する本モデルの妥当性を確認できる。

4.2 実装の前提条件

システムのプラットフォームに WWW を採用する。学習者は通常の WWW ブラウザを通して利用するものとする。したがって、Web ページはその WWW ブラウザ上に直接提示する。

他方、ハイパーフレームはその対話性の高さから、専用のクライアント・ソフトウェアが必要であり、ここでは Java Applet の利用を前提にした。そのため、Viewer オブジェクトにハイパーフレームの提示情報として、利用すべき Java Applet* の名前と格納場所を、またその Java Applet を呼び出すための HTML ストリームを出力するメソッドを管理する。

4.3 システム構成

試作システムの全体構成を拡張 Dexter モデルと対応付けて図 6 に示す。

ハイパーフレームの実体を管理するデータベースと Web ページを取得するサーバ・プロセスが within-component layer に相当する。また、IU (Instructional Unit) サーバはハイパーフレームもしくは Web ページをカプセル化した教材部品を対象に、属性値を利用した集学的検索処理や教材部品間のナビゲーションを実現するための CGI プログラムの集まりである。学習者は WWW ブラウザを利用して IU サーバにアクセスし、教材部品を利用する。

教材部品提示の際には、ノードに対応した within-component layer のオブジェクトがハイパーフレームの場合は Hyper-frame DB を検索し、また、Web ページの場合には URL 情報によって対応する WWW サーバにアクセスすることで、それぞれの形式のコンテンツを取得する。コンテンツの提示は、ハイパーフレームの場合には、まず Viewer オブジェクトによ

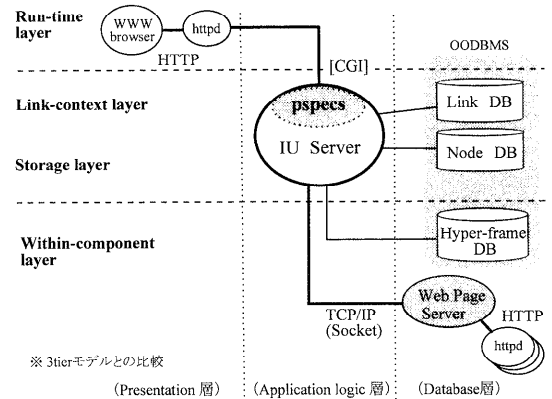


図 6 試作システムの全体構成

Fig. 6 Software architecture of our prototype system.

て、その提示に適した Java Applet を呼び出すための HTML ストリームを WWW ブラウザに送る。その後、Live Connect 機能を通じて Java Applet を取得したうえで学習者に利用される。Java Applet は Web サーバ上に置かれている。コンテンツが Web ページの場合には、Tailor オブジェクト、または Frameset オブジェクトによってビューとして加工され、HTML ストリームの形式で CGI によって WWW ブラウザに送られる。

このように、拡張した pspecs 機能は IU サーバ上で作用する。なお、本システム構成は 3tier モデルとしてとらえることもできる。

5. 適用例

高度情報処理教育を学習のドメインとし、インフォメーション・エンジニアリング (IE: Information Engineering)¹⁵⁾の教材部品を実装した。IE を学習するうえでの本質的な項目についてはハイパーフレームを開発し、IE を実践する際の基盤技術や関連する方法論などは WWW 上に公開されている Web ページを利用した。現在、30 件程度の Web ページを登録し、そこに設定されている約 170 の embedded links のほぼ半数を利用可能としている。

IE 学習における Web ページの利用イメージを図 7 に示す。図 8 は属性値による教材部品の検索例である。ハイパーフレーム、Web ページを問わず、検索された教材部品がそのタイトルとして提示されている。図 9 はその中から Web ページによる教材部品を選択したときの画面例である。

なお、図 10 はそのもとの Web ページである。このページには著者の文脈に依存したアイコン、すなわち上位のメニューに復帰するリンクが埋め込まれて

* ハイパーフレームの種別として説明教材、問題教材、シミュレーション教材があり、それぞれ対応する Java Applet が用意される。

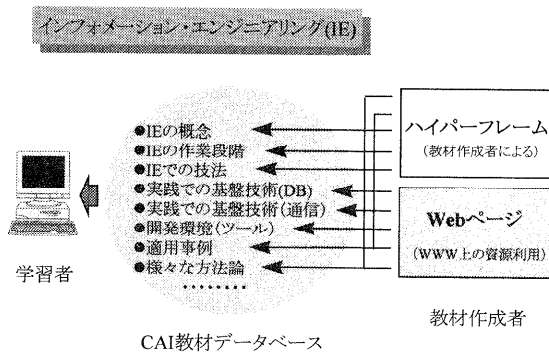
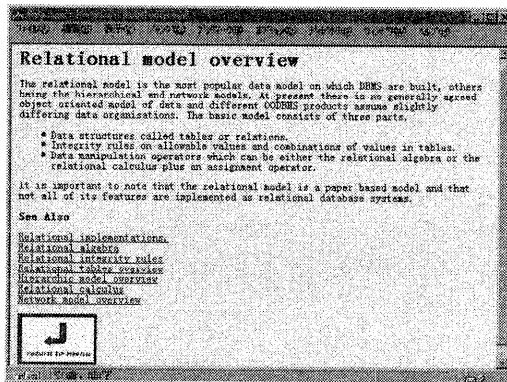


図7 学習における Web ページの利用イメージ

Fig. 7 An example use of Web pages for IE learning.



利用元: http://www.compapp.dcu.ie/databases/f037.html

図10 もともとの Web ページ

Fig. 10 Original Web page.

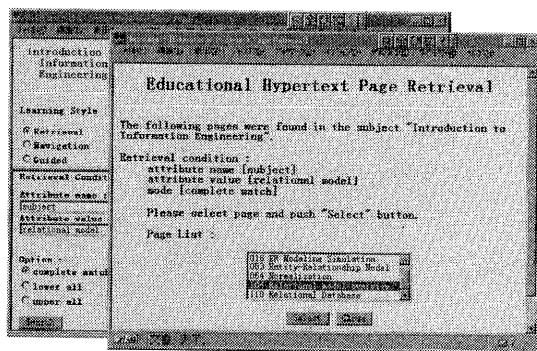


図8 教材部品の検索例

Fig. 8 An example screen for retrieval of instructional units.

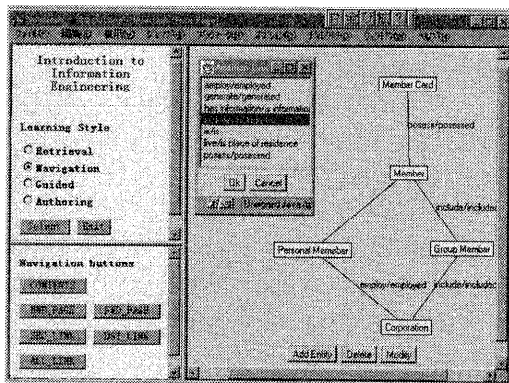


図11 ハイパーフレームの提示例

Fig. 11 An example screen for Hyper-frame.

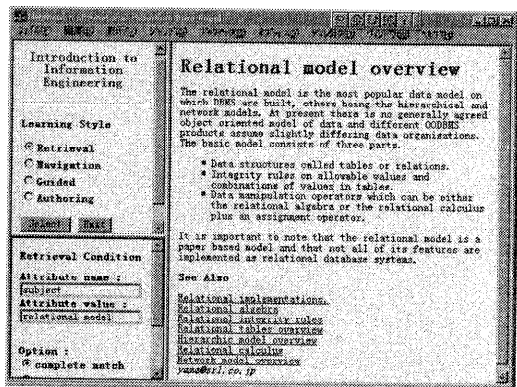


図9 Web ページの提示例

Fig. 9 An example screen for Web page.

いるが、学習で利用するときには隠蔽されていることが分かる。

また、図11にハイパーフレームの提示例を示す。これは企業の業務領域の分析方法を学習するためのシミュレーション教材の1つである。学習者はハイ

パーフレーム上の編集機能を利用してER図(Entity-Relationship diagram)を作成することで業務領域分析の一端を体験できる。ER図としての作図上の制約が教材知識としてハイパーフレームにカプセル化されている。

6. 考 察

今回の適用例からは、Webページをハイパーフレームと分け隔てなく扱うという機能の実現だけでなく、Webページを利用することによる教材部品の開発コスト削減という本来の研究目的に対する効果という点でも評価できた。

テキスト、静止画、音声を利用したハイパーフレーム(説明教材型)は1日に3件程度開発できた。データベースへの登録はバッチ処理的に行っているが、そのためのバッチファイル作成にはほとんど時間はとれない。費やした時間のほとんどは構想と素材データの作成のための時間であった。文献5)、7)での経験で

は、1日1、2件程度という実績であった。他方、Web ページの場合には、そのコンテンツに対するビュー（どのような追加、変更、削除を行うかという操作情報）の検討、および検討結果のバッチファイル化による登録を1日10件程度処理できた。ちなみに、この作業においてビューの検討とデータベース登録には同程度の時間を費やした。こうした実績から、バッチ処理的な教材登録という環境において、教材部品の開発コストをおおよそ1/3程度に削減できたと評価している*。

しかし、教材部品の登録過程をより支援することによって、開発コストを下げる余地が残されている。Web ページの登録では、教材作成者が1ページごとにその内容を確認して、必要ならば手を加えるという作業をせずに済ませることはできないが、ハイパーフレームと異なり、1回の作業単位で embedded links によって互に関連し合う複数のページを登録するのが一般的である。なぜなら教材作成者は、もともとの embedded links を次々とたどっていきながら、Web ページの取捨選択を行い、利用できる Web ページの範囲を決定するからである。こうした特性を活かし、登録しようとする全 Web ページを対象に、ある用語に対して文字列の一括変換を行うとか、相互に関連し合う embedded links を自動的に検出し、一括して変更情報を登録する、といったサポートが考えられる。このようなオーサリング環境はシステム化における今後の課題である。

その他にも取り組むべき課題が残されている。まず、もともとの Web ページの変更に対し、ビュー機能の正常な動作を保証できない場合が残っているという点がある。Embedded links への変更や文字列の一括変換は、文字列のパターンマッチングに基づいているため、もともとの Web ページの記述内容の変更に対応できる。しかし、文章の追加、削除、ページの分割については、あらゆる場合についての保証は難しい。これは、ビュー操作の対象となる個所の指定が Web ページ自身の記述に物理的に依存する（たとえば、オフセット値の利用など）限り本質的には避けられないことによる問題である。試作システムでは、文章の追加を Web ページの先頭、もしくは最後限にのみ限定して対応している。この問題は文書上のアンカー領域の指定方法に関する議論^{4),9)}でもなされており、今後の検討課題である。なお、Web ページの URL の変更や Web ページ

自体の削除については、WWW サーバからのエラー情報を受け取ったときにノード識別子や URL を保持しておき、オーサリング環境で対応するといった対策が考えられるが、今回は Web ページ上での内容変更までを考慮の対象として取り組んでいる。

次に実装上の問題であるが、試作システムでは学習者（クライアント・プログラム）とデータベースとの通信に CGI を利用した HTTP（HyperText Transfer Protocol）ベースの実装となっているために、データベースのトランザクション管理のためのオーバヘッドが大きく処理効率が悪いこと、知的マルチメディア CAI で必要な学習者のセッション管理が複雑になる、といった問題点を解決する必要がある。そのアプローチとしては、現在分散オブジェクト技術として標準化が進められている CORBA（Common Object Request Broker Architecture）の利用が有効と考えられる²⁰⁾。標準化の動向を見極めつつ、改善を検討したい。

7. おわりに

知的マルチメディア CAI の教材部品であるハイパーフレームと既存の Web ページとを Dexter モデルに基づいたハイパーメディア型データベースで統合的に管理するための要件を整理し、Dexter モデルにおける presentation specifications の機能を拡張することによって、Web ページを学習の文脈で利用するために必要な情報の管理方式を Web ページ統合モデルとして提案した。本論文で提案した統合モデルはオブジェクト指向に基づいており、様々な分野における Web ページの統合のフレームワークとしても有効である。

今後は先に述べた問題点の克服、実装方法の検討、オーサリング環境の開発を進めながら、統合モデルへのフィードバックを重ねていく予定である。

謝辞 日頃、ご指導、ご助言いただいている図書館情報大学教授増永良文氏に感謝の意を表します。

参考文献

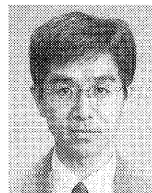
- 1) Baloian, N., et al.: Integration of external WWW-Documents into the Electronic Classroom, *Proc. ED-MEDIA 97 & ED-TELECOM 97*, Vol.I, pp.32-37 (1997).
- 2) Brusilovsky, P., et al.: ELM-ART: An Intelligent Tutoring System on World Wide Web, *Proc. 3rd International Conference, ITS '96*, pp.261-269 (1996).
- 3) Cattell, R. (Ed.): *THE OBJECT DATABASE STANDARD: ODMG-93 Release 1.1*, Morgan Kaufmann (1994). 野口喜洋ほか (訳): オブジェ

* 利用したい Web ページを教材作成者が用意していない場合には Web ページの検索作業が必要となる。しかし、Web ページの検索コストは検索エンジンの利用により今後ますます小さくなると期待できる。

- クト・データベース標準, ODMG-93 Release 1.1, 共立出版 (1995).
- 4) Davis, H.C., et al.: Light Hypermedia Link Services: A Study of Third Party Application Integration, *Proc. European Conference on Hypermedia Technology 1994*, pp.41-50 (1994).
 - 5) 富士 隆ほか: ハイパーフレームを用いた知的マルチメディア CAI の開発, 1994 年情報学シンポジウム講演論文集, pp.133-141 (1994).
 - 6) Fuji, T., et al.: Using Case-Based Reasoning for Collaborative Learning System on the Internet, *Trans. IEICE*, Vol.E80-D, No.2, pp.135-142 (1997).
 - 7) Fuji, T.: A Repository-based Approach to Reuse of Educational System Resources, *Proc. ED-MEDIA 97 & ED-TELECOM 97*, Vol.I, pp.378-383 (1997).
 - 8) Furuta, R., et al.: Hypertext Paths and the World-Wide Web: Experiences with Walden's Paths, *Proc. Hypertext '97*, pp.167-176 (1997).
 - 9) Grønbaek, K. and Trigg, R.H.: Toward a Dexter-based model for open hypermedia: Unifying embedded references and link objects, *Proc. Hypertext '96*, pp.149-160 (1996).
 - 10) Grønbaek, K., et al.: Designing Dexter-based hypermedia services for the World Wide Web, *Proc. Hypertext '97*, pp.146-156 (1997).
 - 11) Halasz, F. and Schwartz, M.: The Dexter hypertext reference model, *Comm. ACM*, Vol.37, No.2, pp.30-39 (1994).
 - 12) Hardman, L., et al.: The Amsterdam Hypermedia Model, *Comm. ACM*, Vol.37, No.2, pp.50-62 (1994).
 - 13) 畑田 稔, 遠藤裕英: WWW-RDB 連携システムの開発, 情報処理学会論文誌, Vol.38, No.2, pp.349-358 (1997).
 - 14) Kay, J. and Kummerfeld, R.J.: An Individualised Course for the C Programming Language, *Proc. 2nd International World Wide Web Conference* (1994).
 - 15) Martin, J.: *INFORMATION ENGINEERING Book1 Introduction*, Prentice Hall, Englewood Cliffs, N.J. (1989). 竹林則彦 (監修): インフォメーションエンジニアリング I 統合化 CASE のための方法論, トッパン (1991).
 - 16) 丸山美奈ほか: 分散教育環境 CALAT におけるハイパーリンクアーキテクチャの提案~知的ナビゲーションの実現手法について, 人工知能学会 SIG-J-9501-16, pp.104-111 (1995).
 - 17) Nielsen, J.: *MULTIMEDIA AND HYPERTEXT The Internet and Beyond*, Academic Press, New York (1995).
 - 18) 日経オープンシステム: WWW と DBMS の連携ソフトが出始めた, 日経オープンシステム, No.31, pp.240-245 (1995).
 - 19) Rumbaugh, J., et al.: *OBJECT-ORIENTED MODELING AND DESIGN*, Prentice Hall, Englewood Cliffs, N.J. (1991). 羽生田栄一 (監訳): オブジェクト指向方法論 OMT モデル化と設計, トッパン (1992).
 - 20) 鈴木純一: CORBA, Web, Java ORB — その統合と Web の可能性, *Computer Today*, No.77, pp.15-26 (1997).
 - 21) 田中克己: ネットワーク社会とマルチメディアデータベース, 情報処理, Vol.38, No.1, pp.24-29 (1997).
 - 22) 山北隆典ほか: Dexter モデルに基づくハイパーリンク・ビューの WWW 上での実現, 情報処理学会アドバンストデータベースシステムシンポジウム 96 論文集, pp.49-56 (1996).

(平成 9 年 8 月 28 日受付)

(平成 10 年 2 月 2 日採録)



山北 隆典 (正会員)

昭和 59 年弘前大学大学院理学研究科物理学専攻修士課程修了。同年 (株) エスシーシー入社。情報システムの設計, 開発に従事。現在, (株) 学習情報通信システム研究所第 1 研究室研究主幹。高度個別型学習情報通信システムの研究開発に従事。



富士 隆 (正会員)

昭和 44 年上智大学理工学部数学科卒業。同年安田生命保険 (相) 入社。情報システムの企画, 開発に従事。現在, (株) 学習情報通信システム研究所第 1 研究室長。平成 4 年より北星学園大学非常勤講師。高度個別型学習情報通信システムの研究開発に従事。IEEE, ACM, 電子情報通信学会, 人工知能学会各会員。



三枝 武男 (正会員)

昭和 18 年旧制都立高等工業学校電気工学科卒業, 東京工業大学留学。都立大学講師, 防衛大学校教授を経て名誉教授。平成元年より北海道情報大学経営情報学部教授 (学部長)。約 42 年間電子計測に関する研究に従事し, 現在, 高度個別型学習情報通信システムの研究開発に従事。工学博士 (東京工業大学)。IEEE, 電気学会, 電子情報通信学会, 計測自動制御学会, 応用物理学会など各会員。