

ハイパーメディアデータベースにおける参照一貫性管理

大本 英徹

工学部 情報通信工学科
京都産業大学

WWW(World Wide Web) システムが急速に普及しつつあり、現在のインターネットサービスの主力となってきており、この WWW を用いて、大量文書のデータベースとして運用することも各所で開始されている。しかし、WWW を文書データベースもしくはハイパーメディアデータベースとして機能させるには、いくつかの機能が決定的に不足している。本論文では、WWW 型のハイパーメディアデータベースシステムの大規模で組織的な構築時に問題となるハイパーリンクの一貫性管理に注目し、オブジェクト指向データベースにおける一貫性制約の一つである経路存在従属性 Path Existence Dependency の概念を応用して、ハイパーリンク参照経路の一貫性制約表現について述べる。

Integrity Constraints of Object References in Hypermedia Database Systems

Eitetsu Oomoto

Department of Computer & Communication Sciences
Faculty of Engineering
Kyoto Sangyo University

The World Wide Web (WWW) is currently being familiar with us very much, and will become the most significant service on the Internet. Many web servers are running on the world as the large scale document database systems. However, the WWW system does not have enough facilities to be a real *document or hypermedia* "database system." In this paper, we focus on the management of integrity constraints for hyperlinks on WWW type hypermedia database systems. The management of hyperlinks is one of the major problems on the large-scale and systematic construction of hypermedia systems. We describe the notion of the *path existence dependencies* (PEDs), and, using PEDs, propose a method to represent the integrity constraints of object references in hypermedia databases.

1 はじめに

現在、WWW(World Wide Web) システムが急速に普及しつつあり、現在のインターネットにおける各種サービスの主力となってきている。この理由として以下のような理由が挙げられる。

- Mosaic, Netscape Navigator に代表される直観的な操作インターフェイスが充実しており、特に難しい操作方法を覚えなくても容易にシステムにアクセスでき、様々なネットワークサービス (mail や news 等) を統合化することができる。
- 利用者が容易に情報発信が可能である。すなわち、インターネットに接続してサーバーさえ立ち上げられれば、全世界に向かって自己の有する情報をリアルタイムに発信できる。
- 画像や音声、動画といったマルチメディア情報の扱いが容易であり、専用の特殊なソフトウェアを必要としないため、利用者が様々な表現メディアを自由に用いて情報表現が行える。

特に文字情報以外のマルチメディアデータを利用した情報提供が可能であり、ハイパーメディアシステムシステムを容易に構築できる点が、これほどまでに WWW が隆盛している原動力であると考えられる。この WWW を用いて、大量の文書をアクセスできるようなサービスを提供し、ある種の文書データベースとして機能させることも各所で開始されている。単に文書を蓄積するだけでなく、出版プロセスを全て WWW ベースのオンライン化した Web Publishing なる概念まで出現してきている。

一般にハイパーメディアシステムでは、テキストや画像など様々なメディアを用いて、情報を“ノード”として表し、情報間の関連を“リンク”を用いてノード同士を結び付けることで表現する [11]。しかし、WWWではHTML文書 [12]間のリンクを文書に埋め込まれたURLと呼ばれる文字列表記で静的に表現される。しかもリンクの組織的な作成や保守に関する効果的な手段を提供しておらず、人手によって莫大な労力を費やして管理されているのが現状である。本論文では、今後重要となってくるハイパーメディアデータベースシステムの大規模で組織的な構築という問題に着目し、最も重要な問題の一つであるハイパーリンクの一貫性管理に関して述べる。以下、第2節では、現状のWWWの問題点に関して述べる。第3節では、本論文で述べるリンクの一貫性管理の基本的考え方について説明する。基本的なモデル定義は第4節である。第5節では、経路存在従属性の概念とそれを用いたハイパーリンクの一貫性管理について述べる。第6節はまとめである。

2 WWWのデータベースとしての問題点

現在、オンラインマニュアルやQ&A集等をWWW [12]で提供しようとする試みがいくつか為されており、Webパブリッシングという用語まで現れ始めている。今後は、それら大規模文書データベースを大規模Hypertext(Hypermedia)、すなわちWWWサービスを用いて提供してゆくことが、ごく普通になってゆく可能性は高い。確かにWWWブラウザの高い直観性や操作インターフェイスは注目すべきであり、今後も大規模情報システムに於ける同様の操作性は有用であると考えられるが、現状のWWWシステムをそのまま大規模情報ベース構築に適用しようとするれば、いくつかの問題点が直ちに露呈する。例えば、大規模オンラインマニュアル等をWWWで提供するには、当然、何名、何十名もの共著者による協調編集作業が必要不可欠であり、そのような協調編集作業の過程においては、以下に述べる問題点を解決する機能を備えねばならない。

現状のWWWは非常に古典的なHypertextモデルに基づいており、様々な文書やメディアデータをノードとし、それらを単純に結合するものがリンクであって、それらの集合がシステム全体を構成している。各ノードはHTML文書ファイルやJPEG、MPEGファイルとしてサーバに格納され、リンクはURLとしてHTML文書に埋め込まれる。従ってWWWをHypermediaデータベースシステムとして捉えた場合、データベースシステムならば当然有すべき基本機能が未熟であり、次のような問題がある。これらは全てWWWがHTMLファイルを中心とするファイルシステムを、その記憶管理の基盤に置いているためであると考えられる。

1. 文書に適当な属性(日付、名前、分類等)を与え、付加的な情報を記述する手段はなく、それらに基づく条件検索機能等もない。いわゆるサーチエンジンでは単にファイルとして格納されているHTML文書中の文字列に対してキーワードがマッチするかどうかを調べているに過ぎない。
2. 並行制御機能が無く、情報がファイルベースで格納されるため、複数ユーザによる更新等に際して矛盾無くシステムを稼働させるための並行制御機能は無い。
3. リンクや文書の参照に対して適当な認証を設定して、機密管理をするための基本的な枠組みが弱い。
4. WWW中の文書やリンクの参照関係を管理するための一貫性管理機構が無い。

ところで、DBMSにHTTPプロトコルでアクセスすることを可能にするレイヤーを被せることにより、WWWクライアントを積極的にデータベースアクセスの手段として利用するシステムが現れ始めている。例えばオブジェクト指向データベースシステムを基盤とするものとしてはO₂Web、関係データベースシステムを基盤とするものとしてはOracle WebServer等がある。これらのシステムでは、データベースへの検索結果をHTML形式に変換して、各クライアントに転送することで、リンクやボタンをクリックするだけという単純な操作性、マルチメディアデータ表示、およびWWWブラウザであれば何でも利用できるという高い機種独立性などといった特徴を実現している。しかし、これらは全て、従来、各クライアント機種毎に開発していたデータベースアクセスソフトウェアを、HTTPという制限された(Hypermedia表現言語)形式を介することで、WWWブラウザをグラフィックユーザインターフェイスとして利用し、またその機種独立性を利用してクライアント開発コストを削減させるための手段としてWWW環境を利用しているに過ぎず、上記の問題点を全て解決するものではない。

3 基本的アイデア

WWWの様々なサイトにアクセスしていると、次のような状況に出会うことがしばしばある。1) HTML文書中に埋め込まれたURLで指される文書が既に削除若しくは移動された、又は、URL自体の誤りによって、“Not found.”なるメッセージが表示される。2) リンク先の文書が、別のURLで指される場所に移動されたため、“ここにあったページは都合により移動しました。”なるメッセージとURLだけが記述された文書が表示される。3) URL間違いにより、無関係な文書へリンクされてしまっている。これらは全てWWWにおける文書間リンクがURLとして文書に埋め込まれていることが原因であり、原則的には人手による更新しかない。大規模Hypermediaシステムの組織的な構築において、その文書の品質や編集作業効率を向上させるには、これらの状況を積極的に排除するための編集支援機構が必要不可欠である。その機構の一つとして、本論文では、大規模ハイパーメディアシステムにおけるリンクの参照一貫性の維持という問題に関して焦点を当てる。

3.1 HyperLink

WWW サイトは日々増大しているが、それらは玉石混交であり、リンクが選びやすく内容が読み取りやすいサイトと自分がどの文書を参照しているかすぐに判らなくなるサイトの二つがあることに気がつく。前者の場合、リンクの参照が暗黙的または明示的な論理構造を有しており、利用者がそれを意識することで次に辿るべきリンクを選択する手掛かりとすることが考えられる。典型的な HyperLink の論理構造としては図 1 に示すようなタイプが存在する [12]。(A) 直線構造：各ページが逐次的に参照されることを前提とするタイプで、利用者に順番にアクセスすべき制約を課するものである。例えば、短編小説や機械の操作手順書などがこのタイプに相当する。(B) 階層構造：リンクされた文書が元文書のより詳細な表現となっているようなタイプである。例えば、オンラインヘルプの様に、いくつかの大まかに分類されたトピックがあり、それぞれのトピックをさらに詳細に記述した文書がリンクされているといった構造を有している。(C) 分岐構造：直線的な構造に加えて、利用者に適当な参照経路を選択させるタイプである。例えば、ソフトウェアインストール手順書に於て、ほとんどは同じだが、プログラムファイルを媒体からコピーする部分だけが、利用システム毎に異なるなどといった場合である。(D) 網目構造：特定の参照構造がなく、相互にお互いを参照し合っているタイプである。例えばアドベンチャーゲーム等はこのような構造を有している。この構造のページ群を利用者がアクセスすると、その位置を見失う場合がしばしばある。

また、これらのリンク構造とは無関係に、論文の参考文献のように関連する文書を参照するリンクも適度に存在しているのが通常である。本論文では HyperLink を二つのタイプに分類して捉える。

1. 構造的リンク (Structural Link)：文書の論理的構造を構成するリンクである。例えば、通常の文書における章や節などに対応し、WWW 文書であれば階層構造や直線構造を構成しているリンクである。
2. 意味的参照リンク (Semantic Link)：意味的に関連のある文書を単純に繋ぐリンクであり、一群の文書が構成する論理的構造とは無関係に作成されるリンクである。例えば技術論文の参考文献を指したりするものである。

本論文では、単なるテキストだけでなく動画や音声などひとまとまりのマルチメディアデータを「文書」又は「オブジェクト」と捉える。また文書間の構造的リンクはそれぞれのオブジェクトが有する適当な属性値にリンク先のオブジェクトを直接代入する複合オブジェクト構造で表現するものとする。また、オブジェクト間の巡航操作を考慮する際、属性値としての巡航操作、リンク参照としての巡航操作を厳密に区別はしない。

3.2 リンクの一貫性管理

図 2 に示すようなリンク構造を有するハイパーメディア (ハイパーテキスト) データベースが存在すると仮定する。この例では、あるコンピュータシステムのオンラインマニュアルを仮定しており、基本的に階層構造のリンク参照構造を有している。最上位は *ManualHome* 文書オブジェクトであり、それを起点にして *hardware*, *software* 及び *misc* の 3 本のリンクが出てくる。*hardware* リンクの先は *Index* 文書へと繋がっており、そこから、さらに 3 本のリンク *sec1*, *sec2*, *sec3* が出ており、いわゆる「第 1 節」、「第 2 節」、「第 3 節」の文書オブジェクトを指している。それぞれの *Section* オブジェクトからは次の節へのリンク *next* や詳細情報へのリンク *moreinfo* などが出ている。さらに、*DetailDoc* オブジェクトからは元の *Index* へ戻るためのリンク *backToIndex* が張られている。この構造を図 3 のような構造へと更新しようとするとき、*MotherBoard* 文書オブジェクト以下に位置する *backToIndex* リンクの更新作業が不可避である。この例では、高々 2 本のリンクを修正するだけで済むが、数千数万もの文書オブジェクトが存在し、かつ多人数での同時編集更新がなされるような場合を仮定すると、手作業による更新たちどころに破綻する。この場合、*DetailDoc* オブジェクトへと至る巡航経路の起点となった *Index* 文書へ *backToIndex* リンクの終点が自動的に設定されるような機構があれば有用である。我々は、既にオブジェクト指向データベース [1, 6] における経路存在従属性 (Path Existence Dependency, PED) という一貫性制約について検討してきており [9, 10]、本稿では、その概念をリンクの参照経路一貫性管理に応用する。

4 基本定義

本論文における基本的な枠組みとしては、 O_2 データモデル [2] に制限を加えたモデルを用いている。直観的には本論文におけるオブジェクトとは、ハイパーメディアシステム中のノードに対応する文書や音声や画像などのマルチメディアデータを表現し¹、その構造は組型オブジェクトのみを仮定する。それぞれの属性と属性値の対が文書構造や付加的信息を表現できるものとする。また、属性値には原子値またはオブジェクト識別子のみを用いることが出来る²。

4.1 基本集合

以下において形式的な議論を進めるために、次の互いに素な集合を仮定する。

- 全てのバイト列の集合 S
- 全ての数値の集合 N
- 全ての真偽値の集合 B

¹具体的に、どのような方法で各種メディアを表現するかは、本論文では言及しない。

²議論の簡単化のためである。

- 全てのオブジェクト識別子の集合 I
- 全ての属性名の集合 A
- 全てのクラスの集合 C , 但し, $String, Number, Boolean$ はクラスであり, C の要素であるとする。
- 全てのリンク名の集合 L ,

また, オブジェクト識別子からクラスへの関数 $class$, クラスからオブジェクト識別子のべき集合への関数 $oids$ が予め与えられていると仮定する。これらは形式的には以下のように表される関数である³。

$$class : I \rightarrow C$$

$$oids : C \rightarrow 2^I$$

Definition 1 値 (value)

値は以下のように定義される。

1. 特別な記号 nil は値である。いわゆる空値を表す。
2. 全ての文字列, 数値, 真偽値は値である。これらを原子値と呼ぶ。
3. 任意のオブジェクト識別子 $i \in I$ は値である。
4. v_1, \dots, v_n がそれぞれ nil 又は原子値又はオブジェクト識別子であるとき, n -組 $[A_1 : v_1, \dots, A_n : v_n] (A_i \in A, 1 \leq i \leq n)$ は値である⁴。これを組値と呼ぶ。

Definition 2 型 (type)

型を次のように定義する。

1. クラス C は型である。
2. クラス C_1, \dots, C_n 及び属性名 A_1, \dots, A_n に対し, n -組 $[A_1 : C_1, \dots, A_n : C_n]$ は型である⁵。

ここで, クラスの集合 C から, C 中の各要素を用いて生成可能な全ての型を要素とする集合 $T(C)$ への多対一写像 $type$ が与えられているとする⁶。

$$type : C \rightarrow T(C)$$

さらに, 型から値の集合への写像 Dom を次のように定義する。 C 及び C_1, \dots, C_n をそれぞれクラスとしたとき,

1. $Dom(String) = \{nil\} \cup S$
2. $Dom(Number) = \{nil\} \cup \mathcal{N}$
3. $Dom(Boolean) = \{nil\} \cup \mathcal{B}$
4. $Dom(C) = \{nil\} \cup oids(C)$
5. $t = [A_1 : C_1, \dots, A_n : C_n]$ 但し, $A_i \in A$ かつ $C_i \in C (1 \leq i \leq n)$ の時,

$$\begin{aligned} Dom(t) &= Dom([A_1 : C_1, \dots, A_n : C_n]) \\ &= \{[A_1 : v_1, \dots, A_n : v_n] \mid v_i \in Dom(C_i), 1 \leq i \leq n\} \end{aligned}$$

Definition 3 オブジェクト

オブジェクトとは次のような対 $o = (i, v)$ である。

- i はオブジェクト識別子である。すなわち, $i \in I$ 。
- v は n -組値 $[A_1 : v_1, \dots, A_n : v_n]$ である⁷。

但し, 任意のオブジェクト $o = (i, [A_1 : v_1, \dots, A_n : v_n])$ は, $type(class(i)) = [A_1 : C_1, \dots, A_n : C_n]$, かつ, $v_i \in Dom(C_j) (1 \leq j \leq n)$ を満足するものとする。□

ここで, オブジェクト $o = (i, [A_1 : v_1, \dots, A_n : v_n])$ の有する属性値を, $v_j = i(A_j) (1 \leq j \leq n)$ と表記する。

さらに, データベース中に存在する全てのオブジェクトの集合を DB と表記し, DB 中の任意の相異なる 2 つのオブジェクト $(i_1, v_1), (i_2, v_2)$ において, $i_1 \neq i_2$ であるとする。

³直観的には, $class$ はオブジェクトの識別子から, そのオブジェクトが所属するクラスを返す関数であり, $oids$ はあるクラスに所属可能なオブジェクトの識別子の集合を返す関数である

⁴議論の簡単化のため, O_2 データモデルとは異なり, 入れ子型の値は考慮しない。

⁵ O_2 では型の入れ子を扱っているが, 本論文では簡単のため, 型の入れ子は考慮しない。

⁶直観的には, $type$ とはクラスからそのクラスに所属するオブジェクトの属性構造を返す関数である。

⁷議論の簡単化のため, 組型オブジェクトのみを考慮している。

Definition 4 クラスの外延

クラスの外延とは以下のように定義される関数 $extent(C)$ が返す集合である。 $extent(C)$ は、クラスを引数としてデータベースに存在する全てのオブジェクトの集合 DB の部分集合を返す関数であり、下記のような性質を満たす。

$$extent(C) = \{o \mid o = (i, v), class(i) = C \text{ かつ } o \in DB\}$$

但し、相異なるクラス C_1, C_2 に対し、 $extent(C_1) \cap extent(C_2) = \emptyset$ が成り立つ。

ここで、あるオブジェクト o が $o \in extent(C)$ を満たすとき、 o はクラス C のインスタンスと呼ぶ。

Definition 5 リンク

任意のオブジェクト $o_1 = (i_1, v_1)$, $o_2 = (i_2, v_2)$ 及びリンク名 $lname \in \mathcal{L}$ に対して三つ組 $(lname, o_1, o_2)$ をリンク (*Link*) と呼ぶ。この時、オブジェクト識別子 i_1 と i_2 の関係を $i_1[label] = i_2$ と表記する。また、任意の相異なる2つのリンク $l_1 = (lname_1, o_{11}, o_{21})$ 及び $l_2 = (lname_2, o_{12}, o_{22})$ に対し、 $lname_1 \neq lname_2$ 又は $o_{11} \neq o_{12}$ 又は $o_{21} \neq o_{22}$ のいずれかが成り立つとする。□

ここで、データベース中で定義される全てのリンクの集合を $LINK$ と表記する。

4.2 経路と経路式

Definition 6 巡航列

0 個以上のラベル A_i ($A_i \in \mathcal{A} \cup \mathcal{L}, 1 \leq i \leq n$) をドット記法 [5] を用いて連ねた $A_1.A_2 \dots A_n$ を巡航列と呼ぶ。特に、長さが 0 の巡航列を Id なる記号で表す事とする。□

ここで任意のオブジェクト識別子 i に対して $i(Id) = i$ と定義しておく。

直観的には、巡航列とはオブジェクトからオブジェクトへの参照経路を表す、ドット記法で表された属性又はリンク名の連なりである。例えば、

$$\mathcal{A} \cup \mathcal{L} = \{name, address, contents, reference, author, publisher\}$$

であるとき、 $reference.publisher.address$ は一つの巡航列である。

Definition 7 経路

1. データベースインスタンス DB 中のオブジェクト $o = (i, v)$ 及び $o' = (i', v')$ とラベル A に対して、表記 $i.A$ を巡航経路 (又は単に経路) と呼び、以下のようなラベル付き有向グラフを表すものとする。

- $v = [\dots, A : v', \dots]$ (但し、 $v' \neq nil$) の場合 : $i \xrightarrow{A} v'$
- リンク $(A, o, o') \in LINK$ の場合 : $i \xrightarrow{A} i'$

2. DB 中の任意のオブジェクト $o = (i, v)$ 及びラベル $A_i \in \mathcal{A} \cup \mathcal{L}$ ($1 \leq i \leq n$) に対して、 DB 中にオブジェクト $o_1 = (i_1, v_1), o_2 = (i_2, v_2), \dots, o_{n-1} = (i_{n-1}, v_{n-1}), o_n = (i_n, v_n)$ が存在して、これらの間に経路 $i.A_1.A_2 \dots A_{n-1}.A_n$ が存在するとき、表記 $i.A_1.A_2 \dots A_{n-1}.A_n$ はやはり巡航経路であり⁸、次のようなラベル付き有向グラフを表す。

- $v_{n-1} = [\dots, A_n : v', \dots]$ (但し $v' \neq nil$) である場合 : $i \xrightarrow{A_1} i_1 \xrightarrow{A_2} i_2 \dots i_{n-1} \xrightarrow{A_n} v'$
- リンク (A_n, o_{n-1}, o_n) が存在する場合 : $i \xrightarrow{A_1} i_1 \xrightarrow{A_2} i_2 \dots i_{n-1} \xrightarrow{A_n} i_n$ □

ここで、経路 $i.A_1 \dots A_n$ ($0 \leq n$) に対して、関数 $tail()$ を、経路 $i.A_1 \dots A_n$ を構成する末尾の値を返す関数とする。すなわち、 $i.A_1 \dots A_n = i \xrightarrow{A_1} i_1 \dots \xrightarrow{A_{n-1}} i_{n-1} \xrightarrow{A_n} v$ なる経路が存在する時、 $tail(i.A_1 \dots A_n) = v$ となる。

Definition 8 巡航経路式

任意のクラス C と巡航列 $A_1 \dots A_n$ について、表記 $C.A_1 \dots A_n$ を巡航経路式 (又は単に経路式) と呼ぶ。巡航経路式 $C.A_1 \dots A_n$ に関して、関数 $path(C.A_1 \dots A_n)$ は次のようなデータベース中の巡航経路の集合を表す。

$$path(C.A_1 \dots A_n) = \{i.A_1 \dots A_n \mid o = (i, v) \in DB, class(i) = C\}$$
 □

Definition 9 セレクタ変数

巡航経路式 $C.A_1 \dots A_j \dots A_n$ を構成するクラス C 又はラベル名 $A_j \in \mathcal{A} \cup \mathcal{L}$ ($1 \leq j \leq n$) に変数 X を付加した表記 $C.A_1 \dots A_j[X] \dots A_n$ も経路式の一つであり、特にセレクタ変数付き巡航経路式と呼ぶ。また、 $A_j[X]$ をセレクタ変数付きラベル [4] と呼ぶ。但し $C[X].A_1 \dots = C.Id[X].A_1 \dots$ とする。

⁸通常、ドット記法はデータベース中の属性巡航の結果の末尾の値またはオブジェクトを指し示す記法として用いられるが、本論文では、オブジェクト間の参照経路そのものを表す記法として用いる。

経路式中出现する同名のセレクト変数は、同一の値に束縛されるものとする。例えば、セレクト変数付き経路式 pe 中に 2 つの同一変数が出現するとき、即ち

$$pe = C.A_1 \dots A_j[X] \dots A_k[X] \dots A_n$$

に対し、関数 $path(pe)$ は pe に対応する有向グラフ上のラベル A_j, A_k を有する弧の端点が等しい次のような経路の集合を表す。

$$path(pe) = \{i.A_1 \dots A_j \dots A_k \dots A_n \mid o = (i, v) \in DB, \\ class(i) = C, tail(i.A_1 \dots A_j) = tail(i.A_1 \dots A_j \dots A_k)\} \quad \square$$

ここで、経路 $i.A_1 \dots A_n$ が経路集合 $path(C.A_1 \dots A_n)$ の要素である時、 $i.A_1 \dots A_n$ は $C.A_1 \dots A_n$ に“適合する”と呼ぶ。

5 経路存在従属性とハイパーリンク制約管理への応用

本節では、経路存在従属性の概念と定義、及びそれを用いた HyperLink の参照制約の表現例について述べる。

5.1 経路存在従属性

Definition 10 経路存在従属性

参照経路経路式 pe_1, pe_2 に対して、 $pe_1 \Rightarrow pe_2$ を経路存在従属性 (**Path Existence Dependency, PED**) と呼ぶ。与えられたデータベースに対して、この経路存在従属性が成立する時、以下の条件が成り立つ。

1. pe_1 と pe_2 とに共通のセレクト変数が現れない場合、

$$\forall p(p \in path(pe_1)) \supset \exists q(q \in path(pe_2))$$

2. pe_1 と pe_2 とに共通なセレクト変数 $X_h (1 \leq h \leq n)$ それぞれに対して以下の条件 (*) が成り立つ。

$$pe_1 = C_1.A_1 \dots A_j[X_h] \dots \\ pe_2 = C_2.B_1 \dots B_k[X_h] \dots \\ \text{但し } A_l (1 \leq l \leq j-1), B_m (1 \leq m \leq k-1) \text{ はセレクト変数 } X_h \text{ を持たない。}$$

に対して、

$$(*) \quad \forall p \exists q (p \in path(pe_1) \supset q \in path(pe_2) \quad \wedge \quad tail(i_1.A_1 \dots A_j) = tail(i_2.B_1 \dots B_k)) \\ \text{但し } p = i_1.A_1 \dots A_j \dots \\ q = i_2.B_1 \dots B_k \dots$$

□

直観的に言えば、PED が成立するとは、PED の左辺に適合するデータベース中の任意の巡航経路に対して、右辺に適合する巡航経路が少なくとも一つ存在することを意味する。

5.2 経路存在従属性を用いたリンク参照制約表現

以上のように定義される経路存在従属性を用いて、ハイパーメディアデータベースにおけるリンクの巡航経路に関する一貫性制約を表現できる。ここでは第 2 節で用いた例を用いて説明する。

1. *Index* 文書から *sec1, moreInfo* なる名前の参照が出来る時、*DetailDoc* 文書から *backToIndex* リンクが出ていなければならない。

これは、ほとんど意味を成さない非現実的な例であるが、あえて経路存在従属性で表現すると次のようになる。

$$Index.sec1.moreInfo \Rightarrow DetailDoc.backToIndex$$

Index クラスの任意の文書オブジェクト $o = (i, v)$ について *i.sec1.moreInfo* なる巡航経路が存在する (経路を辿って到達できる各々のオブジェクトが末端に至るまでデータベース中に存在する) 時には、必ず少なくとも一つのクラス *DetailDoc* のインスタンス $o' = (i', v')$ が存在し、かつ *i'.backToIndex* なる経路が存在する (o からデータベース中に存在する一つのオブジェクトへ参照が存在する) という事を表している (図 4)。

2. ある *Index* 文書から *sec1*, *moreInfo* なる順の参照が出来るとき、最終的に到達した *DetailDoc* 文書から元の *Index* 文書へ戻るような *backToIndex* リンクが出ていなければならない。

先程の例と異なり、この例では特定のオブジェクトに関する制約になっている。このように、特定のオブジェクト（オブジェクト群）に係わる経路存在制約を表現するために用いるのがセレクト変数である。一つの PED 表現中に現れるセレクト変数は同一のオブジェクトに束縛される。

$$Index[X].sec1.moreInfo[Y] \Rightarrow DetailDoc[Y].backToIndex[X]$$

この例では、両辺に出現するセレクト変数 X と Y のそれぞれが、同一の文書オブジェクトを指し示しており、*backToIndex* リンクを辿った先が必ず出発点となった *Index* オブジェクトをとることを表している（図 3 参照）。

6 おわりに

本論文ではオブジェクト指向データベースにおける経路存在従属性 PED を応用した、ハイパーメディアデータベースのリンク参照経路の一貫性管理に関して報告した。PED とは、データベース中にある参照経路が存在するとき、それに対応して、別の参照経路が存在しなければならないという制約を表現するものであり、これは WWW などハイパーメディア型情報システムにおける参照リンクの一貫性制約の一つの大きなクラスとなるものである。

本稿で提案するリンク参照経路制約に関しては、未だ形式的な議論は不十分であり、さらなる拡張や詳細な性質の検討などが残されており、例えば以下のような点が挙げられる。

1. PED に関する推論則の詳細な数学的性質の検討
オブジェクト指向データベースにおけるオブジェクトの参照経路に関しては、PED の有する諸性質に関しては既に検討を開始しており、いくつかの推論則が存在することが判っている。但し、それらを網羅的に調べ上げ、完全性を有する閉じた体系をなすのか、またモデルの適当な修正が必要なのかはこれからの検討課題である。
2. セレクト変数に関する述語の利用による表現能力の向上
今のところ、セレクト変数は単に同一のオブジェクトを指し示すだけに過ぎないが、適当な述語を付加し、その述語で表現される条件を満たすオブジェクトのみに束縛されるような能力拡張も考えられる。但し、実用的には、なるべく単純な表現能力の方が扱いやすいであろう。
3. アンカー式の導入
現在の扱いでは、リンクは単純にリンク名及び始点と終点のオブジェクトの 3 つ組に過ぎないが、これはハイパーメディアのリンク定義方法としては不十分である。通常、ハイパーテキスト（ハイパーメディア）では、文書の一部（例えばキーワード）にアンカーと呼ばれるリンクの端点を定義して、そのアンカーからリンクを定義するのが通常である。そこで (*linkname*, o_1 に関するアンカー式, o_2 に関するアンカー式) といったようなリンク定義方法に拡張することが考えられる。
4. 経路表現への正規表現の導入
全ての参照経路に関して、全てユーザが PED 制約を書き下すというのは、あまり現実的とは言えない。ユーザがなるべく少数の制約を定義するだけで、なるべく多くの制約が表現できるほうが望ましい。そのための一つの手段がシステムが自動的に新しい制約を導出するという推論則であるが、さらに一つの PED が表現できる制約を拡張する手段として、経路を表すラベル中に正規表現を導入 [7, 8] できる可能性がある。例えば、「*Index* 文書から *section* リンクを経由して迎える全ての文書から出る *backToIndex* リンクは元の *Index* 文書へ戻るようになっていなければならない」という制約は

$$Index[X].section.*[Y] \Rightarrow Document[Y].backToIndex[X]$$

と表現できる可能性がある。

また今後の方向として、プロトタイプシステムの実装を予定しており、プラットフォームとして O_2 OODBMS と O_2 Web を用いてシステムの基本設計を開始している。

References

- [1] Atkinson, M., Bancilhon, F., DeWitt, D., Dittrich, K., Maier, D., and Zdonic, S., *The Object-Oriented Database System Manifesto*, Proc. of The first International Conference On Deductive and Object-Oriented Databases (DOOD'89), pp.40-57, Dec. 1989.
- [2] Deux, O. et al., *The Story of O_2* , Trans. on Knowledge and Data Engineering, Vol. 2, No. 1, pp. 91-108 (March 1990).
- [3] Khoshafian, S. N. and Copeland G.P., *Object Identity*, Proc. of OOPSLA'86, pp.406-416, 1986.
- [4] Kifer, M., Kim, W., Sagiv, Y., *Querying Object-Oriented Databases*, Proceedings of the 1992 ACM SIGMOD International Conference on Management of Data, pp393-402, June 1992

- [5] Tsukamoto, M., Nishio, S. and Fujio, M., *Dot: A Term Representation Using Dot Algebra for Knowledge-Bases*, Proc. of the 2nd International Conference on Deductive and Object-Oriented Databases (DOOD'91), Lecture Notes in Computer Science, Vol. 566, pp.391-410, December 1991
- [6] Cattell, R. G. G. Ed., *The Object Database Standard: ODMG-93*, Morgan Kaufman Pub., 1994.
- [7] Yoshikawa, M., *Circulation and Reuse of Database Constructs through Common Dictionaries*, Future Databases '92 (Proc. of the 2nd Far-East Workshop on Future Database Systems), Advanced Database Research and Development Series - Vol. 3, World Scientific, pp. 67-70, April 1992.
- [8] 吉川 他, *ObaseSQL: 拡張経路式と継承演算子を持つオブジェクトベース言語*, Proceedings of Advanced Database System Symposium '93, Vol. 93, No. 9, pp.63-72, Dec. 1993.
- [9] 神谷誠, 大本英徹, 田中克己, *オブジェクト指向データベースにおける経路存在従属性と属性キー制約*, Proceedings of Advanced Database System Symposium '93, Vol. 93, No. 9, pp.73-82, Dec. 1993.
- [10] 大本英徹, 高松利行, 田中克己, *オブジェクトデータベースの経路存在制約とその応用*, 情報処理学会, DBS 研究会, 95-DBS-104, pp. 113-120.
- [11] Tanaka, K. and Qing, Q., *Two-Level Schemata and Generalized Links for Hypertext Database Models*, Future Databases '92 (Proc. of the 2nd Far-East Workshop on Future Database Systems), Advanced Database Research and Development Series - Vol. 3, World Scientific, pp. 231-239, April 1992.
- [12] Laura Lemay 著 武倉広幸他訳, *HTML 入門 - WWW ページの作成と公開 -*, プレンティスホール出版, 1995.

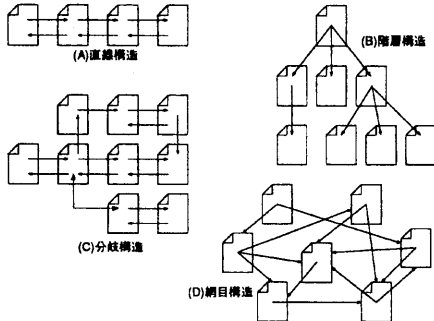


図 1: ハイパーメディアの基本論理構造

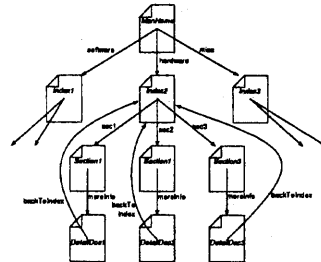


図 2: ハイパーメディアデータベース

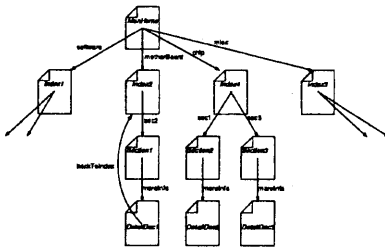


図 3: リンク更新

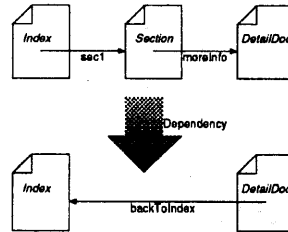


図 4: 経路存在従属性

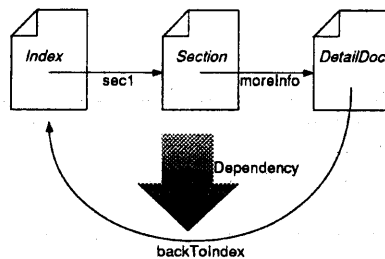


図 5: 実際の経路存在従属性の例