

## ディレクトリ情報ツリー構成法の検討

藤原 進 福村 好美

NTT情報通信網研究所

番号案内を対象としたディレクトリサービスを提供するためのディレクトリ情報ツリー(DIT)構成法について、以下を報告している。(1)公衆DITにおける新たなオブジェクトクラスやテナント、チェーン店等の多様な階層構造の表現法の提案、(2)エントリの属性情報をRDBMSで格納管理することを前提としたDIT情報管理方式に関して、保持すべき情報の分類、及びそれらを分離、統合、あるいは多重管理する方式の比較、及び(3)エリア先も検索範囲とする場合の効率的なエントリ検索処理方式に関して、評価モデルを用いた3方式案のスループット特性評価、の3点である。

### A STUDY ON DIRECTORY INFORMATION TREE MANAGEMENT METHOD

Susumu Fujiwara Yoshimi Fukumura

NTT Network Information Systems Laboratories

This paper describes a study on the configuration and management method of directory information tree (DIT) aimed for telephone number guide service. New object classes and representation of various hierarchical structure such as tenant shop and chain store are proposed. Then we compared alternative methods for managing DIT information supposing that the attribute information of entries are stored in RDBMS. Throughput characteristics of three entry search methods are also evaluated using an evaluation model in the case where the aliased areas are also to be searched.

### 1. まえがき

電話、FAX、パソコン通信等電子通信の普及・発展に伴い、通信に関する各種情報を体系的に管理、運用できるシステムの必要性が高まっている。[1] ディレクトリシステムは、通信に関わる情報を共通の構成法、アクセス法で提供することを目的としたシステムであり、そのサービスとプロトコルの国際標準が、X.500シリーズとして1988年にCCITTにより勧告され[2][8]、国内外で研究開発が進められている。[3][4][5] 今後の通信の国際化、通信網の多様化・複雑化に伴い、X.500ディレクトリシステムの適用が期待されるサービスの範囲は拡大しており、当研究所でも、電子メール、電話番号案内などの分野で、その実装技術や運用技術の研究を進めている。[6]

本稿では、(1)番号案内サービスを例にして、公衆ディレクトリサービスを提供する際に有効なオブジェクトクラスや階層構造の表現法、(2)リレーショナルデータベース管理システム(RDBMS)の適用を前提としたDIT情報管理方式の比較、および(3)エリアス先も含めた検索要求を効率的に処理するためのエントリ検索方式に関する特性評価の3点について報告する。

### 2. ディレクトリ情報ツリーの概要

X.500で規定されるディレクトリ情報ベース(DIB)は図1に示すように階層構造を持つエントリの集合から構成され、各エントリは複数の値を保持可能な属性の集合からなる。エントリはDIB内で一意な識別名(DN: Distinguished Name)で識別される。この識別名は親の識別名に自分の相対識別名(RDN: Relative Distinguished Name)を付け加えるという階層構造を持つ。

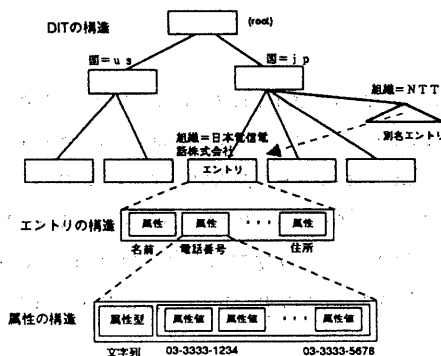


図1 ディレクトリの情報格納モデル

この名前の階層構造をツリーで表現したものをディレクトリ情報ツリー(DIT)と呼ぶ。あるエントリは識別名に加え、別の名前(例ではNTT)で参照できる別名機能がある。

### 3. 公衆DITの構成モデル

X.500ディレクトリの国際勧告[2]では、ディレクトリ情報の格納構造について、国名までは規定しているが、それ以下の構造は各国で規定することとなっている。日本においては、TTC、INTAP等の標準機関による実装規約等の規定があるが通信情報の案内の観点からみるとディレクトリ利用者から見える論理的構造を表わすDITの構成については、十分であるとはいえない。特に、公衆で利用するディレクトリサービスでは、一般の人が利用して分かりやすいDIT構成を実現する必要があり、この点に關した具体的な研究はなされていない。

本章では、番号案内サービスを検討対象として、公衆ディレクトリサービスを提供する際の利用可能なオブジェクトクラスや階層構造の表現等のDIT構成モデルについて述べる。

#### 3.1 オブジェクトクラス

データを維持管理する側ではなく、ディレクトリを利用するユーザ(番号情報検索者)の視点から番号案内における検索要求について分析した結果、以下に示すオブジェクトクラスの導入が有効であるといえる。

##### (1) 管轄企業クラス

電気、ガス、電話等の住所が決まれば一つの窓口や支店が決まるような管轄を持っている企業あるいは企業グループを表現するクラス。

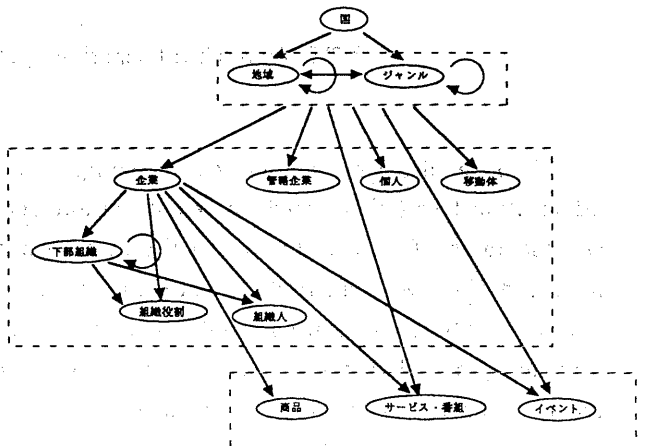


図2 公衆DITで利用されるクラス間の関係

(2) サービス・番組クラス

テレホンサービス、チケットの予約センター、各種相談窓口等、組織階層上は特定の組織単位や部署の配下に管理されるべき情報であるが、ユーザからみると、最も上位あるいは前面に位置してほしい情報を格納するためのクラス。

(3) イベントクラス

一時的に開催されるイベントや選挙事務所等の情報を格納するためのクラス。ユーザからの検索ニーズは高い。一般に開催者やスポンサ企業名の認知度はそれほど高くないため、企業の配下のエントリとして格納するよりも専用のクラスを設ける方が検索しやすい。

(4) 商品クラス

ユーザは会社名だけでなく、広告等で知った商品名等をキーにして検索したいというニーズが多い。それらの商品に関する情報を格納するためのクラスが有効である。

個人クラス (Residential Person) や企業クラス (Organization) 等の X. 500 標準クラスと上記クラスを適用した場合のクラス間の関係の例を図2に示す。

3.2 組織の階層構造

ハローページ電話帳において、組織の階層構造はインデント (文字の字下げ) として表現されている。しかし、上下の情報間の関係を分析すると以下のようなパターンが存在することが分かる。

(1) 支店型

地域対応に支店や営業所を持つような組織の階層構造に対応している形態。

(2) テナント型

建物内にその企業の組織のほかにも、テナントとして別の企業が入っている形態。テナント店は、自社の支店をいくつか持ち、(1)の支店型の階層構造を持っている場合もある。

(3) フランチャイズ型

個人商店ではあるが、フランチャイズやチェーンに加盟しており、その中の1メンバーとしても階層上組み込まれている。

以上の検討により実現されるDITの構成モデルを図3に示す。階層構造のパターン(2)(3)におけるテナントや加盟のようなリンク関係は、X.500のエリアス機能を利用して実現することができる。

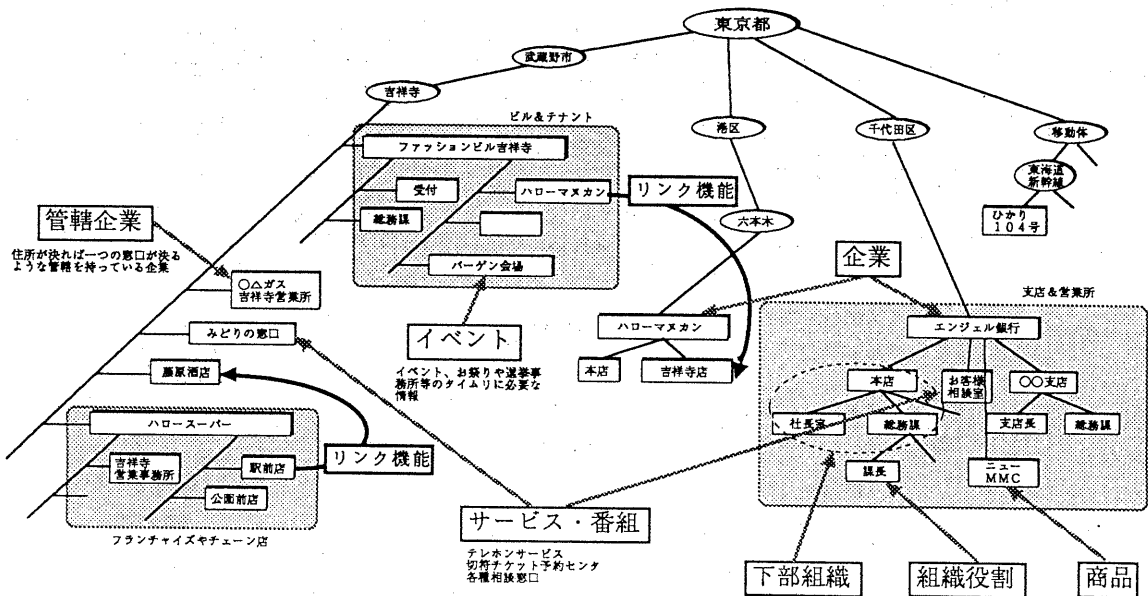


図3 公衆DIT構成モデル

#### 4. DIT情報管理方式

##### 4.1 前提

DIBに対するアクセスは、(a)識別名からDIT上のエントリを特定する処理と(b)該当エントリの属性情報の参照/更新の2つに大別できる。2つの処理に対応して、DIBを実現するために保持すべき情報は、エントリ間の上下関係及び識別名に関するトリー情報と各エントリの属性情報がある。本検討では、後者の属性情報の管理にRDBMSを適用して、データ更新に伴うトランザクション管理やデータベース障害処理等の属性情報管理に必要な機能を実現するアプローチをとっている。

##### 4.2 方式案

トリー情報の保持方法の観点から、以下の3案について考察する。

###### 案1 分離方式

トリー情報を管理するDIT管理部と属性情報を管理する属性管理部に分離する方式。DIT管理部と属性管理部間の情報の対応は、各エントリにシステム内でユニークなID(エントリID)を持たせて対応づける。

###### 案2 統合方式

トリー情報と属性情報をRDBMSで一括管理する方式。あるエントリに対応するレコード内にトリー情報と属性情報を同時に保持する。なお、トリー情報と属性情報をテーブル分割して管理する方式が[3]で提案されている。

###### 案3 多重方式

案1と同様に、DIT管理部と属性管理部とを分離し、さらに属性管理部にもトリー情報を多重に保持する方式。案1同様、エントリIDを導入する。

表1に方式概要およびディレクトリに対する操作からみた比較を示す。案1および案3の特徴は、トリー情報を分離することにより、Listオペレーション(子エントリのRDN一覧要求)等のDITに関連する処理についてはRDBMSではなく専用のモジュールで高速処理可能となる点である。特に、案1は、名前の階層構造の変更が属性管理部に影響を与えずに実行できるという柔軟性を持つ。

エントリ検索は、指定したエントリ(ベースDN)をルートとするサブトリーを検索対象範囲と

表1 DIT情報管理方式の比較

方式概要	案1 分離方式		案2 統合方式		案3 多重方式	
	DIT管理部	属性管理部	DIT&属性管理		DIT管理部	属性管理部
子エントリ名一覧	DN→子RDN一覧		DN→子RDN一覧		DN→子RDN一覧	
エントリ読出し	DN→EID	EID→属性情報	DN→属性情報		DN→EID	EID→属性情報
エントリ検索	ベースDN→EID ベースEID→検索範囲のEID集合	属性条件→該当エントリのEID集合 両者の交わり集合	ベースDN+属性条件→該当エントリの属性情報		ベースDN→ルートからベースまでのEIDリスト	EIDリスト+属性条件→該当エントリの属性情報
エントリ追加/削除	エントリ追加/削除	エントリ追加/削除	エントリ追加/削除		エントリ追加/削除	エントリ追加/削除
エントリ変更		属性変更	属性変更			属性変更
RDN変更	RDN変更	RDNの属性値変更	配下の全エントリのRDN変更	×	RDN変更	RDNの属性値変更
サブトリー移動	親子リンク付替え		配下の全エントリの名前変更	×	親子リンク付替え	配下の全エントリのEID変更
情報格納効率	情報の二重管理がない		RDN情報が冗長	×	EID情報が冗長	

する条件（サブトリー条件）と属性値に対する一致や大小比較等の条件（属性値条件）の2つから構成される。

案1では、サブトリー条件をDIT管理部が、属性値条件は属性管理部がそれぞれ判定し、両者の交わりをとるというオーバーヘッドがある。案2は、情報を全て同一テーブルに保持することにより、検索については最も有利な方式である。案3は、ベースDNをDIT管理部でルートからベースエントリまでのエントリID列に変換し、属性管理部でエントリIDと属性値条件から検索を実行する方式である。

ここで、X.500プロトコルでは、エントリ検索要求におけるオプションとしてエリアス先も検索範囲に含むかどうかを選択できる（パラメータ名：SearchAlias）。エリアス先も検索範囲に含む場合の処理方式については次章で述べる。

RDN変更等のDIT構造の更新の点からみると、案1が最も柔軟性に優れている。案2はRDN情報に関して冗長となるため、DIT構造の更新に対しては処理量が増加する。案3は、属性管理部で保持するトリー情報をRDNではなくエントリIDとして仮想化することにより、RDN変更に対する柔軟性を確保している。

各案の適用領域としては、案1はDIT構造の更新頻度が比較的高い場合に有効であり、情報の格納効率の面でも優れている。その際、エントリ検索をいかに高速化するかが課題となる。案2は逆に、DIT構造の更新頻度が少なく、検索性能

を優先する場合に適している。案3は、案1、2の中間解と位置付けられる。なお、案2、3については、トリー情報をテーブル上に保持するため、DITの深さに制限を設ける等の措置が必要となる。

## 5. エントリ検索方式

前章では、DIT情報の管理方式について定性的な比較を行った。本章では、サービスを提供する上で重要かつ頻繁に利用されると予想されるエントリ検索に関して、エリアス先も検索範囲に含む場合に注目して、具体的な処理方式の代替案の比較評価を示す。以下では前章の案1分離方式を前提に議論を進め、最後に他の情報管理方式への適用性について述べる。

### 5.1 方式案

エントリ検索要求で、エリアス先も検索範囲に含むかどうかを選択するSearchAliasパラメータを真に指定した場合には、図4でハッチングで示した4つのサブトリー領域が検索範囲となる。このような場合には、サブトリー条件の判定は指定されたベースエントリだけでなく、エリアスでポイントされたエントリもベースエントリと同様に扱う必要がある。本節では、エリアス先の検索も含めた場合のエントリ検索方式として、以下の3方式を検討する。

3方式とも、属性値条件を満たすエントリの候補については、属性管理部の機能、すなわちRD

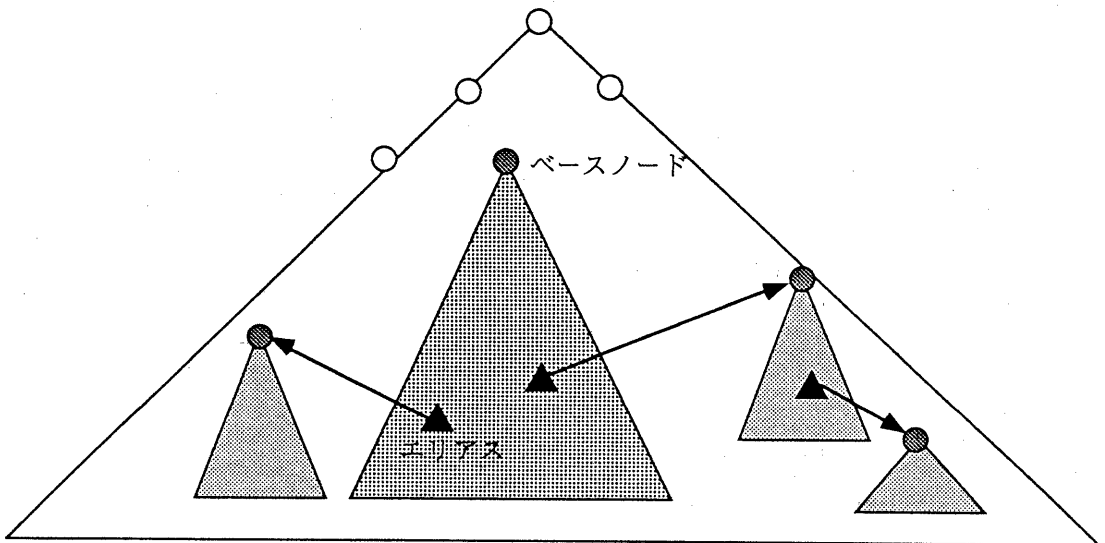


図4 サブトリー検索範囲

BMSの検索機能を利用して得られていることを前提としている。

#### 案A 数え上げ方式

ベースエン트리からDITを辿り、エリアス先も含めて検索範囲内の全エン트리ID（以下、EID）を数え上げ、属性値条件を満たす候補エントリーのEID集合との交わり集合を求める。

#### 案B 遡り方式

図4における4つの領域のベースEIDの集合をまず求める。次に、属性値条件を満たす各候補エン트리について次の手順を実行する。DIT上で順次、親エントリーのEIDを求め、ベースEID集合に含まれるかどうかのチェックをルートまで遡って実行し、サブツリー条件を判定する。

ベースEID集合を求める方式としては、各エントリーに自分の配下のサブツリー内に属するエリアスのEID集合を管理情報として保持する方式を採用する。

#### 案C エリアス逆ポインタ方式

エリアスで指されている実エン트리からエリアスへの逆ポインタを管理情報として保持する。属性値条件を満たす各候補エン트리について次の手順を実行する。DIT上で順次、親エントリーのEIDを求め、ベースEIDとの比較をルートまで遡って実行する。その際、親エントリーの管理情報も同時に読み出し、エリアス逆ポインタが存在した場合には、エリアス元でも同様の遡り手順を実行する。

案A数え上げ方式は、処理論理が単純で、新たな管理情報を必要としないという特徴があるが、反面、検索範囲が広大になった場合に、数え上げる処理が膨大になり、性能上不利である。これに対し、案B遡り方式、案Cエリアス逆ポインタ方式の基本的考え方は、属性値条件を満たす各エントリーに対し、DIT上を遡り、対象となるサブツリー領域のサブノードを通過するかどうかチェックしてサブツリー条件を判定するというアプローチである。従って、属性値条件を満たす候補が比較的少ない場合には、検索範囲の広さにあまり依存せず、検索性能上有利となる。

次節では、これらの方式の特性や適用領域を定量的に評価するためのモデルを提示する。

## 5. 2 トリーモデル

処理対象となるDITの特性をモデル化するため以下のパラメータを導入する。

D: DITトリーの深さ

l: あるエントリーを親に持つ子エントリーの数

p: 自分の兄弟エントリーにエリアスが発生する確率

d: 指定エントリーをベースとするサブツリーの深さ

上記パラメータを利用してトリー内のエントリー数等に関する以下の関数群を定義する。

$$\text{Tree}(l, d) = (l^d \cdot d - 1) \cdot (l - 1)$$

指定サブツリー内のエントリー数。但し、エリアスは含まない

$$\text{Alias}(l, d, p) = (\text{Tree}(l, d) - 1) \cdot p$$

指定サブツリー内に含まれるエリアスの数

$$\text{Entry}(l, d, p)$$

$$= \text{Tree}(l, d) + \text{Alias}(l, d, p) \cdot \text{Tree}(l, d)$$

エリアス先も含めた検索範囲内のエントリー数

$$\text{Back}(D, d, p) = (D - d) \cdot (1 + (D - d) \cdot p)$$

逆ポインタ方式において遡る親エントリーおよびエリアスの数

なお、関数Entry()、およびBack()において、エリアスの発生確率は十分小さく、pに関する2次以上の項は無視できると仮定した。また、エリアス先のサブツリーも元のサブツリーと同じ深さdを持つと仮定した。

## 5. 3 評価モデル

各方式を実現する場合の基本となる処理として以下のパラメータ群を導入する。

Sc: 親EIDから子EIDを一つ取り出す処理量

Sp: 子EIDから親EIDを取り出す処理量

Sr: エントリーの管理情報を読み出す処理量

Se: DITにエリアスエントリーを追加する処理量

Si: エントリーの管理情報を更新する処理量

Scmp: 2つのEIDを比較する処理量

Scmpパラメータを利用して、n1個とn2個のEID集合の交わりを求める処理量の関数Compareを以下に定義する。

$$\text{Compare}(n1, n2) = n1 \cdot n2 \cdot \text{Scmp}$$

また、属性管理部により抽出される、属性値条件を満たす候補エントリー数をmとする。

以下、上記パラメータおよび関数を利用して、各方式におけるエントリー検索時とエリアスエントリー追加時の処理量の評価式を算出する。

### 5. 3. 1 数え上げ方式

#### (1) エントリ検索

ベースエントリから子エントリを順次展開する処理 (Scの項) とエリアスから実エントリを求める処理 (Srの項) と求めたE I D集合と属性値条件の候補E I D集合との交わりを求める処理 (Compareの項) から構成される。

$$P1S(d, m) = Sc \cdot (Entry(l, d, p) + Alias(l, d, p) \cdot 1) + Sr \cdot Alias(l, d, p) + Compare(Entry(l, d, p), m)$$

#### (2) エリアス追加

D I T管理部にエリアスエントリを追加する処理 (Seの項) のみで構成される。

$$P1A(d, m) = Se$$

### 5. 3. 2 廻り方式

#### (1) エントリ検索

ベースエントリの管理情報を読み出してベースE I D集合を求める処理 (Srの項) と属性値条件の各候補エントリに対して親E I Dを廻り、ベースE I D集合と比較する処理 (Compareの項) で構成される。

$$P2S(d, m) = Sr \cdot (Alias(l, d, p) + 1) + \{Sp(D \cdot d) + Compare(D \cdot d, Alias(l, d, p) + 1)\} m$$

#### (2) エリアス追加

D I T管理部にエリアスエントリを追加する処理 (Seの項) とルートまでの各親エントリの管理情報にエリアス情報を追加する処理 (Siの項) で構成される。

$$P2A(d, m) = Se + Si(D \cdot d)$$

### 5. 3. 3 エリアス逆ポインタ方式

#### (1) エントリ検索

属性値条件の候補エントリの各々に対して、親E I D抽出 (Spの項)、ベースE I Dと比較 (Compareの項)、逆ポインタの読み出し (Srの項) の処理をBack()で評価した回数だけ実行する。

$$P3S(d, m) = \{ (Sp + Compare(1, 1) + Sr) \cdot Back(D, d, p) \} m$$

#### (2) エリアス追加

D I T管理部にエリアスエントリを追加する処

理 (Seの項) とエリアスで指される実エントリの管理情報に逆ポインタを登録する処理 (Siの項) で構成される。

$$P3A(d, m) = Se + Si$$

### 5. 4 評価および考察

前節の評価式をもとに、必要な処理量からD I T管理部のスループットを算出した結果を図5～図7に示す。ここでグラフ縦軸のスループットは、D I T管理部の処理能力WをW=1,000,000処理単位において、W/Pにより算出した。

評価に使用したパラメータ値は、文献[6]で作成したプロトタイプシステムでの評価値を参考に決定した。(表2参照)

#### (1) 検索と追加の比率r

エントリ検索とエリアス追加の要求比率r

$$\text{検索} : \text{追加} = 1 - r : r$$

に対するD I T管理部のスループット特性を図5に示す。

数え上げ方式は、新たな管理情報を持たないため、エリアス追加に関する性能が高い。よってrの非常に高い領域では有利となる。しかし、一般的な運用状況では、エリアス追加のトラヒックはせいぜい数%程度と考えられる。従って、検索性能を優先している廻り方式が有利となる領域が広い。

#### (2) サブツリーの深さd

サブツリーの深さdに対するD I T管理部のスループット特性を図6に示す。

数え上げ方式は、dの増加、すなわち、対象エントリ数の増加に伴いスループットが低下する。

表2 評価に使用したパラメータ値

パラメータ	値
D	10
l	5
p	0.01
Sc	60
Sp	60
Sr	300
Se	550
Si	300
Scmp	1

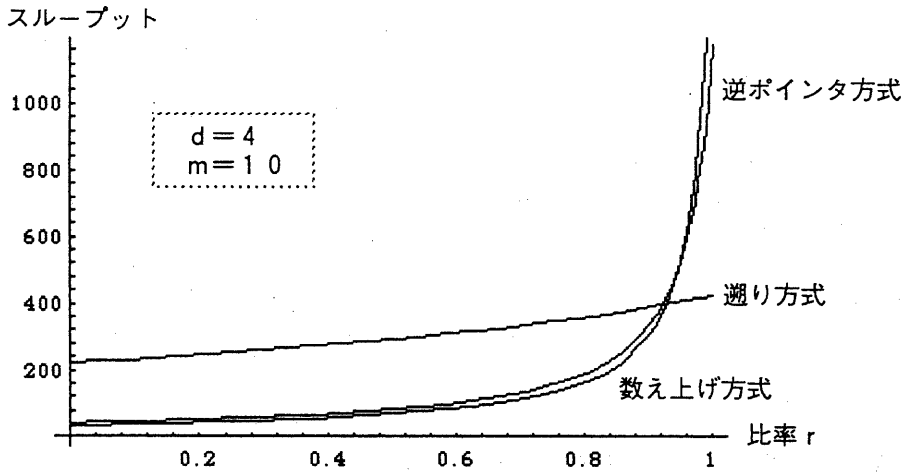


図5 スループット vs 検索/追加比率

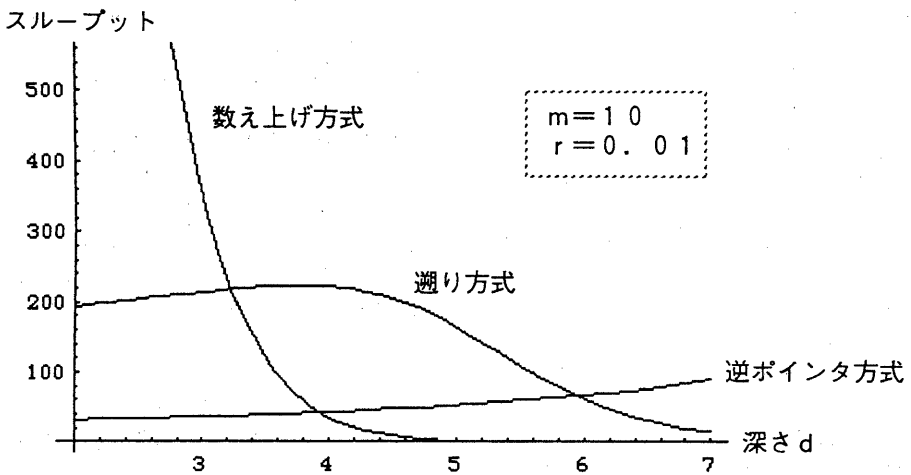


図6 スループット vs 深さ  $d$

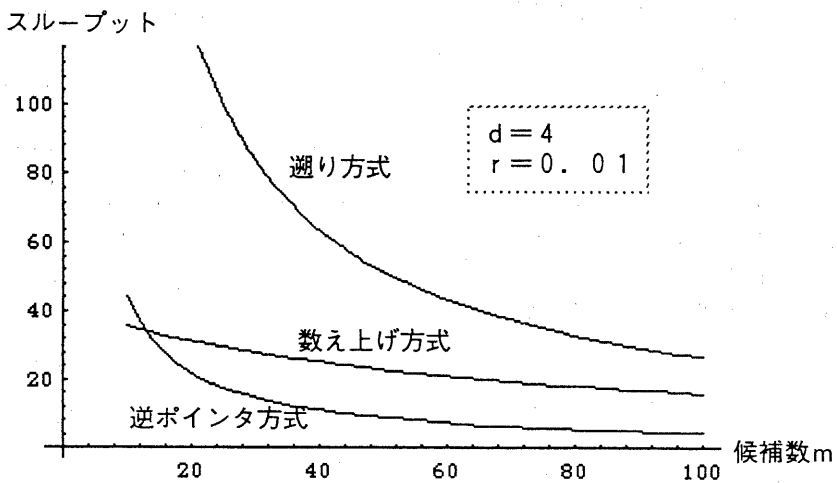


図7 スループット vs 候補数  $m$



これに対し、逆ポインタ方式はdの増加に伴い、ルートまで遡る回数が減少するため、スループットは漸増している。

また、遡り方式は、dの増加によりルートまでの遡り回数が減る要因と、範囲の増加によりエリアスの発生確率が増える要因とのトレードオフからd=4あたりで極大になり、その後、漸減している。

### (3) 属性値条件による候補数m

属性値条件による候補数mに対するDIT管理部のスループット特性を図7に示す。

3方式ともmの増加に対しスループットは減少している。本図のd=4の領域では、図6の結果より、遡り方式が有利となっている。

また、数え上げ方式が、mの増加に対するスループットの低下率が最も小さく、mが大きい領域では逆ポインタ方式よりも有利である。

以上の結果をもとに、各方式の適用領域について考察する。

(1)数え上げ方式は処理論理の単純さを特徴とし、検索対象サブツリーの深さdが3程度の小さい場合には高スループットを実現できることから、システムからのガイダンスや端末アプリケーションの誘導でサブツリー範囲を比較的小さく絞り込める環境で有効である。

(2)遡り方式は、配下のサブツリー内のエリアス情報を保持することにより、ベースEID集合を効率よく抽出し、属性値条件の各候補エントリから親を遡ってベースEID集合と比較することにより、範囲内エントリ数の増加に対しても耐力のある方式である。検索を優先している分、エリアスの管理情報のオーバーヘッドが大きく格納効率の点で劣る。従って、エリアスの比率が低く、しかも検索要求範囲のエントリ数の分散が大きいような環境で有効である。

(3)逆ポインタ方式の適用領域は、上記2方式の中間に位置し、検索性能を向上させつつ、管理情報のオーバーヘッドも遡り方式に比べ少ない。また、深さdが6以上と比較的大きい領域では、遡り方式よりも高い検索性能を達成できる。

最後に、本章で述べたエントリ検索処理方式と4章の情報管理方式との整合性について述べる。案1分離方式は3つの検索方式全てが適用可能であり最も柔軟性がある方式といえる。これに対し、案2統合方式および案3多重方式では、ベ-

スエントリ情報と属性値条件を属性管理部に入力して、RDBMSの検索機能を利用してエントリ検索を実現する方式のため、その特性を活かすためには遡り方式のみ適用可能である。

## 6. あとがき

本稿では、ディレクトリサービスを提供する際のユーザに見える空間であるディレクトリ情報トリー(DIT)に着目して、以下の3点について検討した結果を報告した。

### (1)DIT構成モデル

番号案内サービスを例に、公衆DITの構成モデルについて検討し、管轄企業やサービス・番組等の新たなオブジェクトクラスの導入、およびテナントやチェーン店等も含めた多様な階層構造の表現法について提案した。

### (2)DIT情報管理方式

エントリの属性情報をRDBMSで格納管理することを前提としたDIT情報管理方式に関して、ディレクトリを実現するために保持すべき情報をエントリ間の上下関係及び名前に関するトリー情報と各エントリの属性情報に分類し、両者を分離、統合、あるいは多重管理する方式について比較し、適用領域を明らかにした。

### (3)エントリ検索方式

指定されたサブツリー範囲内にエリアスが存在し、かつエリアス先も検索範囲とするオプションが設定されている場合の効率的なエントリ検索処理方式について検討した。3つの方式案を提示し、評価モデルを導入して、各方式の処理量やスループットを評価した。具体的には、検索/追加の要求比率、サブツリーの深さ、RDBで検索した候補エントリの数等のパラメータに対する各方式のスループット特性を明らかにした。

## <謝辞>

本検討に際し、活発な議論や助言をいただいた情報通信網研究所武藤主任研究員に感謝します。

#### 参考文献

- [1] E. Mackintosh, P.Tan: X.500 and the Electronic Directory: the Business Opportunity, Ovum Ltd, 1991.
- [2] CCITT勧告X.500シリーズ、1988.
- [3] 西山 他：リレーショナル型データベースを用いたOSIディレクトリ情報ベース(DIB)の実現と評価、第42回情報処理学会マルチメディアと分散処理研究会、89-MDP-42-12 1989.
- [4] 中川 他：OSIディレクトリシステムにおけるDIB(ディレクトリ情報ベース)のオブジェクト指向アプローチによる実現、情報処理学会論文誌、Vol.32 No.3、1991.
- [5] C. Amos: Corporate Directory Systems, British Telecommunications Engineering, Vol.9, Jan. 1991.
- [6] 藤原 他：番号案内用ディレクトリシステムの開発、NTT R&D Vol. 41 No. 8、1992.
- [7] 橋田、武藤、戸部、高野：高付加価値型番号案内システム(CUPID)の概要、NTT R&D Vol. 39 No. 6、1990.
- [8] CCITT 92年版X.500シリーズ勧告草案.
- [9] TTC標準 第V巻 第4分冊 高位レイヤプロトコル [ディレクトリ関連].