

ハイパーメディア型CAI教材データベースの モデルとアーキテクチャ

山北隆典† 富士 隆† 三枝武男†

†学習情報通信システム研究所 †北海道情報大学

ある一つの分野を対象としたCAI教材は構造的に独立した局所的なハイパーメディアといえる。視野の広い学習を達成するためには、分野を越えた教材間に意味的な参照関係を設定し関連知識を提供する必要がある。そのためには、局所的なハイパーメディアが複数存在する環境において、複数のリンク設定者による独自の視点に基づいた参照関係を同じノードに設定したり、独自の参照関係から他の視点による参照関係へ移行できるようなハイパーメディア・モデルが必要になる。本稿では、意味的参照リンクをノードと区別して管理するリンクコンテキストと、リンクコンテキストの重ね合わせという概念を取り入れ、その要件を満たすためのメカニズムを示す。それに基づいて、Dexter Hypertext Reference Modelのstorage layerを、link context layer、context selecting、node-base layerに分割、拡張したハイパーメディア・モデルを考案し、ハイパーメディア型データベースのアーキテクチャの一例を示す。

The Model and Architecture of a Hypermedia CAI Material Database System

Takanori Yamakita† Takashi Fuji† Takeo Saegusa†

†Software Research Laboratory †Hokkaido Information University

We regard the CAI materials for a learning domain as a local hypermedia organized by structural links individually. In order to realize a learning with various viewpoints by using these hypermedia based on CAI materials, we propose the hypermedia model that handles the links between different local hypermedia semantically. There are two requirements for our hypermedia model. First, it can allow several authors to set anchors on the same node based on individual viewpoint. Second, it can allow them to change semantic relations on a node from his/her viewpoint to another one. For the purpose of fulfilling above two requirements, we adopt the link context that manages semantic links separated from nodes, and a notion of overlapping these link contexts. By dividing the storage layer of the Dexter Hypertext Reference Model into link-context layer, context selecting and node-base layer, we design a new hypermedia model and present an example of a hypermedia database architecture based on this model.

1 はじめに

高度個別型CAIの要素技術の一つとして、教材を教え方から独立させ、かつ教材を部品化するハイパーフレーム [富士94] を提案してきた。CAI教材データベースとは主にハイパーフレームの管理を目的としたデータベースである。その実現にはハイパーメディア型データベースによるアプローチが有効であると考え、Dexter Hypertext Reference Model[Hala94] (以降、Dexterモデルと略す) を基本モデルとするデータベースの検討を続けている。

ハイパーメディアはノードとリンクにより、階層型やネットワーク型の関係を表現できる情報管理形態である。リンクはノード間の論理的な構造を示す構造的リンクと意味的な関連付けを示す意味的参照リンクに分類できる。ハイパーメディアは個々の情報単位や関連を個別に管理するのに有効であり、リンクを用いた2通りの関連付けはCAI教材の構造やハイパーフレーム間の参照関係の表現に適している。本稿では、現在検討中のハイパーメディア型データベースのモデルの概要と、その特徴の一つである意味的参照リンクの取り扱いについて述べる。

2 CAI教材データベースの要件

2.1 個々のCAI教材に関わる要件

はじめに、一つのCAI教材をハイパーメディアとみなし、Dexterモデルの適用を検討した。

CAI教材の分析結果 (図1参照) から次のような要件を洗い出し、各々解決策を導いてきた。

まず、ノードのコンテンツとしてマルチメディアデータだけでなく教材知識も扱うことから、マルチ

メディアの提示の仕方 (位置関係やタイミング) だけでなく教材知識とのやり取りの仕方も教材作成者の意図としてモデル化し、データベース内に統合的に管理するという要件がある。そこで、Dexterモデルの presentation specification に対し、教材知識と学習者との質問/応答といった双方向の対話仕様を記述できるように拡張する。

また、ハイパーフレーム内の階層構造や学習テーマが作り出す階層構造を表現するために、単なる包含関係の表現 (Dexterモデルの composite component) の他、子ノードの間に順序関係を設定できるタイプ (順次型)、子ノードにキーワードを設定し、親ノードがそれをキーにして子ノードを選択できるようなタイプ (選択型) が必要であるという要件もある。これに対しては、composite component の代わりに、「包含型」、「順次型」、「選択型」の3種類の複合ノードを導入することで対応する。

2.2 複数のCAI教材の扱いに関わる要件

ところで、教材としてはその目的の分野だけでなく、関連知識のように周辺分野に関する知識を提供することも、より視野の広い学習を達成する上で重要な要素になる。一般にCAI教材はその分野の専門家によって作成されるが、限られた時間や労力の制約の中で、そのような専門外の分野に関する良質なハイパーフレームを作り、教材に組み入れることは非常に困難である。他の周辺分野の専門家が作成した良質なハイパーフレームを利用することはこの問題を解決する一つの手段を提供する。

このとき、他の教材のハイパーフレームにも教材作成者自身の所有するハイパーフレームと同様に、参照関係を設定していく利用形態と、同じように教材作成者自身の所有するハイパーフレームから他の教材のハイパーフレームへの参照関係を設定するが、そのハイパーフレームから先の参照関係は他の専門家が設定したものに依存するといった2通りの形態が考えられる (その組み合わせもある)。

Dexterモデルでは、複数のリンク設定者による独自の参照関係を同じノードに対してそれぞれ設定したり、独自の参照関係から他者の参照関係に移行したりといった取り扱いは考慮されていない。以下ではこうした意味的な参照関係を表現する意味的参照リンクに関わる上記の要件の解決に焦点を絞り、ハイパーメディア・モデルについて検討する。

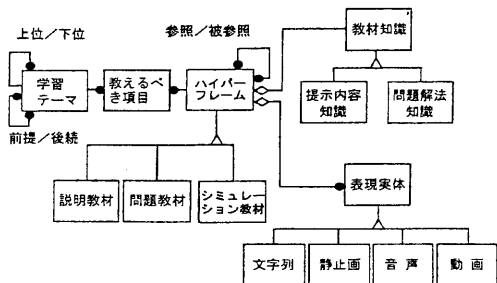


図1: CAI教材データベースのオブジェクト図

3 ハイパーメディアモデル

3.1 基本的アイデア

まず、一つのCAI教材は、複合ノードによる静的な階層構造と、学習テーマとハイパーフレームによる「教えるべき項目」をキーとして動的に決まる構造とが、一つのメモリ空間上に閉じている局所的な(独立した)ハイパーメディア(Local Hypermedia: LHと略す)であると考えられる。このようなLHが、メモリ空間を越えて複数存在する。

さて、先の意味的参照リンクの要件から、まず、LHの枠を越えたノード間に意味的参照リンクを、個々のLHの独立性を犯さぬよう、それによる影響を与えずに、あらゆるノードを対象に設定できなければいけない。そのために、いわゆるハイパーメディアにおけるビューの考え方を適用する。

ビューを実現するポピュラーな手段としてはリンク情報をノードと区別して管理する方法[Haan92]がある。そこで、意味的参照リンクの集合とアンカーの集合からなるリンクコンテキストを導入し、一つのLHをノード集合とリンクコンテキストの組みで管理する。

次に、異なる作成者による参照関係への移行、すなわち他のリンクコンテキストへの移行を実現するには、一つのノードに対して意味的参照リンクを設定している複数のリンクコンテキストを、同時に利用できる必要がある。これをリンクコンテキストの重ね合わせと呼ぶことにする。

ノードを扱う時に、一つのリンクコンテキスト内の意味的参照リンクだけを利用し、別のリンクコンテキストが設定した意味的参照リンクを無視するか、複数のリンクコンテキストの意味的参照リンクを同時に利用するかは意味的参照リンクの設定者の判断であり、リンク設定時に決定される(図2参照)。すなわち、どのリンクによってノードにトラバースしてきたかによって、そのノードに設定されているアンカーを得るためのリンクコンテキストを、設定者の意図どおり適切に選択する必要があり、そのためのメカニズムとしてcontext selectingを導入する。

なお、構造的リンクはLHの独立性を保つために、ノード集合の中で(複合ノードもしくはキー属性による検索機能により)扱う。

ビューは、同じノード集合に対しリンクコンテキ

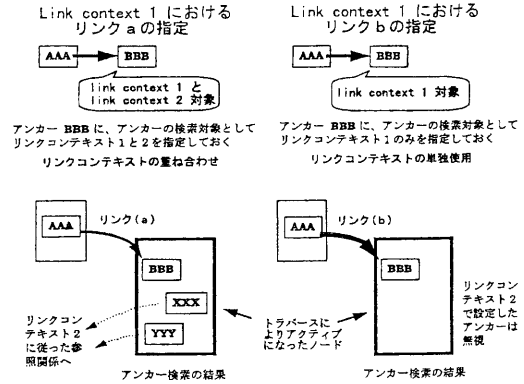


図2: リンクコンテキストの重ね合わせ

ストを切り替えることで実現する。しかし、本モデルの考え方は、ノード集合が構造的リンクを扱った独立のハイパーメディア(LH)であること、その上で、一つのリンクコンテキストがあらゆるノードに対して意味的参照リンクを設定できるだけでなく、他のリンクコンテキストを重ね合わせることで意味的参照リンクのビューの機能を、重ね合わせや移行の実現によって高める点にある。

3.2 ハイパーメディアモデルの概要

本モデルでは、Dexterモデルのstorage layerを、ノードを扱う層(node-base layer)とリンクを扱う層(link-context layer)に分け、そのインタフェースとしてcontext selectingの概念を導入する(図3参照)。その各々について、及びanchoring interfaceについて簡単に述べる。なお、presentation specificationについてはすでに文献[山北96]で述べている。

3.2.1 Node-base Layer

Dexterモデルのatomic componentとcomposite componentにあたる、基本ノードおよび複合ノードの集合を記述する層である。基本ノードはwithin-component layerのデータのコンテナであり、複合ノードは基本ノードもしくは他の複合ノードのコンテナである。また、Dexterモデルと同様に、任意の属性、属性値の対を指定できる。複合ノードは「包含型」、「順次型」、「選択型」の3種類の型を持つ。構造的リンクはこの層で扱う。基本ノード、複合ノードともにLH内で一意な識別子を持つ。

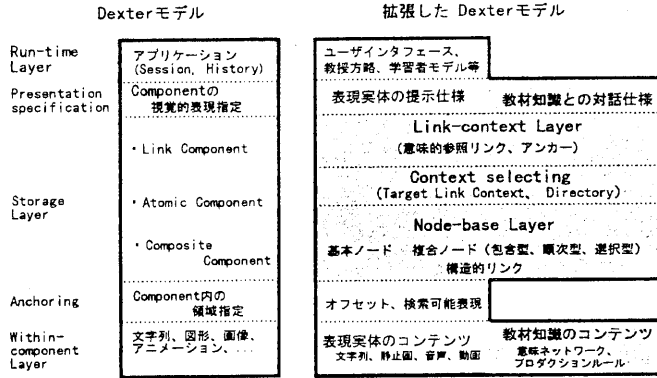


図 3: Hypermedia model

3.2.2 Link-context Layer

リンクコンテキストを管理する層である。リンクコンテキストは一つのLHに対して複数定義してもよく、LH内でユニークな識別子を持つ。また、意味的参照リンクとアンカーはリンクコンテキスト内でユニークな識別子を持つ。なお、ここで扱う意味的参照リンクとはアンカーとアンカーの間を結ぶ2項関係であり、意味的には向きをもっているがどちらへも辿ことのできる双方向リンクである。m対nの関連付けも設定できる。意味的参照リンクには属性、属性値の対の設定もできる。

ノードに設定されているアンカーを知るにはリンクコンテキスト内のアンカー集合を検索する。

3.2.3 Anchoring Interface

アンカーはノード自身にも、そのコンテンツの一部にも設定できる。コンテンツへの設定では、アンカーへのポインタや識別子をノードに埋め込まず、ノードの識別子とノード内のホットスポットへのオフセット表現、もしくはサーチエンジンによる検索可能なある種の表現¹を用いてアンカーとノードの対応をとる [Davis95] ことでノードの独立性を守る。

3.2.4 Context Selecting Interface

Context selectingの前提として、あらゆるノードを識別するためのメカニズムが必要である。そこで、各LHに、対象とする世界で一意的ハイパーメ

¹例えば、テキストデータの場合、アンカーの設定領域として文字列を定義すれば、パターンマッチングによりノード上の対応するホットスポットを特定できる。

ディア識別子を定義する。また、ハイパーメディア識別子とリンクコンテキスト識別子をそのLHを扱うホスト計算機の識別子と対応付ける、ディレクトリ情報を管理する。

Context selectingのメカニズムの実現法の一つとして次のようなデータを定義する。

まず、リンク設定者の意図を表現するため、リンクが保持する両アンカーの識別子と共に、各々について、そのアンカーに達することでアクティブになるノードがアンカー検索の対象とすべき、リンクコンテキスト (Target Link Context: TLCと略す)の集合²を保持する。

さらに、各ノードに対し、TLCの集合を保持するためのデータ領域を定義する。ノードに意味的参照リンクのアンカーを設定するたびに、そのアンカーに対応する先に定義したリンクのTLC集合の要素をこのデータ領域に加えていくことで、そのノードにアンカーを設定しているすべてのリンクコンテキストを知ることができる。しかし、このままでは他のユーザが設定したTLCについてもアンカー検索の対象としてしまう。そこでTLC集合のうち、どれが現在意味を持ち得るかを非永続的な状態フラグ(真偽値データ)として実行時のインスタンス上で保持する。このフラグは、意味的参照リンクが定義されたとき、ナビゲーションによってノードがインスタンス化されたとき、及び明示的にリンクコンテキストを指定してノードを検索しインスタンス化したときに真となる。

²より正確にはTLCの識別情報の集合である。識別情報とはTLCの属するLHの識別子とその中で一意なTLCのリンクコンテキスト識別子の組みである。

4 アーキテクチャ

4.1 意味的参照リンク

意味的参照リンクの仕組みの概略を、ナビゲーションを例に図4に従って示す。

1. ノードのTLC集合内で現在有効なTLCの識別情報、すなわちハイパーメディア識別子(HMid)とリンクコンテキスト識別子(LCid)を使ってディレクトリ情報を参照する。
2. リンクコンテキストの存在するホスト計算機名(Host)を取得し、LCidをキーにリンク情報、及びアンカー情報を識別する。
3. 識別したアンカー情報を当該ノードのHMidとノード識別子(Nid)、及びホットスポット(extent)で検索しアンカーを特定する。
4. アンカー識別子(Aid)をキーとしてリンク情報を検索し、リンクを特定する。
5. リンクが特定できたら、反対側の端点に相当するアンカーのAidに付随したTLC集合の要素(HMidとLCid)を使ってディレクトリ情報を参照する。
6. リンクコンテキストの存在するHostを取得し、LCidをキーにアンカー情報を識別する。
7. 識別したアンカー情報を、先に得た「反対側の端点に相当するアンカーのAid」で検索し、アンカーを特定する。
8. アンカーにある、ノードのHMidによりディレクトリ情報を参照する。
9. Hostを取得し、ノード情報を特定する(LHが決まればノード情報も一意に決まる)。
10. ノード情報からアンカーにあるNidをキーにノードを取得し、アンカーのextentデータからホットスポットを特定する。

参照一貫性の維持に対する基本方針は、形式的な一貫性(つまりdangling linkの抑制)はシステムが管理する。しかし、ノード内容の変更などで発生する意味的な不整合は、その可能性を知った時に意味的参照リンクの設定者(教材作成者)へ通知し、判断を仰ぐものとする。

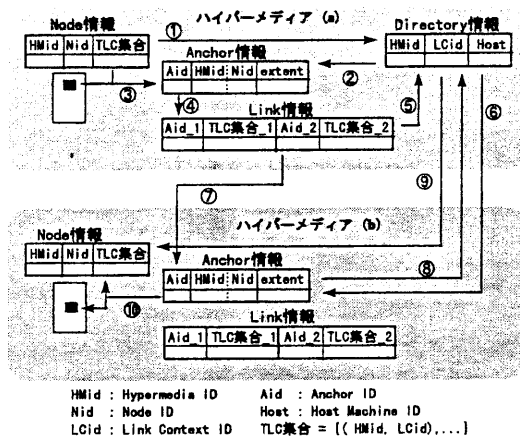


図4: 意味的参照リンクの仕組み

意味的参照リンクによる関連はリンクコンテキストとノード、およびリンクコンテキスト間に生じる。そこで、次のようなイベントに基づいた一貫性維持のためのアルゴリズムを定める。

1. リンクコンテキストは、ノードと同様に、その中で管理しているリンクに設定しているTLCのすべての和集合を保持する。
2. ノードやリンクコンテキストに対して内容変更、削除操作を行ったときには、それを参照しているノード、リンクコンテキストにイベントとして通知する。
3. 通知を受けたノードやリンクコンテキストは、イベントに応じた処理(形式的な一貫性を維持する処理や意味的な一貫性が崩れる可能性についての作成者への通知)を実行し、結果を通知元へ送信する。
4. イベントの通知が正常に完了しないときは、当該イベントの履歴を管理しておき、異常の復旧後、通知のみを再送信する。

イベントの種類としては、参照の開始/終了、内容変更、削除がある。また、イベント通知が不成立の際の履歴データとしては、イベントの発生時刻と種別、対象オブジェクト、通知先、さらに複数回再送信を行う場合や他のイベントとの整合性を考慮して、最新通知時刻も保持する。

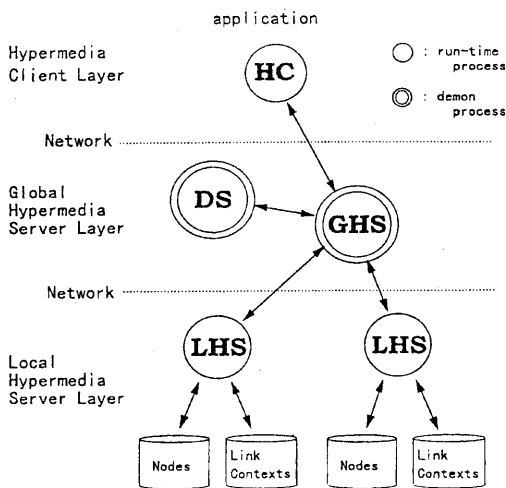


図 5: システムアーキテクチャ

4.2 3階層のアーキテクチャ

システム・アーキテクチャの一例として、ハイパーメディア・クライアント(Hypermedia Client:HCと略す)、グローバル・ハイパーメディア・サーバ(Global Hypermedia Server:GHSと略す)、ローカル・ハイパーメディア・サーバ(Local Hypermedia Server:LHSと略す)からなる3層アーキテクチャを図5に示す。GHSの層には、先に述べたディレクトリ情報を一元管理するディレクトリ・サーバ(Directory Server:DSと略す)を含む。LHSには商用のデータベース管理システムを利用する。HCはアプリケーションである。

5 関連システムとの比較

Aalborg大学のHyperBase[Wiil93]やTektronixのHAM[Camp88]はすべてのノードを一つの計算機上で集中管理している。また、Intermedia[Haan92]やDGS[Shack93]ではLAN内に分散して存在する計算機上のノードを一つのハイパーメディアの資源として透過的に管理している。いずれも異なるメモリ空間で構造的に独立した複数のハイパーメディアを意味的参照リンクで関連付けるといったアプローチは取られていない。SP3[Leg94]は遠隔地に存在する私的なハイパーメディアと一個所のホスト計算機が管理する公的なハイパーメディアを対象としているが、それらの間をコンテキストスイッチという

仕組みで切り替えて使用できる環境を実現するものである。また、WWWでは他の作者によるハイパーメディアのノード内にアンカーを設定することはできない。

6 今後の課題

今後の課題としてまず、CAI教材ではデータの追加に比べ、変更や削除の頻度はかなり少ないと考えられることから、教材内容に変更が発生しても版管理機能により旧バージョンでの一貫性を保持し、定期的に最新状態に移行するような一貫性維持の手法についても検討したい。

また、教材作成の過程におけるハイパーフレーム等の管理を考慮すると、ノードやリンクのインスタンスレベルでのアクセス制御や、教材作成を支援する協調作業環境についての検討も必要となろう。

参考文献

- [富士94] 富士 隆 他; 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
- [Hala94] F. Halasz and M. Schwarz; The Dexter Hypertext Reference Model, Comm. of the ACM, Vol.37, No.2, pp.30-39, Feb. 1994
- [山北96] 山北隆典 他; Dexterモデルに基づいたCAI教材データベースの検討, 情報処理学会 第52回全国大会講演論文集(4), pp.299-300, 1996
- [Haan92] B.J. Haan, P. Kahn, V.A. Riley, J.H. Coombs and N.K. Meyrowitz; IRIS Hypermedia Services, Comm. of the ACM, Vol.35, No.1, pp.36-51, 1992
- [Davis95] H. Davis; To Embed or Not to Embed..., Comm. of the ACM, Vol.38, No.6, pp.108-109, 1995
- [Wiil93] U.K. Wiil; Experiences with HyperBase: Hypertext Supporting Collaborative Work, SIGMOD Rec., Vol.22, No.4, pp.19-25, 1993
- [Camp88] B. Campbell and J.M. Goodman; HAM: A GENERAL PURPOSE HYPERTEXT ABSTRACT MACHINE, Comm. of the ACM, Vol.31, No.7, pp.856-861, 1988
- [Shack93] D.E. Shackelford, J.B. Smith and F.D. Smith; The Architecture and Implementation of a Distributed Hypermedia Storage System, Hypertext'93 Proceedings, pp.1-13, 1993
- [Leg94] J.J. Leggett and J.L. Scnase; Viewing Dexter with Open Eyes, Comm. of the ACM, Vol.37, No.2, pp.76-86, Feb. 1994