ハイパーメディア型のネットワーク構造に基づく記憶システムの設計と実装

片山 紀生　　高須 淳宏　　安達 淳

学術情報センター 研究開発部
〒 112 東京都文京区大塚 3-29-1
{katayama, takasu, adachi}@rd.nacsis.ac.jp

マルチメディアデータには、抽象データ型によるカプセル化の側面と、ハイパーメディアに代表されるネットワーク構造の側面とがある。前者については、既にオブジェクト指向モデルの有効性が示されているが、後者については一貫性管理など未解決な問題が多い。そこで我々は、ハイパーメディア型のネットワーク構造を対象とした記憶システムとして HyNeSS (Hyper Network Storage System) の開発を試み、その設計およびプロトタイプシステムの実装を行った。このシステムの特徴は、一貫性管理機構として、参照一貫性管理機構、導出機構、無矛盾性管理機構を備えている点であり、マルチメディアデータベースや電子図書館システムに応用できると考えている。

Design and Implementation of a Storage System based on Hypermedia Network

Norio KATAYAMA　　Atsuhiro TAKASU　　Jun ADACHI

Research and Development Department
NACSIS (National Center for Science Information Systems)
3-29-1 Otsuka, Bunkyo-ku, Tokyo 112 Japan

Multimedia data has two properties: one is encapsulation based on abstract data types, and the other is network structure which is seen in hypermedia. It is already known that the object-oriented data model is suitable for encapsulation. However, open problems still remain with respect to network structure. Then, we designed a storage system called HyNeSS (Hyper Network Storage System) and implemented its prototype system. The significant feature of HyNeSS is the incorporation of the integrity management mechanisms, namely, referential integrity, link derivation, and consistency management mechanisms. These mechanisms are advantageous to multimedia database and digital library systems.

1　はじめに

マルチメディアデータには二つの側面がある。ひとつは、抽象データ型によるカプセル化、もうひとつはハイパーメディアに代表されるネットワーク構造である。前者については、既にオブジェクト指向データモデルの有効性が示されているが、後者については未解決な問題が多い。特に、ネットワーク構造の一貫性管理については、部分的な実装は見られるものの一般的な枠組はまだ存在しない。

そこで、我々はハイパーメディア型のネットワーク構造を対象とした記憶システムの開発を試み、その設計およびプロトタイプシステムの実装を行った。この記憶システムの特徴は、ネットワーク構造を持ったデータのための、参照一貫性管理機構、導出機構、無矛盾性管理機構を備えている点にある。

本稿の構成は以下のようになっている。2 節では研究の背景として、マルチメディアデータに関する考察および関連研究の概略について述べる。3 節では提案する記憶システムの構成について、4 節ではそのプロトタイプシステムについて説明する。そして、5 節で今後の課題について検討する。

2　研究の背景

2.1　電子図書館システムにおけるマルチメディアデータ

学術情報センターでは学術文献を対象とした電子図書館システムの開発を行っており、現在試行サービスを実施しつつシステムの開発を続けている。この電子図書館システムが対象とするのは、学会誌、論文誌などの学術雑誌、講演論文集、研究会資料などであり、以下の三種類の情報を蓄積する。一次情報としてはページ画像のみを対象としており、現在の試行サービスでは全文情報は対象としていない。

- 雑誌、研究会資料等の書誌情報
- 文献の二次情報
- ページ画像での一次情報

書誌情報や二次情報については従来から学術情報センターでもサービスしており、関係 DBMS などを用いて蓄積することが可能である。電子図書館システムとしての特徴はページ画像を蓄積している点である。ページ画像を蓄積するためには、画像とともに以下の情報を蓄積する必要がある。

- ページ画像の前後関係
- 雑誌とページ画像の間の全体–部分関係
- 文献とページ画像の間の全体–部分関係

そしてさらに、利用者インタフェースでの便宜を考慮すると、文献の前後関係や、二次情報とページ画像の関係など様々な関係を検索できることが必要になる。このような情報は本質的に図 1のようなネットワーク構造を持っており、利用者インタフェースにおいて、雑誌の書誌情報からページ画像へ移動したり、ページの前後関係を辿るなど、航行的な問合せが必要になる。この図には、ページとページ番号との関係や、文献と二次情報との関係などは含まれておらず、全体ではさらに大きなネットワーク構造となる。

2.2　マルチメディアデータの一貫性管理

上のようなネットワーク構造を持ったデータを大量に蓄積する場合、その一貫性管理が大きな問題になる。これは、ハイパーメディア型のデータの自由度が大きいためであり、以下の一貫性管理機構が必要になる。

(1) 参照一貫性管理機構

従来、参照一貫性は、主にアプリケーションプログラムのレベルで維持するという方法が取られてきた。すなわち、アプリケーションプログラムの更新処理の中に参照一貫性を維持するための処理を加えることによって、参照一貫性の維持を行っているのである。このようなアプローチは、データ構造が画一的で参照関係を限定できる場合には可能だが、ハイパーメディアのように参照関係に自由度がある場合には、システムの側で参照一貫性を維持することが必要になる。

(2) リンクの導出機構

ハイパーメディアにおいてリンクを動的に生成する機構の必要性が指摘されている [6]。例えば、あるページ画像からその次の文献へ飛ぶリンクは、文献とページの間の全体–部分関係と、文献間の前後関係とによって導出できる。このような導出関係にあるリンクを個別に蓄積した場合、記憶領域が増加するだけでなく、データベースの無矛盾性を維持する上でも脆弱になってしまう。そのため、

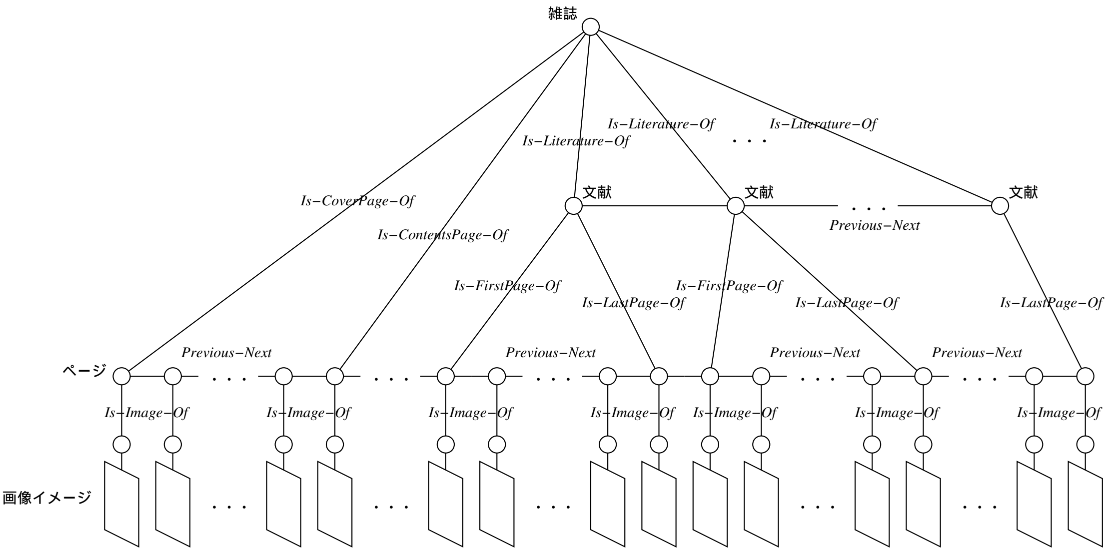


図 1: ページ画像データの上に存在するネットワーク構造

導出可能なリンクはルールで記述しておき、導出機構によって動的にリンクを生成するという形態が必要になる。

(3) 無矛盾性管理機構

無矛盾性管理の意味でも、ハイパーメディア型のデータは脆弱である。抽象データ型のようにカプセル化されたデータの場合、メソッドを誤りなく定義すればカプセル内の無矛盾性は保証される。しかし、ネットワーク構造を持っている場合、データ相互の関係において矛盾が発生するため、無矛盾性の管理は難しい。この問題に対しては二通りのアプローチが考えられる。一方は、抽象データ型のようにネットワーク構造に対する更新手続きを限定する方法、他方は、システム内に無矛盾性を維持するための制約を定義する方法である。しかし、前者の方法だと実行可能なあらゆる更新手続きを予め決めておく必要があり、ハイパーメディアとしての拡張性および自由度の面でデメリットが大きい。

2.3 関連研究

ハイパーテキストやハイパーメディアを蓄積するためのデータベースシステムに関する研究としては TextLink/Gem[6, 9] があり、動的にリンクを生成する機構や ECA ルールによるスキーマ進化機構が提案されている。ハイパーメディアに関数型データモデル AIS を適用する研究も行われており、映像データのモデル化手法や関数型の操作系が提案されている [8]。

また、オブジェクト指向データベースシステムを用いた研究としては Jasmine[2]、OMEGA[4]、Obase[7] などがあり、マルチメディアデータのモデル化手法や ECA ルールによる一貫性管理機構などが提案されている。

データベースの一貫性管理機構に対して形式的な枠組を与えようとする研究としては、演繹データベース [10] がある。

3 記憶システム HyNeSS

3.1 システムの概要

HyNeSS (Hyper Network Storage System) は、ハイパーメディアのようなネットワーク構造を持つデータを蓄積するための記憶システムであり、次の二つの機能の実現を目的としている。

1. データの多様性を吸収するためのネットワーク構造による蓄積

   電子図書館システムなどでは、ネットワーク構造を持つデータを大量に蓄積する必要がある。この時、データ構造が画一的であればリレーションやクラスといった形でデータ構造を定義できる。しかし、多様なデータや関連を蓄積するためには、ネットワーク構造をそのままの形で蓄積する必要がある。

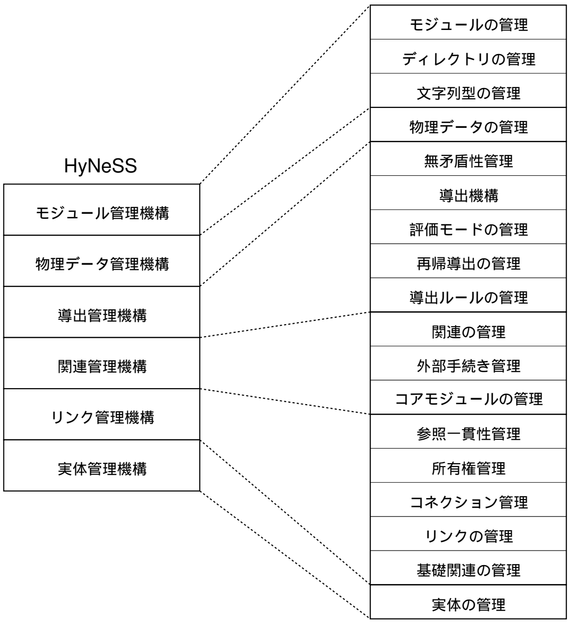


図 2: HyNeSS のシステム構成

2. ネットワーク構造の一貫性管理
   2.2 節で述べた通り、ネットワーク構造を蓄積する場合、一貫性管理機構が必要不可欠であり、参照一貫性管理機構、リンク導出機構、無矛盾性管理機構を実現する。

HyNeSS のアプローチは、データのネットワーク構造をハイパーネットワークの形でモデル化し、その上に述語論理に基づく一貫性管理の枠組を実現するというものである。実現にあたっては一般的な枠組を提供することを目標とし、個々のデータに依存する個別的な特徴を排除している。そのため、HyNeSS が提供している機構は基本的なものであり、実際にデータベースシステムを構築する場合には、HyNeSS が提供する機構の上に、対象とするデータに応じた導出ルール、算術演算オペレータなどを定義することになる。この点が、我々が HyNeSS をデータベース管理システムではなく記憶システムと呼んでいる所以である。

図 2に HyNeSS のシステム構成を示す。システムは大きく六つのレイヤに分かれ、実体管理機構、リンク管理機構、関連管理機構、導出管理機構、物理データ管理機構、モジュール管理機構から成る。下の三つの管理機構はネットワーク構造を蓄積するためのものであり、その上に一貫性管理機構や物理データのための機構が重なる形になっている。

3.2 ネットワーク構造の構成要素

ハイパーメディア型のネットワーク構造を持つデータを蓄積するには、オブジェクト指向データモデルに基づく方法と、意味データモデルに基づく方法とが考えられる。オブジェクト指向データモデルに基づくと、ネットワーク構造のノードをオブジェクトとしてモデル化し、ノード間のリンクは属性による参照として記述することになる。これに対して、意味データモデルに基づくと、ノードを実体として表現し、ノード間のリンクを関連として記述することになる。ただし、オブジェクト指向データベースの中には、オブジェクトの一種として関連を定義できたり、相互参照属性によって双方向リンクを定義できるものもあり [1]、厳密に二分されるわけではなく両者を組み合わせた方法もある。

HyNeSS ではネットワーク構造を意味データモデルの概念に基づいてモデル化する。これは、リンク導出機構や一貫性管理機構を取り扱うときに、リンクを関連によって記述した方が定式化しやすいからである。そこで、HyNeSS では、以下の構成要素でネットワーク構造を取り扱う。

(1) 実体

意味データモデルにおける実体であり、実世界に存在する「もの」を表す。抽象概念も実体であるというのは他の意味データモデルと同様である。

ただし、HyNeSS ではさらに、数値、文字列、テキスト、イメージといった物理的なデータも実体として取り扱う。物理的なデータを抽象データ型によってカプセル化した場合、個々のデータを独立した実体として扱うことができる。すなわち、データ型と内部データの対で物理データを識別し、ある型のある内部データを持った物理データをひとつの実体と見なすのである。これにより、ネットワーク構造におけるノードを一律に実体として扱うことが可能になる。

それぞれの実体には識別子を割り当て、一意に識別出来るようにする。

(2) リンク

ハイパーメディアと同様に、実体間にリンクを張ることによって意味的なつながりがあることを示す。実体間のつながりは、三項関係になること

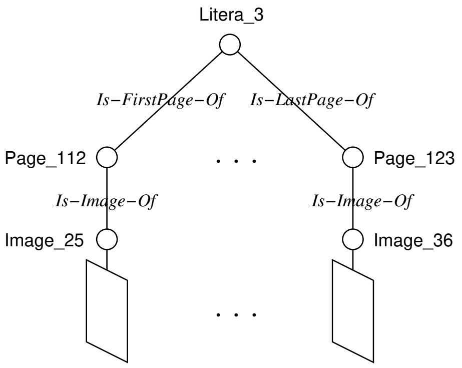


図 3: 実体、リンク、関連によるネットワーク構造

もあり、一般には多項関係になるのでハイパーリンクになる。

(3) 関連

単にリンクを張るだけでは、それがどのような意味を持っているのか識別できないので、それぞれのリンクに意味を識別するための述語を付与する。例えば、文献とその先頭ページとを結ぶリンクには Is-FirstPage-Of という述語を与え、各ページと画像イメージとの間には Is-Image-Of という述語を与える (図 3)。すると図 3中のリンクは、述語と実体の識別子とを用いて下のように表現できる。

```
Is-FirstPage-Of [ Litera = Litera_3, Page = Page_112 ]
 Is-LastPage-Of [ Litera = Litera_3, Page = Page_123 ]
    Is-Image-Of [ Page = Page_112, Image = Image_25 ]
    Is-Image-Of [ Page = Page_123, Image = Image_36 ]
```

括弧でくくられているのは項の並びであり、それぞれの項の役割を示すために項にはラベルがついている。項の出現順序を限定すれば、下のようにラベルを省略しても構わない。

```
Is-FirstPage-Of [ Litera_3, Page_112 ]
 Is-LastPage-Of [ Litera_3, Page_123 ]
    Is-Image-Of [ Page_112, Image_25 ]
    Is-Image-Of [ Page_123, Image_36 ]
```

述語は、意味データモデルにおける関連に相当し、同じ意味をもっているリンクをクラス化する。すなわち、関連は実体間のつながりについての型であり、リンクはそれらのインスタンスということになる。そして関連ごとに、述語、項の数、項のラベルが決まる。

例えば、Is-Image-Of 関連は下のような型を持っている。

```
Is-Image-Of [ Page = $X, Image = $Y ]
```

ここで、$ で始まる文字列 $X, $Y は変数である。

実体間の関連を記述する方法としては関数を用いる方法もあるが [8]、ハイパーリンクを記述するには述語の方が適している。というのは、関数で記述すると、ひとつのハイパーリンクを複数の関数で記述しなければならないからである。例えば、下の述語で表現されるリンクを考える。

```
Is-Image-Of [ Page = Page_112, Image = Image_25 ]
```

このリンクを関数で記述すると下のようになる。

```
Image_25 = Is-Image-Of ( Page_112 )
Page_112 = Is-Page-Of ( Image_25 )
```

このようにひとつのリンクが複数の表現を持ってしまうと、実体から実体への写像関係が強調されてしまい、ネットワーク構造としての性質を記述するのが難しくなる。また、述語を用いた場合、導出や無矛盾性管理の枠組として述語論理を用いることが可能になるので利点が大きい。

ただし、これはあくまでも記憶システムの枠組を取り扱う場合の話である。関数を用いれば、実体から実体へリンクを辿る操作を写像の形でわかり易く表現できるので、利用者インタフェースにおける表現形式としては有効である。

3.3 一貫性管理機構

HyNeSS が管理する一貫性は、参照一貫性、リンクの導出関係、意味的な無矛盾性の三つである。

1. 参照一貫性

   ネットワーク構造としての一貫性であり、参照を受けている実体が削除されないことを保証する。

2. リンクの導出関係

   リンク相互の依存性に関する一貫性であり、あるリンクの存在が他のリンクの存在に依存するとき、すなわちリンク間に導出関係があるとき、導出ルールを定義することによってそれらの一貫性を維持する。

3. 無矛盾性

   意味的な一貫性であり、制約を課すことによって無矛盾性を維持する。

ここではこれらの機構の概略についてだけ述べる。詳細は文献 [3] を参照して頂きたい。

3.3.1 参照一貫性管理機構

参照一貫性管理の問題は、実体およびリンクをどのように削除するかという問題である。参照一

貫性管理の原則は、「他の実体から参照されている実体は削除されてはならない」であり、実体およびリンクを削除する時には以下の処理が必要になる。

1. 実体を削除する時、削除可能かどうか判定する。

   実体の削除が要求されると、実体につながっているリンクの中にその実体を参照しているものがあるかどうか判定する。もしその実体を参照しているリンクがあればそのリンクは削除不可能なので削除要求を棄却する。

2. リンクを削除する時、削除可能な実体が生じるかどうか判定する。

   リンクの削除によって、ネットワークが分割され到達不可能な実体が生じたり、参照がなくなることで削除不可能だった実体が削除可能になることがある。そこで、削除可能な実体が生じるかどうか判定し、生じる場合にはこれらを自動的に削除する。

これらの処理を HyNeSS では、参照関係に基づく非巡回有向グラフを構築することによって実現している。グラフの節点は実体または実体の集合であり、有向枝は参照関係を意味する。節点に実体の集合が来るのは、実体間に相互参照を含む場合があるためである。

3.3.2　リンク導出機構

リンクの導出には二つの側面がある。ひとつは、記憶領域を節約するという面であり、導出可能なリンクをルールという縮退された形で蓄積することによって情報量を減らすことができる。もうひとつは、リンク間に存在する導出関係の一貫性を維持するという側面である。

そこで、HyNeSS では一階述語論理に基づいた一貫性管理機構を導入している。例えば、あるページから次の文献へ飛ぶリンクを、文献・ページ間のリンクと文献間の前後関係とから導出するには下の導出ルールを定義する。

```
Is-NextLiterature-Of [ Page = $X, Litera = $Y ]
    ⟨= Is-Page-Of [ Litera = $Z, Page = $X ],
       Previous-Next [ Previous = $Z, Next = $Y ].
```

ここで、$ で始まる文字列は変数、⟨= は導出記号であり、右辺に合致したリンクが存在するとき左辺のリンクが導出される。導出は層状プログラム [5] に基づいており、否定や再帰導出も定義できる。否定を示すときには述語の前に ~ を付ける。

HyNeSS の一貫性管理機構は、一階述語論理に基づいているという点で、論理プログラミングや演繹データベースと共通しているが、算術演算や文字列演算といった演算処理を組み込む手法が従来のものとは異なっている。すなわち、機能的述語という概念を導入し、算術演算や文字列演算も述語の形で表現する。そしてさらに、そのような演算処理を、ネットワーク構造におけるノードやリンクの動的な生成としてモデル化し、述語の評価アルゴリズムや、その停止性についての定式化を行っている。

例えば、ページの前後関係をページ番号の大小関係で定義する場合、Is-Sum-Of という内部で加減算を行う機能的述語を用いることで下のように定義できる。

```
Previous-Next [ Previous = $X, Next = $Y ]
    ⟨= Is-Page-Of [ Magazine = $Z, Page = $X ],
       Is-Page-Of [ Magazine = $Z, Page = $Y ],
       Is-PageNumber-Of [ Page = $X, Number = $V ],
       Is-PageNumber-Of [ Page = $Y, Number = $W ],
       Is-Sum-Of [ Sum = $W, Value1 = $V, Value2 = 1 ].
```

右辺の最初の二式は $X, $Y が同じ雑誌に属することを指示するものであり、続く二式は $V, $W に $X, $Y のページ番号を割り当てるためのものである。そして最後の式で、$V と $W の間に $W = $V + 1 の関係があることを指示している。

このルールで特徴的なのは、この記述だけで Previous から Next を求めることも、逆に Next から Previous を求めることも可能なことである。すなわち、宣言的な記述になっているのみならず、自然とリンクの双方向性も記述できているのである。これは、導出機構の枠組として述語論理を用いたこと、および、機能的述語の概念を導入したことによる大きな特長である。

3.3.3　無矛盾性管理機構

無矛盾性管理機構は、導出機構の上に構築することができる。すなわち、意味的に矛盾な状態を導出ルールの形で定義しておき、その導出が行われた場合を矛盾した状態と判定するのである。例えば、「全てのページは画像イメージを持っていなければならない」という制約は、導出ルールの形で次のように書くことができる。

```
HasNoImage [ Magazine = $X ]
    ⟨= Is-Page-Of [ Magazine = $X, Page = $Y ],
       ~Is-Image-Of [ Page = $Y, Image = $Z ].
```

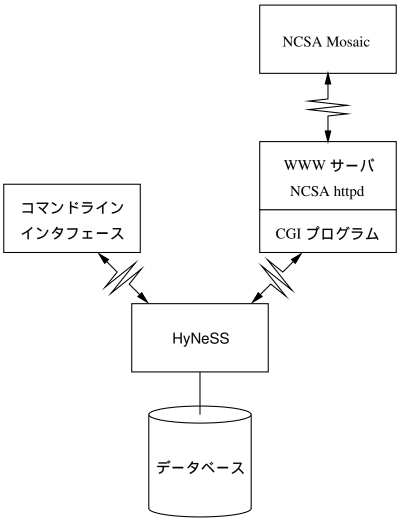


図 4: プロトタイプシステムの構成

そして、HasNoImage を矛盾として定義しておき、この導出が行われた場合に無矛盾性が損なわれたと判定するのである。

3.4 検索処理

検索処理は、航行的問合せと宣言的問合せの双方を実現する。まず、実体から実体へリンクを辿ることによって航行的問合せが可能である。そして、検索条件を導出ルールとする関連を定義することによって、宣言的問合せが可能である。例えば、タイトルに “media” という文字列を含む文献の先頭ページを全て取得するためには、次のような関連を定義すればよい。そして、この関連の導出結果を取得することで検索結果が得られる。

```
Query [ Page = $Z ]
    ⟨= Is-Title-Of [ Litera = $X, Title = $Y ],
       Is-Substring-Of [ Whole = $Y, Part = "media" ],
       Is-FirstPage-Of [ Litera = $X, Page = $Z ].
```

このように HyNeSS における導出機構は、一貫性管理の意味でも検索処理の意味でも大きな役割を果たしているのである。

4 プロトタイプシステムの実装

4.1 システムの構成

前節の設計に基づいてプロトタイプシステムを実装した。プロトタイプシステムの構成を図 4 に示す。プロトタイプシステムでは、ひとつのサーバプログラムに対して、クライアントとして二種類のユーザインタフェースを実装した。一方は、UNIX のシェルに似せたコマンドライン・インタフェースで、HyNeSS の全てのプリミティブを起動できるようになっている。もう一方は、グラフィカルユーザインタフェースによる検索処理を実現するために作成したものであり、NCSA Mosaic をフロントエンドとして利用したものである。

4.2 コマンドライン・インタフェース

コマンドライン・インタフェースからは、更新処理と検索処理のいずれも実行可能であり、HyNeSS の以下の機能を利用できる。

(1) データベースの更新機能
- 実体、リンク、関連の生成・削除
- アトミックトランザクション
- 参照一貫性管理
  - 参照関係の定義
  - 参照されている実体の削除の禁止
  - 参照されなくなった実体の自動削除
- 導出ルールの定義・削除
- 無矛盾性管理機能
  - 無矛盾性制約の定義
  - コミット時の無矛盾性チェック

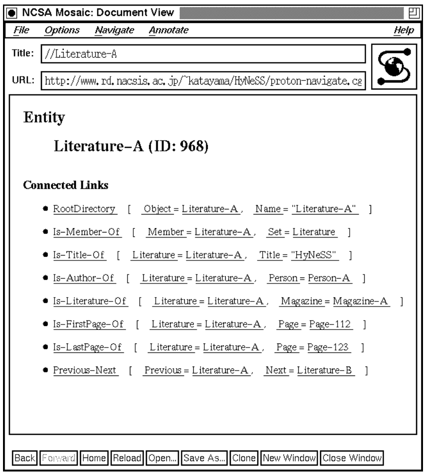
(2) データベースの検索機能
- 実体に接続しているリンクの取得
- 関連に定義されているリンクの取得

4.3 Mosaic を用いたインタフェース

このインタフェースは、述語論理による宣言的問合せとハイパーメディア型の航行的問合せとを実現する。宣言的問合せは、検索条件をテキストフィールドに入力する形態になっており、該当するリンクが一覧表形式で出力される。航行的問合せは、実体から実体にリンクを辿る形態であり、実体ごとに接続しているリンクの一覧が表示される (図 5)。そして、表示されているリンクの述語、ラベル、実体を選択することによって、次の実体に移動できるようになっている。

5 今後の課題

プロトタイプシステムの実装を行ったことで、HyNeSS の実現可能性については確認できたと考えている。そこで今後は、HyNeSS の実用性につ



Title: //Literature-A

URL: http://www.rd.nacsis.ac.jp/~katayama/HyNeSS/proton-navigate.cg

Entity

Literature–A (ID: 968)

Connected Links

- RootDirectory [Object = Literature–A , Name = "Literature–A"]
- Is–Member–Of [Member = Literature–A , Set = Literature]
- Is–Title–Of [Literature = Literature–A , Title = "HyNeSS"]
- Is–Author–Of [Literature = Literature–A , Person = Person–A]
- Is–Literature–Of [Literature = Literature–A , Magazine = Magazine–A]
- Is–FirstPage–Of [Literature = Literature–A , Page = Page–112]
- Is–LastPage–Of [Literature = Literature–A , Page = Page–123]
- Previous–Next [Previous = Literature–A , Next = Literature–B]

図 5: Mosaic インタフェースの画面の例

いて検討を進めていく考えである。具体的には以下のテーマに取り組む計画である。

(1) システムの高速化と実装技術の検討

現在のプロトタイプシステムは、設計の妥当性を検証することのみを目的としており、処理効率については考慮していない。そのため実用的な処理速度は持っておらず、実用性を検証するためには高速化が不可欠である。システムの効率化を図るとともに、HyNeSS 特有の実装法の可能性を模索していく考えである。

(2) 実データへの適用と大規模な蓄積に関する検討

ある程度実用的な速度が得られた時点で実データを蓄積し、実験用データベースを構築する計画である。そして、その結果をもとにして、大規模な蓄積を行うために必要となる機能について検討していく考えである。例えば、識別子の管理法、データの分散配置法などについて検討したいと考えている。

(3) 更新処理の実装技術に関する検討

現在の設計では、無矛盾性の維持は可能であり、矛盾が生じる場合には更新処理を棄却できるようになっている。しかし、一貫性管理手法としては、更新処理を限定することによって一貫性を維持する方法もあり、処理効率の上ではこちらの方が有利な場合も考えられる。また、バージョン管理やスキーマ進化についても検討を進める必要があり、これらの更新処理に関する実装技術についても検討していく考えである。

6 あとがき

ハイパーメディアのようなネットワーク構造を持ったデータを対象とし、それらの一貫性管理機構を含んだ記憶システムとして、HyNeSS (Hyper Network Storage System) を提案した。HyNeSS は、ネットワーク構造を持ったデータを蓄積・管理するための一般的な枠組を構築することを目的としており、マルチメディアデータベースや電子図書館システムの記憶システムとして応用できると考えている。

参考文献

[1] Cattell, R., G., G., *“Object Data Management: Object-Oriented and Extended Relational Database Systems,”* Addison-Wesley (1994).

[2] 石川 博, “次世代ハイパーメディアエンジンとしてのオブジェクト指向型知識ベース管理システム,” 信学技報, DE89-30, 9–16.

[3] 片山紀生, “明示的な意味表現を持つデータベースシステムに関する研究,” 東京大学大学院工学系研究科 1994 年度 博士論文 (1995).

[4] 増永良文, “マルチメディアデータベースシステム開発プロジェクト OMEGA,” 信学技報, DE93-57, 17–24.

[5] Minker, J.(Ed.), *Foundations of Deductive Databases and Logic Programming*, Morgan Kaufmann Publishers (1988).

[6] Tanaka, K., Nishikawa, N., Hirayama, S., and Nanda, K., “Query Pairs As Hypertext Links,” *Proc. of the 7th Int. Conf. on Data Engineering*, IEEE (Apr. 1991), 456–463.

[7] 田中克己, 西尾章治郎, 吉川正俊, 下條真司, 上善恒雄, “Obase: 動的継承とアクティブルール機構を有するインスタンス指向オブジェクトデータベースシステム,” 情処研報, 94-DBS-100, 87–96.

[8] 富井尚志, 由井仁, 石川英彦, 有澤博, “関数型データモデルに基づく映像データベースの構成とその実現,” 情処研報, 94-DBS-100, 63–70.

[9] 銭晴, 谷崎正明, 原伸一郎, 田中克己, “ハイパーテキストデータベースシステム TextLink / Gem におけるオブジェクトとスキーマの動的、段階的な構築機能,” 信学技法, DE92-39.

[10] Ramakrishnan, R. and Ullman, J., D., “A Survey of Deductive Database Systems,” *J. of Logic Programming* (May 1995) 125–149.