

集合演算によるディレクトリツリーのビュー定義

長谷川靖[†] 田辺弘実[†] 岸本義一[†] 武田英昭[†]

我々は、ネットワーク上のコンテンツの増大に対応する流通・管理基盤としてコンテンツ・ディレクトリ管理システムを研究している。このシステムでは、急速な普及が見込まれている LDAP ディレクトリによりマルチメディア・コンテンツのアクセス情報を管理し、これらのディレクトリを汎用的に利用する際の利便性を高めるため、ディレクトリを分割・統合する View 機能の実現を目指している。本稿では、ディレクトリの View 機能を集合演算である関係代数（射影、選択、結合、和）として定義することを述べる。

View definition for directory tree using set operations

YASUSHI HASEGAWA[†], HIROMITSU TANABE[†], GIICHI KISHIMOTO[†], and HIDEAKI TAKEDA[†]

There are the increase of contents in the network, then we work about contents directory management system for information sharing and management base. The LDAP directory manages an access information of multimedia contents, is expected for spread rapidly. In order to improve benefit and convenience using these directories generally, aim at an achievement of a view function divide and integrate these directories. The work described in this paper define the view function of the directory as a set operation of the relational algebra (projection, selection, join, and union).

1. はじめに

近年、X.500 ディレクトリモデル[1]をベースとして、TCP/IP 上での利用のためにプロトコル手順を簡略化した LDAP(Light Weight Directory Access Protocol)[2]が開発された。この LDAP を実装した製品は、Netscape 社、Microsoft 社等の各社から出荷され、急速な普及が見込まれている。ディレクトリとは、通信に関する各種の情報を体系的に保存し、多くの利用者が共通的に利用する目的のためのシステムまたは、サービスであると X.500 では規定している。そのディレクトリサービスの典型例として、電話番号案内サービスがある。しかし、本来、1つの通信アドレスのサービスに留まらず、電子メール、ファクシミリ、郵便、OSI ネットワーク情報、URL、マルチメディア・コン

テンツ(様々な形式の画像、音声、動画ファイル)、ネットワーク機器の管理等の情報を横断的に統合し、地理的にも世界規模での情報の体系化を目指した統合案内サービスを目指している。このような現状の中以下の問題がある。

- (1) ネットワーク上のマルチメディア・コンテンツは、爆発的に増大しているが、その流通・管理基盤が存在しない。
- (2) 現状、インターネットでは、Yahoo!に代表される多くのディレクトリ型の検索サイトにおいて、マルチメディア・コンテンツの URL を管理するディレクトリ型の情報案内サービスが提供されている。しかし、これらの個々のサイトでは、特定の個別システムだけが利用できるローカルなディレクトリによる情報提供サービスしかしていない。つまり、ディレクトリと AP が一体

化した形で構築されているため、個々のサービスで提供しているディレクトリと情報を複数のサイトで横断的に統合して利用することは困難である。

- (3) 本来、ディレクトリとディレクトリに管理される情報は多くの利用者に共通的に利用される目的に、情報を汎用的に体系化して保存しているため、個々のサービス毎に利用しやすいディレクトリは提供できない。

- (4) 複数のサイトの情報を統合して活用するために、元のディレクトリ情報をもつ階層構造が必要であり、その階層構造を保ち、階層構造とその情報を共に統合することが必要とされるが、現状のLDAPでは、階層構造をもつディレクトリから必要な情報をフラットな値の集合として取得することしかできない。

我々の目標は、上記問題を解決する、APと独立にディレクトリ及び、コンテンツを管理する共通基盤である、コンテンツ・ディレクトリ管理システムの実現である。コンテンツ・ディレクトリ管理システムは、コンテンツへのアクセス情報（所在、手段、名前）をLDAPディレクトリで管理し、AP毎にカスタマイズ（複数ディレクトリの統合、参照範囲外のディレクトリ階層の隠蔽）されたディレクトリの階層構造を提案するものである[3][4]。図1にコンテンツ・ディレクトリ管理システムの例を示す。このシステムでは、ディレクトリモデルの外部スキーマを利用者毎のViewとして提供する。リレーショナルデータベースでは、関係代数を用いて、効率的に、利用者毎に使いやすいリレーションを柔軟にViewとして導出する。従来、ディレクトリに対し、簡易なインタフェースから必要なクラスや属性を指定し、LDAPの検索条件を自動生成し、それにより得られるエン트리列をビューとして提供する提案がされている[5]。また、ディレクトリと同様にオブジェクトクラスを階層構造で管理するオブジェクト指向データベースに

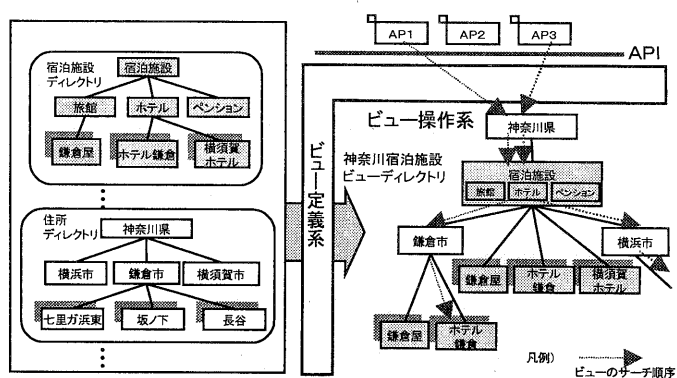


図1. コンテンツ・ディレクトリ管理システムの例

対し、特化・汎化したオブジェクトクラスを新たに追加するというビュー機能の提案がされている[6]。しかし、これらの例では、演算によるViewの導出について言及していない。

本稿では、ディレクトリツリーモデルに対し、関係代数をベースとした集合演算（射影、選択、結合、和）によるViewの導出を提案する。

2. ディレクトリツリーモデルの定義

本節で、ディレクトリツリーモデルを定義する。実世界の事物でその特徴を表すものをオブジェクトクラスと呼ぶ。オブジェクトクラスがもつ1つ1つの情報を属性と呼ぶ。属性は、属性型と属性値からなる。属性型は、ある属性がどのような種類の情報であるかを表し、その属性型の具体的な情報内容が属性値である。属性 $A_i (i=1, \dots, n)$ の集合 A を

$$A = \{ A_1, \dots, A_n \} \quad (i=1, \dots, n) \quad (式 3.1)$$

とすると、属性集合 A はオブジェクトクラス C の要素であるため、

$$C(A) \quad (式 3.2)$$

と表す。また、オブジェクトクラスのインスタンスをエン트리と呼ぶ。このとき、ディレクトリツリーモデルでは、エン트리集合 E をツリー状に配置し、エン트리間の関係をTree構造にすることで、オブジェクトクラス間に実世界の階層関係を反映させた階層関係を構成する。エン트리間の関係において、あるエン트리より上位の階層にあるエン

トリを上位エン트리と呼ぶ。同様に下位の階層にあるエン트리を下位エン트리と呼ぶ。ディレクトリの根であり、親のないエントリをルート (root) と呼び、ディレクトリの末端であり、子のないエントリをリーフエン트리と呼ぶ。エントリは

$$E(RDN, C(A)) \quad (\text{式 3.3})$$

と表す。ここで、RDN は、上位エントリに対し一意に定めるエントリの相対識別名 RDN(Relative Distinguished Name)である。C(A)はエントリの属するオブジェクトクラスである。ディレクトリツリーTは、

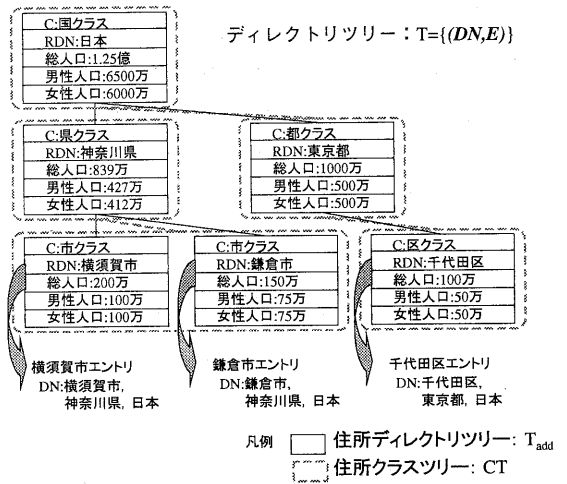
$$T = \{ (DN_1, E_1), \dots, (DN_m, E_m) \} \\ = \{ (DN, E) \} \quad (\text{式 3.3})$$

と表す。

ここで、 $DN_i(i=1, \dots, m)$ はディレクトリツリー上で、各エントリを一意に識別する識別名DN(Distinguished Name)であり、 $E_i(i=1, \dots, m)$ は、エントリである。DN は、そのエントリ及び、そのすべての上位エントリのRDNの列である。ディレクトリツリーモデルのエントリ列をオブジェクトクラス毎にグループ化し、ディレクトリツリーモデルを抽象化したオブジェクトクラスをツリー上に配置し、オブジェクトクラス間の階層関係を構成するクラスツリーCTが考えられ、

$$CT = \{ (CDN_1, C_1), \dots, (CDN_p, C_p) \} \\ = \{ (CDN, C) \} \quad (\text{式 3.4})$$

と表す。このとき、CDN はオブジェクトクラスを一意に識別する識別名である。オブジェクトクラスCに対応するエントリ集合を $E[C]$ とする。この $E[C]$ に含まれるエントリ列をフラットに並べると、オブジェクトクラスCの属性を列、エントリを行とするリレーション R_c と見なせる。また、クラスツリーCTにおいて、ルートのオブジェクトクラスから、リーフのオブジェクトクラスまで辿ったクラス列に対応するエントリ列を並べると、ルートからリーフまでのオブジェクトクラス列の属性列を列、ルートからリーフまでのエントリ列を行と



エントリモデル: $E(RDN, C(A))$ 市クラスのエントリ列: $E[C]$

<table border="1"> <tr><td>C:クラス名</td></tr> <tr><td>RDN:相対識別名</td></tr> <tr><td>属性型1:属性値1</td></tr> <tr><td>属性型n:属性値n</td></tr> </table>	C:クラス名	RDN:相対識別名	属性型1:属性値1	属性型n:属性値n	<table border="1"> <tr><td>C:市クラス</td></tr> <tr><td>RDN:横須賀市</td></tr> <tr><td>総人口:200万</td></tr> <tr><td>男性人口:100万</td></tr> <tr><td>女性人口:100万</td></tr> </table>	C:市クラス	RDN:横須賀市	総人口:200万	男性人口:100万	女性人口:100万	<table border="1"> <tr><td>C:市クラス</td></tr> <tr><td>RDN:鎌倉市</td></tr> <tr><td>総人口:150万</td></tr> <tr><td>男性人口:75万</td></tr> <tr><td>女性人口:75万</td></tr> </table>	C:市クラス	RDN:鎌倉市	総人口:150万	男性人口:75万	女性人口:75万
C:クラス名																
RDN:相対識別名																
属性型1:属性値1																
属性型n:属性値n																
C:市クラス																
RDN:横須賀市																
総人口:200万																
男性人口:100万																
女性人口:100万																
C:市クラス																
RDN:鎌倉市																
総人口:150万																
男性人口:75万																
女性人口:75万																

市クラスに対するエントリのリレーション: R_a

C:市クラス	RDN:横須賀市	総人口:200万	男性人口:100万	女性人口:100万
C:市クラス	RDN:鎌倉市	総人口:150万	男性人口:75万	女性人口:75万

エントリタブルのリレーション: R_e

C: RDN: 総人口: 男性: 女性:	C: RDN: 総人口: 男性: 女性:	C: RDN: 総人口: 男性: 女性:
国 日本 1.25億 6500万 6000万	県 神奈川県 839万 427万 412万	市 横須賀市 200万 100万 100万
C: RDN: 総人口: 男性: 女性:	C: RDN: 総人口: 男性: 女性:	C: RDN: 総人口: 男性: 女性:
国 日本 1.25億 6500万 6000万	県 神奈川県 839万 427万 412万	市 鎌倉市 150万 75万 75万

リレーション R_e に対する識別名のリレーション: DN

日本	神奈川県, 日本	横須賀市, 神奈川県, 日本
日本	神奈川県, 日本	鎌倉市, 神奈川県, 日本

図2. ディレクトリツリーの例

するリレーション R_e と見なせる。このリレーション R_e のタブルをエントリタブル t_e とよぶ。図2にディレクトリツリーの例を示す。

3. 集合演算による View

【定義1】射影

射影とは、ディレクトリツリーTにおいて、そのノードを構成するエントリから必要な属性を切り出し、切り出した属性のみをもつエントリからなるディレクトリツリーを導出する演算である。

ディレクトリツリーTについて、

射影後のディレクトリツリー: T_{add}

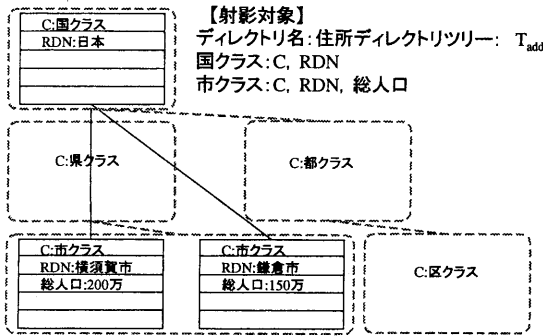


図3. 射影の例

$$T = \{ (DN, E) \} \quad (式 3.1)$$

このエン트리集合をオブジェクトクラス毎にまとめると,

$$E = \{ E[C1], \dots, E[Cn] \} \quad (式 3.2)$$

となる。次に、右辺をルートのオブジェクトクラスからリーフのオブジェクトクラスまで組みとして、まとめると,

$$E = \{ (E[C1], \dots, E[Cm]), \dots, (E[C1], \dots, E[Cp]) \} \quad (式 3.3)$$

となる。このとき、クラスツリーにおいてルートのオブジェクトクラスを $C1$ とする。この右辺は、エン트리タプル t_i のリレーション $R_i (i=1, \dots, q)$ (q はクラスツリーのリーフの個数) からなるため,

$$E = \{ R1, \dots, Rq \} \quad (i=1, \dots, q) \quad (式 3.4)$$

となる。ここで、あるリレーション R の属性集合

$$X = \{ A1, \dots, Ak \} (X \subseteq A) \quad (式 3.5)$$

への射影は,

$$\pi_x(R) = \{ t_i[A1], \dots, t_i[Ak] \mid t_i \in R \} \quad (式 3.6)$$

と表し、これは、 X 上のリレーションとなる。このとき、 $t_i[A1]$ は、エン트리タプル t_i から属性 $A1$ の値を取出したものである。

また、エントリの識別名の集合 DN についても上記のエン트리集合と同様に、オブジェクトクラス毎にまとめると,

$$DN = \{ DN[C1], \dots, DN[Cn] \} \quad (式 3.7)$$

となる。次に、クラスツリーのルートからリーフの組としてまとめると,

$$DN = \{ (DN[C1], \dots, DN[Cm]), \dots, (DN[C1], \dots, DN[Cp]) \} \quad (式 3.8)$$

となる。この右辺は、エン트리タプル t_i のリレーション $Si (i=1, \dots, q)$ (q はクラスツリーのリーフの個数) からなるため,

$$DN = \{ S1, \dots, Sq \} \quad (i=1, \dots, q) \quad (式 3.9)$$

となり、これに対し、(式 3.6) から得られる射影したエン트리列の RDN 集合を元に、射影されないエント리를考慮して、射影されるエントリの DN を導出する。こうして、射影により (式 3.6) から得られるエン트리集合と、導出後の DN からなる Tree 構造のノードとなるエン트리集合を導出する。 □

この射影により必要な属性のみを見せることと同時に、機密性の高い属性は見せないということや、目的のエン트리までの検索操作の回数を減らすことができる。図3に射影の例を示す。

【定義2】選択

選択とは、ディレクトリツリー T において、エン트리集合の属性に関する選択条件を指定し、その選択条件を満たすエン트리タプルからなるディレクトリツリーを導出する演算である。

ディレクトリツリー T のエン트리集合 E をエン트리タプル t_i からなるリレーションの集合と考えると、あるリレーションに対する選択を,

$$\sigma_F(R) = \{ t_i \mid t_i \in R \wedge t_i[X] \theta D \} \quad (式 3.8)$$

と表す。このとき、 X はリレーション R の属性集

選択後のディレクトリツリー: T_{add}

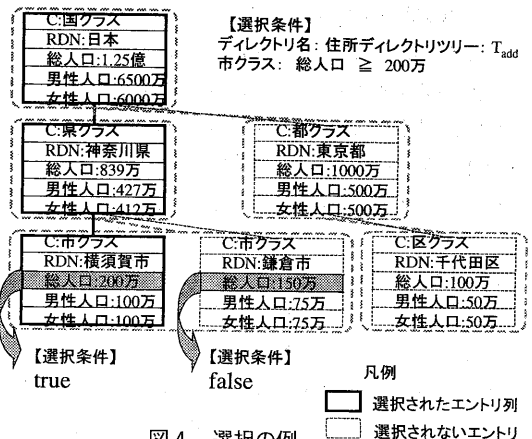


図4. 選択の例

合, θ は比較演算子, D を定数, これらからなる $t_i[X]\theta D$ を選択条件である論理式 F とする. このように, 選択はリレーション R のエントリタプルから, 属性集合 X と定数 D とが比較演算子 θ の関係を満たすエントリタプルだけを取り出して, 新しいリレーションを導出する.

また, 選択ではエントリタプル毎に取出すため, エントリの識別名集合 DN に影響がない. こうして, 選択により(式 3.6)から得られるエントリ集合からなる Tree 構造のノードとなるエントリ集合を導出する. □

この選択により属性に関する条件を指定して, 必要なエントリのみを見せることができる. 選択の例を図 4 に示す.

【定義 3】 結合

結合とは, 2つのディレクトリツリー T と U において, 両者のエントリの属性に関連がある場合, エントリタプル間で連結をとったプロジェクトクラスでグルーピングされるエントリ集合を 1つの単位として, その 2つのオブジェクトクラス間に対し, 結合条件を指定し, 結合条件を満たすエントリ集合の組を特定する. その次に, T と U のエントリタプル間を連結したツリーを構成する. ディレクトリー T, U のオブジェクトクラス C, G のエントリタプルからなるリレーション R, S を考える. この 2つのリレーション R, S の属性集合 A, B との結合

$$R \underset{F}{\omega} S = \sigma_F(R \times S) \quad (\text{式 3.9})$$

は, ω と表す. このとき, $R \times S$ は, リレーション R と S の直積であり, $A_i \theta B_j$ を選択条件の論理式 F として, 属性 A_i を含むエントリタプルに対して選択条件 F を満たす属性 B_j を含むエントリタプル群を取出し, リレーション R, S のエントリタプルを連結する. また, エントリの識別名集合 DN について, 射影ではリレーション R から S への結合を考えるとき, リレーション R の DN は, 影響がないが, リレーション S の DN には, 連結するエントリタ

プルの DN を考慮して, 結合されるエントリの DN を導出する. こうして, 結合により関連するエントリを連結したエントリ集合を導出する. □

この結合の結果, あらかじめ決められ分類のツリーを属性で関連付けて新しく分類し直したツリー

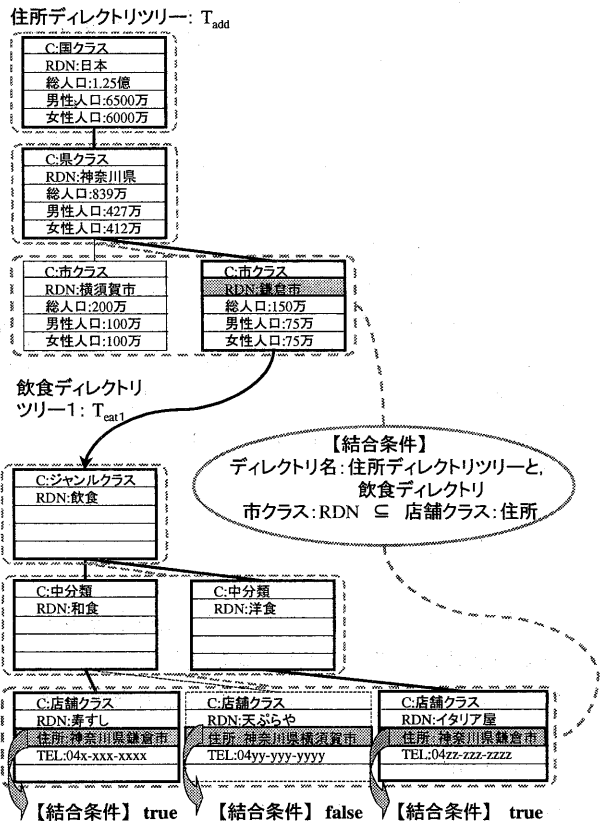


図 5. 結合の例

を見せることができる. 図 5 に結合の例を示す.

【定義 4】 和

和とは, 2つのディレクトリツリー T, U において, 両者のエントリタプルを要素とする集合の和集合をとり, 新たなディレクトリツリーを導出する演算である. 和集合では, クラスやその属性が等しいエントリは存在しない. ここで, 2つのディレクトリツリー T, U のエントリタプルをタプル

とするリレーション R, S が和をとれる制約条件として以下の2つを指定する。

(1) 2つのリレーションで次数が等しい。

(2) リレーションを構成するオブジェクトクラス, 属性が等しい。

ここで, 次数とは, リレーションがもつ属性の数をいう。

リレーション R, S のエンリタプル集合の和を

$$R \cup S = \{t_i | t_i \in R \vee t_i \in S\} \quad (\text{式 3.10})$$

と表す。このとき, t_i をリレーション R, S のエンリタプルとする。 □

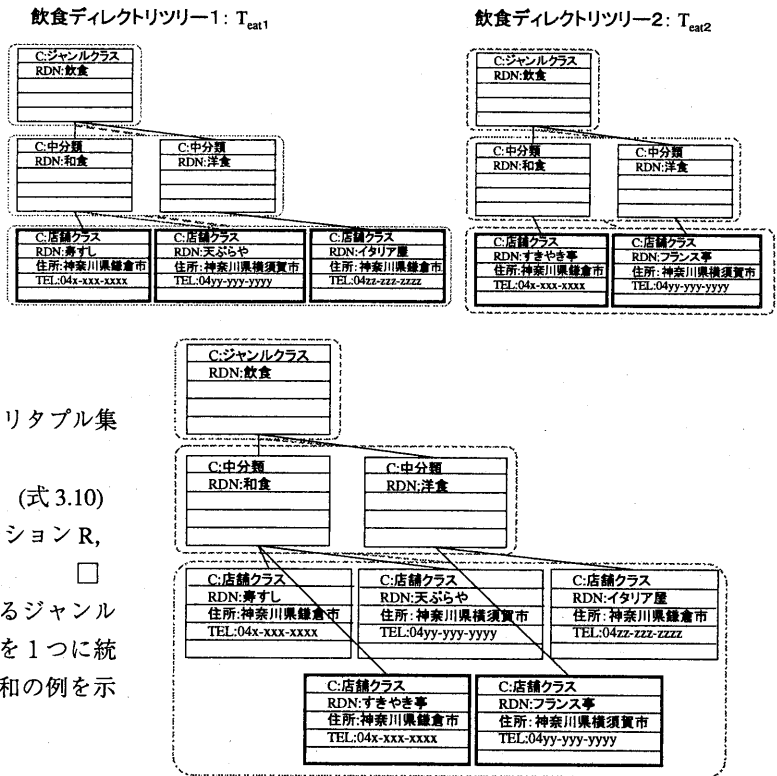
この和の結果, 分野の異なるジャンルを分類したディレクトリ同士を1つに統合することができる。図6に和の例を示す。

4. おわりに

本稿では, ディレクトリツリーの View を導出する集合演算として, 以下の4つを定義した。

- (1) 1つのディレクトリツリーから必要な属性をディレクトリの階層方向でエントリを切り出す射影
- (2) 1つのディレクトリツリーから属性に関する選択条件を満たすエントリをディレクトリから縦方向で切り出す選択
- (3) 2つのディレクトリツリーの属性に関する結合条件を満たすディレクトリ間を連結する結合
- (4) 2つのディレクトリツリーにおいて共通のエントリを統合する和

今後は, View 機能の実装と具体的な事例による評価を実施する。



参考文献

- [1]ISO/IEC 9594-1~9 ITU-T X.500 (1995)
- [2]Tim Howes, Ph.D Mark Smith: LDAP Programming Directory-Enabled Applications with Lightweight Directory Access Protocol, (1997)
- [3]加藤他:コンテンツ・ディレクトリ管理システムのシステム構成法 (1998, 第57回情処全国大会)
- [4]長谷川他:コンテンツ・ディレクトリ管理システムのView機能 (1998, 第57回情処全国大会)
- [5]Rui J.P.Jose, others: Providing external views on directory user interfaces (1996, Computer Network ISDN System, vol.28, p543-550)
- [6]福本他:オブジェクト指向データベースにおけるビュー機能の試作(1994, 第48回情処全国大会)