マルチメディア通信と分散処理 51－7
（１９９１．７．５）

OSIディレクトリサービスの実験

吉田茂樹　片山泰子　松山直道　砂原秀樹

東京大学 生産技術研究所　慶應義塾大学 理工学部　(株) 創夢 研究部　電気通信大学 情報工学科

広域分散環境において、ディレクトリサービスは重要なアプリケーションサービスの一つである。現在 WIDE プロジェクトでは、大規模広域分散環境におけるディレクトリサービスに必要な機能、運用技術に関する検討を行なっている。その第一段階として ISODE パッケージに含まれる QUIPU システムを用い、OSI ディレクトリサービス (X.500) の調査および実験を行なっている。本論文では、WIDE インターネット上での全国規模の運用実験に先立って行なった基礎実験について報告する。この実験では、QUIPU システムに関して、プログラムの概要、動作環境、初期設定、利用方法等を調べるために、5 組織で OSI ディレクトリを構築し利用してみた。これにより、OSI ディレクトリサービスの利用環境を構築するために必要な基礎データが取得できた。今後これを元に全国規模の運用実験を行なう予定である。

Experiment of Using OSI Directory Service

Shigeki Yoshida　Yasuko Katayama　Tadamichi Matsuyama　Hideki Sunahara

Inst. of Industrial Sci., Univ. of Tokyo　Dept. of Math., Keio Univ.　Research Dept., SOUM Corp.　Dept. of Comp. Sci., Univ. of Electro-Comm.

7-22-1 Roppongi, Minato-ku, Tokyo 106, Japan

In the widely distributed environment, the directory service is one of the important application service. In the WIDE project, the requirement for directory service and the "Know How" of its operation in the widely distributed environment are investigated. For the first step, authors are investigating the OSI directory service using the QUIPU system that is the part of ISODE package. This paper describes the result of first experimentation of using OSI directory service for the large scale experimentation on the WIDE internet. To examine outline of the QUIPU system, executing environment and initial setting, the OSI directory is constructed on the five organization. According to this experimentation, the primary data to use the OSI directory service are provided. With these experience, large scale experimentation for using OSI directory services is planed to start as next step.

1 背景

最近日本においても広域なコンピュータネットワークの構築が始まっている。現在その多くは実験・研究用のネットワークであるが、そのネットワークを利用している研究者にとっては実用的なネットワークとして利用されている。今後日本において様々な実用広域コンピュータネットワークが構築されていくと思うが、そこには非常に多くの組織の多くの部署が接続されることになるであろう。それらの組織・部署には様々な情報が存在する。コンピュータネットワークが発展すると、これらの情報にアクセスして、各種の用途に利用したいという要求が出てくる。

このような要求に答え、分散型の特定目的のデータベース機能を提供するのがディレクトリサービスである。ディレクトリサービスは、それを利用して各種の分散情報を扱うユーザアプリケーションや、管理用システムなどから利用される。実用的に使用されている既存のディレクトリサービスとしては、NIS (Network Information System) [1] や BIND (Berkeley Internet Name Domain) [2] [3] [4] があるが、これらは広域分散環境での使用に耐えうるものではなかったり、特定の利用目的に特化されすぎていたりしてディレクトリサービスに求められる機能を十分に満たしてはいない。

WIDE (Widely Integrated Distributed Environment) プロジェクト [5] では、広域分散環境を構築するための様々なテーマについての研究を行なっているが [6] [7]、我々のワーキンググループでは WIDE の環境下において、様々なユーザやアプリケーションからの要求を満たすディレクトリサービスとはどうあるべきかについての研究を行なっている。また同時に、OSI アプリケーションの実用性の調査と、既存のネットワークから OSI をベースにしたネットワークへの移行のノウハウの蓄積も目的としている。そのために、今後実用的に使用されるであろう OSI ディレクトリサービスについて利用実験を行ない、WIDE ディレクトリサービスとしての利用の可能性、その問題点などを調査することにした。この実験では WIDE インターネット上で OSI ディレクトリサービスの全国規模の運用実験を行ない、どういう使い方があるか、どのくらい使いものになるか、ユーザの要求にはどういうものがあるか、などを調査する。本論文では、全国規模の運用実験に先だって行なった、ワーキンググループ内での基礎実験について、その概要、実験内容、問題点について報告する。

2 OSI ディレクトリサービス

現在、広域のマルチベンダー型のコンピュータネットワークを構築する場合には、TCP/IP をベースにした製品が使用されることが多いが、将来的には OSI (Open Systems Interconnection) をベースにした製品が普及していくと思われる。OSI は現在、規格が定まってはいるが、その実装規約が各国で検討されている段階であり、OSI をベースにした製品はまだまだ普及はしていない。

OSI においてディレクトリサービスは CCITT 勧告 X.500 シリーズ/ISO 9594 として規定されている [8]。OSI ディレクトリサービスは、メッセージハンドリングシステムやファイル転送といった OSI アプリケーション、OSI 管理プロセス、OSI プロトコルなどに必要な情報を提供するための機能を備えている。なお、OSI ディレクトリ自身は汎用のデータベースではないが、汎用データベースを構築するための基盤を提供することができる。

OSI ディレクトリサービスは、ディレクトリを構成する DSA (Directory System Agent) 群と、ディレクトリにアクセスしてサービスを受けるための DUA (Directory User Agent) から構成される。また、ディレクトリが持っている情報は、DIT (Directory Information Tree) と呼ばれる木構造の名前空間にマッピングされている。

ディレクトリは複数の組織に存在する複数の DSA から構成されており、それぞれが DIT の名前空間のある範囲の情報を保持している。自分が保持していない情報は他の DSA に問い合わせることによって得ることができるが、その問い合わせ形態として、Referral, Chaining, Multicasting を利用することができる。Referral は DUA が DSA から次の DSA の情報を返されることにより、DUA がそれらの DSA に順番に問い合わせを行なう。Chaining は DSA が次の DSA に問い合わせを行ない、その DSA が情報を持っていない場合にはその DSA がさらに次の DSA に問い合わせを行なう。Multicasting は DSA が次の DSA に問い合わせを行ない、その DSA が情報を持っていない場合には元の DSA がさらに次の DSA に問い合わせを行なう。図 1 に OSI ディレクトリの構成と Chaining による問い合わせの概念を示す。また図 2 に OSI ディレクトリの DIT の名前空間の例を示す。

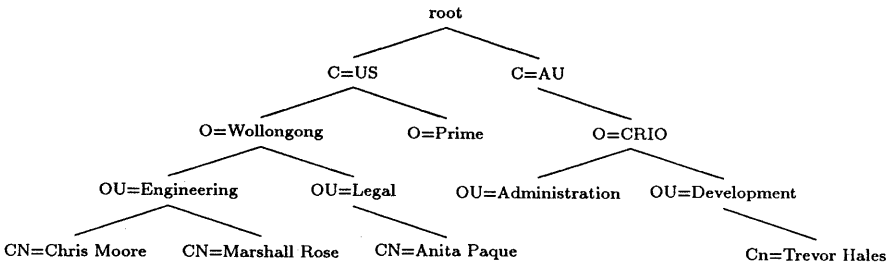

図 2: OSI ディレクトリの名前空間の例

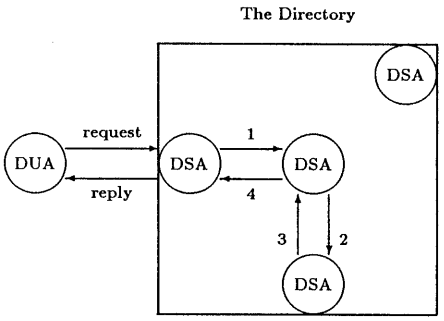

図 1: OSI ディレクトリと問い合わせの概念

3 ISODE と QUIPU

実験にあたり、OSI ディレクトリサービスシステムとして、比較的簡単に入手し、利用することができるという点から、ISODE (ISO Development Environment) に含まれている QUIPU を使用することにした。

3.1 ISODE

ISODE は本来、既存の TCP/IP ベースのネットワーク上で OSI を学習するために作成されたパッケージである [9]。ISODE はフリーソフトウェアとしてソースコードが公開されており、誰でもソースコードを見て、実際に使用してみることができる。そのため現在では、ISODE は学習教材としてだけではなく、OSI 上位プロトコル層の実装例、OSI サービスを利用するためのプロダクト、TCP/IP から OSI への移行手段などとしてとらえられている。OSI は規格を定めてはいるが、その実装方法については規定していない。そのため、いかにして OSI を実装するかの一例として ISODE を参考にすることができる。

ISODE は C 言語で書かれており、BSD 系 (4.2, 4.3)、System V 系 (R2, R3) の UNIX 上で利用できる [10]。現在の最新バージョンは 6.8 で、TCP/IP 上に構築されたトランスポート層サービスおよび、OSI のセッション層、プレゼンテーション層、アプリケーション層の各機能が提供されている。また、OSI アプリケーションとしてファイル転送、仮想端末、ディレクトリサービス、ネットワーク管理の各アプリケーションを使用してみることができる。

3.2 QUIPU

QUIPU は ISODE に含まれているディレクトリサービスパッケージであり、CCITT X.500/ISO 9594 に準拠している [11] [12] [13]。QUIPU には 1 つの DSA と数種の DUA が含まれている。QUIPU では OSI ディレクトリサービスの規格に対して、以下のような実装上の拡張が行なわれている。

- EDB (Entry Data Block) による情報の管理
- アクセスコントロール
- ユーザ認証
- サービスコントロール

また QUIPU の DSA は、DSA 間の問い合わせに Referral と Chaining を使用し、Multicasting は使用しない。

今回の基礎実験では、比較のために ISODE 6.6 と 6.8 に含まれている両方の QUIPU を使用した。ISODE 6.6 と 6.8 の QUIPU では、EDB の記述方法、DSA の情報の管理方法などが変更になり、情報の検索方法、認証機能などが改良されている。

3.3 EDB による情報管理

QUIPU ディレクトリが保持すべき各情報は、EDB フォーマットと呼ばれる形式の ASCII テキストファイルに、情報ごとにエントリとしてまとめられて記述されている。この EDB ファイルの 1 行目には、"MASTER"、"SLAVE"、"CACHE" のいずれかの文字列が記述され、そのファイル内の情報がマスターデータであるのか、スレーブコピーであるのか、キャッシュされたエントリであるのかを示している。2 行目には、その情報が作成された日時が EDB ファイルのバージョンとして記述される。この日時は DSA 間での情報の転送の際に使われるタイムスタンプで、情報が更新されたものであるかどうかをチェックする。

このヘッダに引続き、エントリを記述する。空白行で区切られたものを一つのエントリとして扱い、エントリの第 1 行目はそのエントリを示す RDN (Relative Distinguished Name) でなければならない。各行には属性とその値が "=" で関係づけられている。属性のひとつである `objectClass` がそのエントリのオブジェクトの種類を定義していて、その値は、どのような型の属性を必須、または任意に持つべきかを示している。例えば、`objectClass` が人に関するエントリを意味している値を持っている時、そのエントリは `surName` という属性を必ず持たなければならない。

EDB ファイル内のエントリは、DSA 内に保持される情報として DSA が起動される時に、ハードディスクから読み込まれる。ハードディスク上では DIT の各レベル用の EDB ファイルが、対応した UNIX ファイルシステムの木構造ディレクトリ上に存在する。EDB ファイルの中で定義したエントリに children がある時には、その RDN を名前に持つサブディレクトリに EDB ファイルを持つことになる。

図 3 に今回の実験に使用した EDB ファイルの一部の例を示す。

3.4 QUIPU の DUA

DUA はユーザに代わって問い合わせをしてその結果を表示するユーザインタフェースプログラムである。X.500 では DUA の使うべきプロトコル DAP (Directory Access Protocol) のみが定義されているので、DUA の実現方法は様々なものになりうる。

QUIPU には 5 つの DUA が用意されている。

dish(The DIrectory SHell) QUIPU の基本的な DUA であり、全ての DAP が実装されている。このインターフェースを利用してユーザは DUA にアクセスすることができる。

sid(Steve's Interface to Dish) dish DUA を使いやすく実装したスクリプトで、人と組織の検索に的を絞ったものである。

fred(FRont End to Dish) OSI ディレクトリの複雑な部分を隠蔽した、dish へのインタフェースとして実現され、White Pages 用インタフェースを提供する。

widget curses を基にしたウインドウ・インタフェース。X.500 の一部分しか実装されていない。

sunint suntools を基にしたウインドウ・インタフェース。X.500 の一部分しか実装されていない。

dish には様々なコマンドやフラグが用意されており、OSI ディレクトリサービスの利用実験をする際の基本的な DUA として利用できる。dish はかなり強力なインタフェースであるが、その分、複雑なものになっている。

4 OSI ディレクトリサービスの実験

4.1 実験スケジュールと基礎実験

OSI ディレクトリサービスの実験は、次の 2 フェーズに分かれる。

1. ワーキンググループ内における基礎実験
2. WIDE インターネット上での全国規模の運用実験と、国際的運用実験への参加

基礎実験は、実験に使用する OSI ディレクトリサービスシステムの概要、動作環境、初期設定、利用方法、利用上の問題点等を調査する。また、この基礎実験では、各組織の管理者が容易に OSI ディレクトリサービスを運

```
MASTER
19901221085731Z
cn=Manager
acl=
cn= Manager
aliasedObjectName= c=JP@o=Keio University@ou=Shonan Fujisawa Campus@cn=Yasuko Katayama#
objectClass= top & quipuObject
objectClass= alias

cn=Postmaster
acl=
cn= Postmaster
aliasedObjectName= c=JP@o=Keio University@ou=Shonan Fujisawa Campus@cn=Yasuko Katayama#
objectClass= top & quipuObject
objectClass= alias

ou=Shonan Fujisawa Campus
masterDSA= c=JP@cn=KEIO#
acl= others # read # entry
acl= others # read # default
acl= others # compare # attributes # accessControlList$userPassword
ou= SFC & Keio Univ. SFC
treeStructure= quipuNonLeafObject & organizationalUnit
treeStructure= alias & pilotPerson & organizationalRole
objectClass= top & quipuObject & quipuNonLeafObject
objectClass= organizationalUnit
```

図 3: EDB の例

用し、全国規模の運用実験に参加することができるようなガイドの作成も目的としている。

全国規模の運用実験は、ワーキンググループ内での実験の結果を元にして、WIDE インターネット上で、WIDE に参加している全国の各大学や企業の研究機関からなる大規模な OSI ディレクトリを構築し、ディレクトリサービスとして要求される機能、大規模広域ネットワーク上でのディレクトリサービスの運用における問題点、などについて考察していくものである。これと並行して、ヨーロッパやアメリカを中心に行なわれている、国際的な OSI ディレクトリサービスの運用実験への参加を行なう。

今回はワーキンググループ内の基礎実験を行なった。

4.2 DSA の配置

ワーキンググループ内での基礎実験として、ワーキンググループに参加しているメンバーの 5 つの組織に、各レベルの DSA を配置して実験を行なった。c=JP の MASTER DSA として WIDE 内に cn=Japan Master を設定し、これが o=WIDE の MASTER DSA も兼ねることにした。さらに、東京大学生産技術研究所に o=University of Tokyo および ou=Institute of Industrial Science の MASTER DSA として cn=U-Tokyo を、(株) 創夢研究部に o=SOUM および ou=RD の MASTER DSA として cn=SOUM を、慶應義塾大学環境情報学部に o=Keio University および ou=Shonan Fujisawa Campus の MASTER DSA として cn=KEIO を、電機通信大学情報工学科に o=University of Electro Commuinications および ou=Computer Science の MASTER DSA として cn=UEC を設定した。

構築した DIT 名前空間とそれぞれの DSA の配置の様子を図 4 と表 1 に示す。

4.3 EDB の設定

EDB ファイルはそれぞれのノードごとに作成する必要がある。例えば cn=U-Tokyo の DSA は c=JP@o=University of Tokyo と c=JP@o=University of Tokyo@ou=Institute of Industrial の情報を管理するが、この DSA は起動時にこれらのノードの用の 2 つの EDB ファイルを読み込む。EDB ファイルは QUIPU に含まれている dsaconfig を使用して生成した [14]。各組織の DSA は TCP/IP ネットワークに接続されているホスト上で動作しているため、DSA の Presentation Address は IP アドレス形式で指定した。

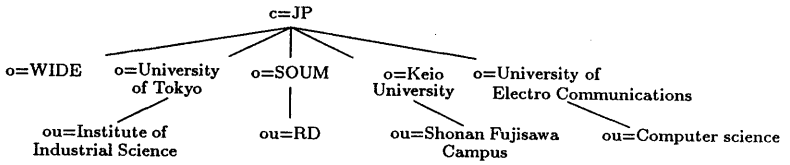

図 4: 実験用名前空間

表 1: DSA の配置

DSA name	Master EDB's Held
cn=Japan Master (SUN3/480 SunOS 4.0.3)	c=JP c=JP@o=WIDE
cn=U-Tokyo (SUN4/370 SunOS 4.0.3)	c=JP@o=University of Tokyo c=JP@o=University of Tokyo@ou=Institute of Industrial Science
cn=SOUM (SUN3/60 SunOS 4.0.3)	c=JP@o=SOUM c=JP@o=SOUM@ou=RD
cn=KEIO (SUN4/470 SunOS 4.0.3)	c=JP@o=Keio University c=JP@o=Keio University@ou=Shonan Fujisawa Campus
cn=UEC (SUN4/330 SunOS 4.1.1)	c=JP@o=University of Electro Communications c=JP@o=University of Electro Communications@ou=Computer Science

4.4 dish によるディレクトリへのアクセス

この実験では、DUA として dish を使用した。dish には表 2 のようなコマンドが用意されている。

表 2: dish のコマンド

list	現ノードの children を表示する。
showentry	エントリの DN を表示する。
moveto	DIT を移動する。
search	指定したオブジェクトを検索する。
add	DIT に新しいエントリを追加する。
delete	DIT からエントリを削除する。
modify	存在しているエントリを更新する。
modifyrdn	エントリの RDN を更新する。
showname	エントリの名前を表示する。
compare	属性を指定した値と比較する。
squid	dish の現在の状態を表示する。
bind	ディレクトリに接続する。
unbind	ディレクトリとの接続を切断する。

なお、dish が起動される時には `.quipurc` というファイルが読み込まれる。これは、ユーザごとに所有されるファイルで、ユーザ名やパスワードの他、ディレクトリを利用する際の環境を設定することができる。パスワードなどの情報も含まれるので、セキュリティのためにこのファイルはユーザによって保護されなければならない。また、使用するコマンドに常にフラグをつけて実行するようにする設定や、エントリを参照する RDN にニックネームをつけることも可能である。

実験では、自組織の DSA の動作、DSA 相互の問い合わせといった基本動作の確認のため、主に moveto, list, showentry, squid コマンドを使用した。dish の実行例を図 5 に示す。ここでは o=WIDE において cn=KEIO の管理情報を表示させている。

5 考察

QUIPU では、情報を EDB フォーマットで書かれたファイルに ASCII テキストとして保持しており、DSA の起動時に管理範囲内の EDB ファイルを全てメモリに読み込んでいる。当然、登録されるエントリの数に比例して EDB ファイルおよび DSA の実行時メモリの大き

```
Dish -> moveto c=JP
Dish -> list
21  commonName=SOUM
22  commonName=KEIO
23  organizationName=WIDE
24  commonName=Japan Master
25  organizationName=Keio University
Dish -> showentry cn=KEIO
quipuVersion          - quipu 6.6 #42 (endo.wide.sfc.keio.ac.jp) of Mon Dec  3 18:06:12 JST 1990
eDBinfo               - ROOT ( FROM c=JP@cn=Japan Master )
eDBinfo               - c=JP ( FROM c=JP@cn=Japan Master )
eDBinfo               - c=JP@o=Keio University ( FROM c=JP@cn=Japan Master )
accessControlList     - (default)
manager               - countryName=JP
                        organizationName=Keio University
                        commonName=Manager
manager               - countryName=JP
                        commonName=Manager
userPassword          - HIDDEN
supportedApplicationContext - X500DSP & X500DAP & quipuDSP
presentationAddress   - '0101'H/Internet=133.27.48.2+17003
description           - Keio University
commonName            - KEIO
objectClass           - dSA & applicationEntity & top & quipuObject & quipuDSA
Dish -> squid
Connected to Japan Master at '0101'H/Internet=133.4.11.11+17003
Current position: @c=JP
User name: @c=JP@o=Keio University@ou=Shonan Fujisawa Campus@cn=Yasuko Katayama
Current sequence 'default'
```

図 5: dish の実行例

さは増えていく。このことは、検索を行なう際の速度にも影響してくるであろう。実験では EDB ファイル上で 500 バイト程のエントリが、DSA に読み込まれると約 1 KB のメモリを消費することを確認した。また、エントリを読み込まない状態でも、DSA は 2 MB 程のメモリを消費していた。同じマシンで動作している NIS サーバが 120 KB 程、BIND サーバが 650 KB 程のメモリ消費であることと比較すると、格段にメモリ消費量が多いことが分かる。全国規模の実験を行なう際には、参加する組織で DSA を動作させるマシンを慎重に選定しなければならない。

データの問い合わせに要する時間は、自組織の DSA への問い合わせの場合で平均 0.1 秒程、他組織の DSA への問い合わせの場合で平均 2 秒程であった。ただし、大量のエントリを持った DSA に対する問い合わせは行なっていないので、これは今後の実験課題になろう。なお、今回の実験では 9.6 Kbps の回線を介した DSA 間の問い合わせを行なった所もあるので、DSA の検索速度よりは回線速度の方が影響が大きいと思われる。また試験的に、ヨーロッパやアメリカで構築されている QUIPU ディレクトリの DSA に対する問い合わせを行なってみたが、平均 5 秒程で情報を取り出すことができた。これは BIND サーバによるアメリカやヨーロッパなどへの名前情報の問い合わせの場合の、平均 3 秒程に比べてそれほど遜色がない速度である。

EDB ファイルについては、オブジェクトのパスワードが文字列で記述されていたり、DIT に対するアクセス・コントロールもこのファイルの中で記述されていたりするのでセキュリティ上の問題が残されている。各ユーザがそれぞれ UNIX のファイルの保護機構を利用して管理することになっているが、不十分のように思える。さらに、アクセス・コントロールの機構そのものに対しても不十分なところが見受けられる。パスワードに頼った認証確認が行なわれている (弱い認証) が、大規模広域ネットワークで運用するにあたっては、もっと強固な認証確認がなされるべきである。OSI ディレクトリサービスでもその試み (強い認証) はなされているが、十分とは言えない。

最後に、QUIPU ディレクトリサービスでは、静的な情報、つまり、一度持ったら更新されるまでにかなりの時間が掛かるような情報しか扱っていない。標準的に広くディレクトリサービスが利用されるようになれば、動

的な情報、例えば、ある研究者が物理的にどこに存在しているか、などといった情報も必要に応じて要求できるようになっていることが望まれる。[1]

6 まとめ

ISODE パッケージの QUIPU システムを用いて、OSI ディレクトリサービスを利用するための基礎実験を行ない、OSI ディレクトリの構築とそれを利用するための基礎データを得ることができた。

今後は今回得られたデータを元に、大規模分散環境での OSI ディレクトリサービスの運用実験を開始して、上記の問題を含めてさらに実験と検討を進めてゆく予定である。

謝辞

本実験を行なうに当たり、指導して頂いた慶應義塾大学の村井純助教授、多大なる助言と提案を頂いた WIDE 研究会のメンバーに感謝します。

参考文献

[1] Sun Microsystems, "The Network Information Service", Chapter 16, System and Network Administration, Sun Microsystems, Inc., 1990

[2] P. Mockapetris, "Domain names - Concepts and Facilities", RFC-882, USC/Information Sciences Institute, 1983.

[3] P. Mockapetris, "Domain names - Implementation and Specification", RFC-883, USC/Information Sciences Institute, 1983.

[4] J.M. Bloom and K.J.Dunlap, "Experiences Implementing BIND, A Distributed Name Server for the DARPA Internet", Proceedings of the USENIX 1986 Summer Conference, 1986

[5] 村井 純, 他, "大規模分散環境 WIDE の構築", マルチメディアと分散処理研究会資料, 情報処理学会, 1989

[6] 村井 純, "WIDE プロジェクトレポート 1990", 第 15 回 UNIX シンポジウム予稿集, 日本 UNIX ユーザ会, 1990

[7] 村井 純, 他, "WIDE 1990 年度研究報告書", WIDE プロジェクト, 1991

[8] "The Directory - Overview of concepts, models and services", ISO/IEC 9594, CCITT Recommendation X.500, 1988

[9] Marshall T. Rose, "The Open Book", Prentice-Hall International, Inc., 1990

[10] Marshall T. Rose, "The ISO Development Environment User's Manual", The Wollongong Group, Palo Alto, 1989

[11] Stephen E. Kille, "The QUIPU Directory Service ", IFIP WG 6.5 Conference on Message Handlink Systems and Distributed Applications, 1988

[12] Stephen E. Kille, "The design of Quipu", Research Note RN/89/19, Department of Computer Science, University College London, 1989

[13] Paul Barker, Colin J. Robbins, "Directory nabigation in the Quipu X.500 system", 1989

[14] Marshall T. Rose, "NYSERNet/PSI White Pages Pilot Project: Administrator's Guide" Performance Systems International, Inc., 1991

[15] Yuko Ishikawa, Kazuyuki Saga, Atsushi Onoe and Jun Murai, "Resource Management Function of WIDE", Proceedings of the 4th International Joint Workshop on Computer Communications, 1989

[1] なお、WIDE ではこのような動的な情報の収集のためのメカニズムの研究も行なっている [15]。