

参照経路の一貫性管理を考慮した ハイパーメディアデータベースモデル

嶋 陽一[†], 大本 英徹[‡]

[†]京都産業大学 大学院 工学研究科 情報通信工学専攻, [‡]京都産業大学 工学部 情報通信工学科

インターネットやイントラネットの普及により, WWW を用いて大量の文書データベースをハイパーメディア構造によって提供する事例が増えてきている。しかし, 大規模になると一貫性を保ちつつこれを人手によって管理することは非常に困難となる。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: Path Existence Dependency) をリンクの一貫性に適用して, ある制約を与えてそれを満たすかどうかの検査を行ない, 文書データベースにおけるリンクの参照一貫性を保つことに関して議論し, リンクの一貫性管理を考慮したハイパーメディアデータベースモデルについて述べる。

キーワード: 経路存在従属性, ハイパーメディア, ハイパーリンク, 一貫性管理

A Management Model for Reference Integrity Constraints in Hypermedia Databases

Youchi Shima[†] and Eitetsu Oomoto[‡]

[†]Division of Information & Communication Sciences, Graduate School of Engineering,
Kyoto Sangyo University

[‡]Department of Computer & Communication Sciences, Faculty of Engineering,
Kyoto Sangyo University

In recent Internet environments, several cases have appeared that the WWW system is used to share many documents involving multimedia data. But it is difficult to maintain the reference integrity among those documents in such WWW based systems. In this paper, we discuss about the management of reference integrity constraints in hypermedia databases as the application of the notion of PED (Path Existence Dependency), and describe a data model for hypermedia databases concerning with the reference integrity constraints.

Keywords: Path Existence Dependency, Hypermedia, Hyperlink, Integrity Constraint

1 はじめに

インターネットやイントラネット上で, WWW を用いて情報発信や情報の共有を行う事例が増えている [6]。理由は, サーバや WWW ブラウザが安価もしくは無料で入手でき, 手軽に利用できるようになっているためである。また, WWW は文書ファイルの閲覧だけでなく, 外部のアプリケーションを起動できるなどの機能を有し, データベースにアクセスするユーザインターフェースとしても利用され始めている。

一般的に, WWW は HTML 形式による文書ファイルにアンカー情報を埋め込み, 設定されている行先へ辿ることができるハイパーメディア構造を有している。このハイパーメディア構造のおかげで, WWW では文書間を順々に辿って情報を得ることができる。しかしながら, WWW におけるハイパーメディアは URL と呼ばれるアドレスによってリンク先を示す簡易的な機構であるので数々の不便な点が存在する。例えば, リンクによって辿った先が存在しないようなときである。これは, 最初からリンクがない場合や当初は存在したが後に削除もしくは移動した場合である。最初からリンク先がない時にはリンクの設定ができないとか, リンク先が削除された時にはそのリンクも削除されるとか, リンク先が移動した場合にはリンク先の情報も変更されるなどの機能があれば一貫性のあるハイパーメディアデータベースの構築ができる。データベース中のある文書を削除するとき, 現在ではその文書にリンクされている他の文書を探す術がほとんどないのが実情である。

最近では, WWW 上で提供される情報も大量になってきており, 参照一貫性を保ったハイパーメディアの構築は人手では困難になってきている。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という一貫性制約を応用したハイパーリンクの一貫性管理を考慮したハイパーメディアデータベースモデルについて論じる。

2 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 [1] 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、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情報を表現できるものとする。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とし、議論の簡単化のために入れ子となるような場合は考慮しない。詳細は、文献 [5] に示すモデルに基づくものである。

2.1 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 $\mathcal{S}$ 、全ての数値の集合 $\mathcal{N}$ 、全ての真偽値の集合 $\mathcal{B}$ 、全てのオブジェクト識別子の集合 $\mathcal{I}$ 、全ての属性名の集合 $\mathcal{A}$ 、全てのクラスの集合 $\mathcal{C}$ (但し、*String*, *Number*, *Boolean* はクラスであり、 $\mathcal{C}$ の要素であるとする)、全てのアンカーの集合 $\mathcal{R}$ 、全てのリンク名の集合 $\mathcal{L}$

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¹。

$$class : \mathcal{I} \rightarrow \mathcal{C}, oids : \mathcal{C} \rightarrow 2^{\mathcal{I}}$$

Definition 1 値

値は空値、原子値 (文字列、数値、真偽値)、オブジェクト識別子及びそれらの n -組の組値からなる。但し、入れ子型の組値は考慮しない。

Definition 2 型

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。

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、 $i \in \mathcal{I}$ 。
- v は n -組値 $[A_1 : v_1, \dots, A_n : v_n]$ である。

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²。□

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。

Definition 4 アンカー

アンカーとは次のような対 $a = (o, r)$ である。

- o は $o = (i, v)$ であるオブジェクトである。
- r はオブジェクト o の領域を示す領域指定子である。但し、 r は o の属するクラス毎に予め定義されているものと仮定する。□

¹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。

²*type* はクラスを引数とし、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。また、*Dom* は型の定義域、もしくはクラスのオブジェクト識別子集合を返す関数である。詳しい定義は文献 [5] を参照のこと。

例えば、テキストオブジェクトに対して、領域指定子は $r_1 = (s_1, s_2)$ とすると、文字列中の s_1 文字目から s_2 文字目までをアンカーであると定義できる。(図 1 参照) また、画像オブジェクトに対しては、領域指定子を $r_2 = ((x_1, y_1), (x_2, y_2))$ とすると、 $(x_1, y_1), (x_2, y_2)$ を結ぶ線分を対角線とする矩形領域をアンカーであると定義できる。(図 2 参照)

領域指定子が 'all' であれば、オブジェクト全体を示すアンカーとする。例えば、 $a = (o, all)$ というアンカーはオブジェクト o 全体の領域上に定義されるアンカーである。

ここで、 $objid()$ という関数を定義しておく。この関数は、値を引数とし、返り値は引数がアンカー $a = (o, r)$ (但し、 $o = (i, v)$) の時はそのアンカーが定義する領域を有するオブジェクトのオブジェクト識別子 i を返し、引数がそれ以外であれば引数をそのまま返す。

Definition 5 クラスの外延

クラスの外延とは、関数 $extent(C)$ が返す集合である。関数 $extent(C)$ 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。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 6 リンク

オブジェクト $o_1 = (i_1, v_1)$, $o_2 = (i_2, v_2)$ が存在するとき、アンカー $a_1 = (o_1, r_1)$, $a_2 = (o_2, r_2)$ 及びリンク名 $lname \in \mathcal{L}$ に対して 3 つ組 $(lname, a_1, a_2)$ をリンクと呼ぶ。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 $l_1 = (lname_1, a_{11}, a_{21})$ 及び $l_2 = (lname_2, a_{12}, a_{22})$ に対し、 $lname_1 \neq lname_2$ 又は $a_{11} \neq a_{12}$ 又は $a_{21} \neq a_{22}$ のいずれかが成り立つとする。□

2.2 経路と経路式

Definition 7 巡航列

0 個以上のラベル $A_i \in \mathcal{L}$ ($1 \leq i \leq n$) ドット記法 [3] を用いて連ねた $A_1.A_2 \dots A_n$ を巡航列³ と呼ぶ。特に、長さが 0 の巡航列を Id なる記号で表すこととする。□

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。

Definition 8 経路

1. データベースインスタンス DB 中のオブジェクト $o = (i, v)$, $o' = (i', v')$ とアンカー $a = (o, r)$, $a' = (o', r')$ 及びラベル 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, a, a') が存在する場合: $i \xrightarrow{A} i'$

2. DB 中の任意のオブジェクト $o = (i, v)$, アンカー $a = (o, r)$ 及びラベル $A_i \in \mathcal{L}$ ($1 \leq i \leq n$) に対して、 DB 中にオブジェクト $o_1 = (i_1, v_1)$, $o_2 = (i_2, v_2)$, $\dots$, $o_n = (i_n, v_n)$ 及びアンカー $a_n = (o_n, r_n)$, 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⁴, 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, a_{n-1}, a_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。

³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し、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、 $\mathcal{L} = \{name, address, contents, reference, author, publisher\}$ であるとき、 $reference.publisher.address$ は一つの巡航列である。

⁴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。

Definition 9 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 10 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 $A_j \in \mathcal{A} \cup \mathcal{L} (1 \leq j \leq n)$ に変数 X を付加した表記 $C.A_1, \dots, A_j[X], \dots, A_n$ も経路式の種類であり、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 [2] と呼ぶ。但し $C[X].A_1, \dots = C.Id[X].A_1, \dots$ とする。

経路式中出现する同名のセレクタ変数は、同一の値に束縛されるものとする。例えば、セレクタ変数付き経路式 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 を有する弧の終点等しい次のような経路の集合を表す。

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

□

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。

3 正規表現を用いた巡航列表現の拡張

2 種類のメタ記号を用いて多くの巡航列表現を簡潔な記述で表すことを可能とする正規表現を導入する。

3.1 正規表現を用いた巡航列表現の拡張

Definition 11 正規表現による拡張巡航列

2 種類のメタ記号について、それぞれ以下のように定義する⁵。

? データベースインスタンス DB 中に存在する任意のリンク名または属性名の 0 回、若しくは 1 回出現

* DB 中に存在する任意のリンク名または属性名の 0 回以上の出現

□

巡航列表現の一部にメタ記号を用いたものも、やはり巡航列を表し、これを拡張巡航列と呼ぶ。また拡張巡航列を用いた経路、経路式をそれぞれ拡張経路、拡張経路式と呼ぶ。

例えば、拡張経路 $i.*reference$ が表現する経路は $i.reference, i.doc.reference, i.doc.ref.reference$ 等である。また、拡張経路式 $C.*reference$ が表現する経路式は $C.reference, C.doc.reference, C.doc.ref.reference$ 等である。

このように、正規表現を導入することにより、複数の経路及び経路式をまとめて簡潔に表現することが可能になる。

4 経路存在従属性とデータベースの一貫性保持

本節では、経路存在従属性の概念と定義、それを用いたリンクの参照制約の表現例、及びハイパーメディアデータベースの一貫性を保持するための機構について述べる。

⁵但し、経路や経路式中で正規表現を用いる時は、グラフが閉路となる場合は考えず、またグラフとして意味のあるものについてのみ考えることにする。

4.1 経路存在従属性

Definition 12 経路存在従属性

経路式 $pe_1, \dots, pe_n, pe$ に対して, $pe_1, \dots, pe_n \Rightarrow pe$ を経路存在従属性 (PED: Path Existence Dependency) 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⁶.

1. pe_i ($1 \leq i \leq n$) と pe について共通のセレクト変数が現れない場合,

$$\forall p_1, \dots, p_n (p_1 \in \text{path}(pe_1), \dots, p_n \in \text{path}(pe_n)) \supset \exists q (q \in \text{path}(pe))$$

2. pe_i ($1 \leq i \leq n$) と pe について共通のセレクト変数 X_h ($1 \leq h \leq m$) が存在する場合, このとき以下の条件 (*) が成り立つ.

$$\begin{aligned} pe_1 &= C_1.A_{11}.\dots \\ &\vdots \\ pe_i &= C_i.A_{i1}.\dots.A_{ij}[X_h].\dots \\ &\vdots \\ pe_n &= C_n.A_{n1}.\dots \\ pe &= C.B_1.\dots.B_k[X_h].\dots \end{aligned}$$

但し A_{il} ($1 \leq l \leq j-1$), B_f ($1 \leq f \leq k-1$) はセレクト変数 X_h を持たない.

に対して,

$$\begin{aligned} (*) \quad \forall p_1, \dots, p_i, \dots, p_n \exists q ((p_1 \in \text{path}(pe_1), \dots, p_i \in \text{path}(pe_i), \dots, p_n \in \text{path}(pe_n)) \supset q \in \text{path}(pe)) \\ \wedge \text{tail}(id_i.A_{i1}.\dots.A_{ij}) = \text{tail}(id.B_1.\dots.B_k)) \\ \text{但し } p_i = id_i.A_{i1}.\dots.A_{ij}.\dots, q = id.B_1.\dots.B_k.\dots \quad \square \end{aligned}$$

4.2 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巡航経路式に関する一貫性制約を表現できる. ここでは, 実用上意味のあるセレクト変数を用いた例のみを取り上げる.

1. *Index* クラスのあるオブジェクトから **ref** というリンクを辿って, *Doc* クラスのあるオブジェクトがあるとき, 辿った先のオブジェクトからもとの *Index* クラスのオブジェクトへのリンク **backToIndex** が存在しなければならない. (図3 参照)

$$\text{Index}[X].\text{ref}[Y] \Rightarrow \text{Doc}[Y].\text{backToIndex}[X]$$

ここで, PED に現れるセレクト変数 X, Y は同一オブジェクトを指し示している.

2. *Index* クラスのあるオブジェクトから **ref1** というリンクを辿って *Doc* クラスのオブジェクト Y があり, かつ *Index* クラスの同一オブジェクトから **ref2** というリンクを辿った先に, とあるオブジェクト Z があるとき, オブジェクト Y からオブジェクト Z へのリンク **ref3** が存在しなければならない. (図4 参照)

$$\text{Index}[X].\text{ref1}[Y], \text{Index}[X].\text{ref2}[Z] \Rightarrow \text{Doc}[Y].\text{ref3}[Z]$$

3. *Index* クラスのあるオブジェクトからデータベース中に存在する 0 回以上のリンクを辿り, 更にリンク **ref** を辿って *Doc* クラスのオブジェクトがあるとき, 辿った先のオブジェクトからもとの *Index* クラスのオブジェクトへのリンク **backToIndex** が存在しなければならない. (図5 参照)

$$\text{Index}[X].*. \text{ref}[Y] \Rightarrow \text{Doc}[Y].\text{backToIndex}[X]$$

⁶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.

4.3 データベースの一貫性の保持について

我々は、これまでに述べてきたモデルに基づいてプロトタイプシステムの構築を考えている。このシステムでは、リンクの参照制約を満足するため、及び文書の移動・削除時にデータベースの一貫性を保持するための方策を検討している。以下に、その方策について述べる。

- リンクの参照制約に関して：我々のシステムにおいて、制約を与えたときやオブジェクトを追加したとき、及びリンクを削除したときに、リンクの参照制約を満たすかどうかの検査を行なう。制約を満たさないときにはユーザに知らせ、一貫性を保つように変更する。
- 文書の移動に関して：WWW では、HTML 文書はいわゆる階層構造によって文書が特定され、階層内の位置を変更することが移動に相当する。このため、一貫性を保つには移動する文書へのリンクの行先を全て変更する必要がある。しかし、我々のシステムでは、オブジェクトはオブジェクト識別子によって特定するため移動という概念がないためにリンク先の変更ということは必要ない。
- 文書の削除に関して：我々のシステムにおいて、オブジェクトを削除する時に、そのオブジェクトを参照するリンクが存在するとユーザに知らせるというような機構を設ける。ここで、リンク先の変更やリンクを削除して一貫性を保つように変更する。

5 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) を応用して参照経路の一貫性管理を考慮したハイパーメディアデータベースモデルについて論じた。ここで述べた参照経路制約ではまだ不十分な点があり、今後の詳細な性質の検討を要する。特に PED に関して、現在はユーザによって指定された参照経路のみしか考慮していない。しかし、PED において推論則が存在することがわかっており、これを使ってユーザが陽に指定した参照経路以外に満たすべき参照経路制約を導出できる可能性がある。また、セレクト変数に対して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。我々の考えたモデルでは、アンカーに対し唯一性を示すアンカー識別子は考えなかったが、このアンカー識別子を取り入れるという方法も考えられる。本論文で示した拡張経路や拡張経路式についての定義はまだ完全ではなく、更なる検討が要求される。

また今後の方向として、プラットフォームに O_2 OODBMS と O_2 Web を用いたプロトタイプシステムの実装を予定している。

References

- [1] Deux, O. et al., *The Story of O_2* , Trans. on Knowledge and Data Engineering, Vol.2, No.1, pp.91-108, March 1990.
- [2] Kifer, M., Kim, W., Sagiv, Y., *Querying Object-Oriented Databases*, Proceedings of the 1992 ACM SIGMOD International Conference on Management of Data, pp.393-402, June 1992.
- [3] 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.
- [4] 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.
- [5] 大本英徹, ハイパーメディアにおける参照一貫性管理, 情報処理学会, DBS 研究会, 96-DBS-108, pp.49-56.
- [6] 日川佳三, 特集 イントラネット, 日経 BP 社, 日経オープンシステム, pp.240-267, 1996.5.
- [7] Frank Halasz, Mayer Schwartz (Edited by Kaj Grønbaek, Randall H. Trigg), *The Dexter hyper text reference model*, Communications of the ACM, Vol.37, No.2, pp.30-39, February 1994.

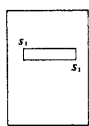


図 3 : テキストオブジェクトのアンカー例

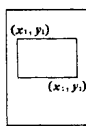


図 4 : 画像オブジェクトのアンカー例

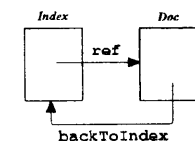


図 5 : リンク参照制約の例 1

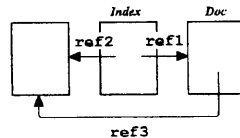


図 6 : リンク参照制約の例 2

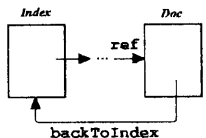


図 7 : リンク参照制約の例 3