広域ネットワーク環境における分散型情報管理ツールの開発

中村 誠一* 山口 英† 尾家祐二* 宮原秀夫‡
*九州工業大学 †奈良先端科学技術大学院大学 ‡大阪大学

概要

ネットワーク上に分散して存在する個人およびハードウェアの情報を管理取得する方法として，これまでにHesiodやX.500を用いたシステムがある．しかし，広域ネットワークの発達によりそれらの情報を容易に管理取得するのは困難になってきている．本稿では，広域ネットワークにおける分散された情報を容易に管理取得するシステムの設計手法に関する考察を行い，それに基づいて開発を行った際の実装技術を報告する．

1 はじめに

近年，ネットワークで接続されているワークステーションの数が爆発的に増えている．今後，ネットワークの相互接続が進み，これらの管理・保守・利用者の個人情報の取得等は一段と困難になっていくと考えられる．

たとえば，障害が発生した場合には，その原因を特定するためにネットワークを構成する各々のワークステーションの情報を取得する必要がある．そのためには，各々のハードウエアに直接アクセスしなければならず，多くの時間と高度な技術を必要とする．したがって，ワークステーションを管理するためには，高度な技術を持つ多くの管理者が必要である．

しかし，現状では，このような管理者が不足しているため，管理者に非常に負荷がかかりワークステーションの管理が困難になっている．これらの問題を解決するためには，ある程度広い範囲のネットワーク上のワークステーションの利用状況や負荷状況等を簡単に取得できるような機構が必要である．

他に，ネットワーク上に多くの利用者が存在するが，それぞれの個人と連絡をとったり，その個人の電話番号などの個人情報を取得したいなどの要求がある．しかし，このような情報を取得するためには，実社会組織下での個人の所属とは無関係であるハードウエアのネットワークアドレス等を要求している個人が知らなければならない．そこで，個人が実社会において所属している組織の名称を手がかりに，その情報を取得するための機構も必要である．

これらの問題を解決するためにディレクトリサービスが提案されている．このサービスは，データに対してユーザーフレンドリな名前をつけることができる．これは，人に限らずディレクトリの利用者に検索する対象の名前を簡単に特定し，付加的な情報を検索し提供する．ユーザーフレンドリな名前をつけることは，単に人に対しても，ハードウェアに対してもある検索対象に関する情報を短く覚えておくことが容易に可能となる．ディレクトリを利用することの目標は，ユーザーの要求する情報を覚え易い形で指し示し，その情報に対する詳細を提供する

Development of directory service of wide-area networks
by Seiichi Nakamura*, Suguru Yamaguchi†, Yuji Oie*, Hideo Miyahara‡
*Kyushu Institute of Technology,
†Advanced Institute of Science and Technology, Nara,
‡Osaka University

ことである．

いくつかの研究に於いては，この様な計算機情報と個人情報を取得する機構を備えたディレクトリサービスの開発を行っているが，ワークステーションの管理に的を絞ると，動的情報をうまく扱えない，計算機に対して負荷が重すぎる等の問題がある．

本研究では，広域ネットワーク環境における分散情報管理ツールの検討を行い，どの様な情報が必要であるかを考察し，この情報を自動的に収集する機構，及びそれを利用するための機構を WINDS(Wide-area INternet Directory Services) と名付けて開発する．

2 既存のシステムとの比較

ネットワークに分散された情報を管理提供するためのシステムは，これまでに MIT で開発された Hesiod[3] を代表的なシステムとして挙げることができる．また，CCITT 勧告の X.500[4] に基づいて同様な機能を果たすシステムを構築することが可能である．以下，Hesiod と X.500 についてそれらの特質及び問題点を考察する．

2.1 Hesiod

このシステムは，MIT のアテナプロジェクトで開発されたものである．Hesiod は，ネームサービスの他に、ユーザ情報の提供を行うことができる．基本的な機能は，ネームサーバである BIND と同等であるが，特徴的な機能は，ファイルをロックしている人の名前情報や，システムライブラリ，RVS，NFS でのファイルのロック，プリンタ情報，メールボックスの場所，ポート情報などを扱うことができる．この様に多機能ではあるが，次章で述べる機能を含んでいない。例えば、3.3 で述べるような名前空間において九州工業大学にある学部の一覧検索などを行う機能がない。

2.2 X.500

X.500 は，CCITT が制定したディレクトリサービスの国際標準である．X.500 の規格では，DUA(Directory User Agent) と DSA(Directory System Agent) の 2 つのシステムを定義している．DUA は，ディレクトリをアクセスするためのクライアントであり，DSA は，ディレクトリの情報を保持しているサーバーである．このシステムは，動的情報を管理するための機構を備えていない．また，ネットワークやハードウェアに対して大きな負荷をかけてしまう．

この X.500 の構成を図 1に示す．

ここで，ユーザーがディレクトリシステムに対して情報の提供を求めた時，まず，ユーザプロセスが DUA に対し情報の提供を依頼し，その後 DUA が DSA のサーバー群に対し情報の提供を依頼する．情報の依頼を受けた DSA は，もし自分自身にその情報をもたなければ，他の DSA に対し情報の提供を求める．もし，もっていればその情報を依頼元に対し情報を提供する．

3 WINDS の設計

WINDS の目的は，ディレクトリサービスとして個人情報，ハードウェア情報を提供することである．そのためには，幾つか考慮すべき点がある．以下，その点をあげると，

1. 階層化された情報に対し、指定された階層下 (例えば実社会の組織下) の情報の一覧提供
2. 効果的な情報の提供
3. 動的および静的情報の取得方法
4. 情報に対するセキュリティ対策
5. 情報の共有や再利用できる様な形での保存
6. ネットワークやハードウェアに対して負荷をかけない構造

等を考慮しながら設計を行う．

3.1 管理情報

3.1.1 情報の選択

管理者が管理を行うために必要と思われる情報は，個人情報，ハードウェア情報等がある．今回必要と思われた情報を表 1 にあげる．これらの情報は 1 つの計算機で一括管理するよりも分散型データベースで管理した方が計算機の負

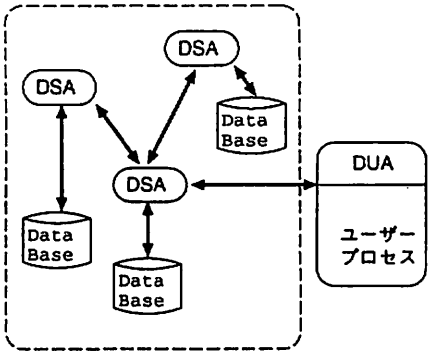

図 1: X.500

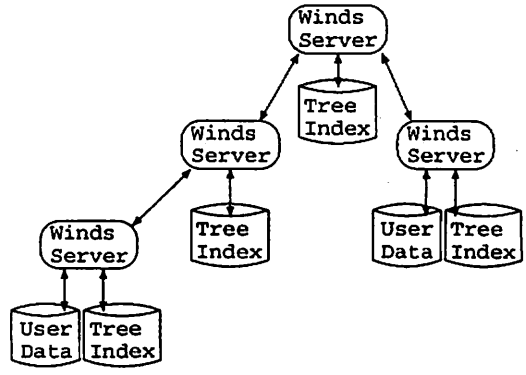

図 2: データベース構成

表 1 管理情報

個人	実名，連絡先の電話番号，住所，所属名， 連絡先，利用計算機，メールアドレス，．．．
ハードウェア	CPU 名，OS 名，接続ハードウェア名，機能
トラフィック	特定ハードウェアの負荷状況
利用者数	特定ハードウェアの利用者数，利用状況
ポート情報	特定ハードウェアのポート情報
コネクション	特定ハードウェアのコネクション情報
ネットワーク	ネットワークに対する管理者情報

荷を下げることができる．また，動的情報である計算機の負荷情報等は，定期的に取得するためのシステムが必要である．

3.2 データベース設計

データベースは，1つのサーバに対し1つのエンジンを起動させ，複数のデータベースの管理をさせる．また，エンジンへのインターフェースは，RPC を使う．データベースは，UC Berkeley で開発された Postgres を使用し，他のシステムとのデータと共有やデータの再利用を可能とする．

3.2.1 データベース分散

データベースは，2つに分離管理される．1つは，実際のデータを格納するデータベース．もう一つは，インデックス情報を管理するデータベースである．まず，実際のデータを格納するデータベースは，個人情報や，ハードウェアの動的情報等を管理し情報の提供を行う．インデックス情報は，実際のデータを格納しているデータベースが存在している計算機のアドレス情報を格納している．図 2のように，管理すべきデータは，各々のサーバで分散管理されている．

3.2.2 インデックス情報

全体のデータベースは，ツリー構造をなし，その中で分散してデータを管理する．このツリー構造の各部分で実際のデータを管理している．このデータを検索するためには，WINDS サーバが稼働している各々の計算機のアドレス情報 (インデックス情報) が必要になる．

しかし，全ての WINDS サーバに関するインデックス情報を各々のサーバが所有すると，ネットワークの規模が大きくなるにつれデータの量が膨大になり，さらに資源の面でも，負荷の

面でも不利になる．データの検索時間や負荷分散等を考えるとサーバ群を幾つかのグループに分けて，インデックス情報の分散化を行なうことが望ましい．具体的には，同じグループ内のサーバは，そのグループに属するサーバのインデックス情報は所有するが，他のグループに属するサーバのインデックス情報は所有しないようにする．

ことなるグループ内におけるインデックス情報の検索については3．3において述べる．

3.2.3 レコード単位のアクセス制御

個人情報などのレコードで，多くの人に対し，公開を行いたくない場合がある．この様な問題を解決するために，UNIXのファイルシステムのようなアクセスレベルの設定を行えるようにする．通常のアクセス制御の利用は，同じ科に所属する人には，全てのデータに対しアクセス可能とし，別の科では知らせたくないようなことを行うと思われる．よって，次のような方法をとる．

このアクセスレベルの設定は，名前階層に対しおこなう．たとえば，

九工大．情報工学部．電子．seichan．tel

の様な名前空間のデータの場合，

-．-．r．r

のようにする．－は，アクセスできないことを示し，rは，アクセスできることを示す．つまり，上記の記述は，電子情報工学科の人と，seichanが，読み出すことができることを示す．この様にする事により，ある程度のアクセス制限が可能となる．

また，これらのアクセス制御のデータは，パケット中にフラグとして転送する．

3.3 名前空間の構造

このシステムでは，人は，組織に属するものとして，現在の社会的な組織階層をそのまま用いて個人情報を管理する．ハードウエアの情報はインターネットワークの名前の階層構造を用いて管理する．この様にする事により特定の人物の探索を容易に行うことができ，管理面でも良い結果をもたらす．実際は次のような組織割りを行う．

電子情報工学科＋尾家研究室＋中村誠一

ここで，記述している＋は，階層を表す．この様な階層を図で表すと，図3のようになる．また，今回開発した，WINDSサーバは，図4のような構成をとる．このツリー構造は，組織の構造を示している．

3.4 管理情報の検索

2つの段階に分けて検索を行う．

1. 検索したい管理情報を所有しているWINDSサーバのアドレスを検索する
2. 取得したアドレスで稼働中のWINDSサーバから管理情報を取得する

管理情報を所有しているWINDSサーバのアドレスを検索する時，同じグループ内に所属しているWINDSサーバのときは，該当するWINDSサーバのアドレスを提供することができるが，同じグループ内に所属していないときは，そのグループの上位サーバのアドレスを返し，その上位サーバに対しアドレスの取得を依頼する．

もし，その上位サーバが所属しているグループ内に要求された情報を提供するサーバなければ，その上位サーバに対し同様のことを繰り返す．その後，取得したアドレスを用いて稼働中のWINDSサーバに対し，管理情報の提供を依頼する．

例えば，サーバＩに接続されているユーザがサーバＥで管理されている情報を取得しようとする場合について、図4を用いて説明する．まずサーバＩに対して必要とする管理情報を所有しているWINDSサーバのアドレスを要求する．

このとき，サーバＩがサーバＥのアドレスを所有していれば，サーバＥのアドレスを返す．しかし，この場合サーバＥのアドレスは所有していないのでサーバＩは，同じグループの上位サーバＧのアドレスを返す．その上位サーバＧに対してもう一度，問い合わせる．このときサーバＧは同じグループに所属しているサーバＥのアドレスを所有しているのでユーザプロセスに対し，サーバＥのアドレスを返す．その後，ユー

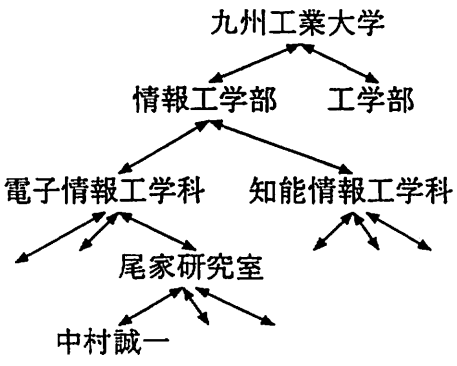

図 3: 名前空間

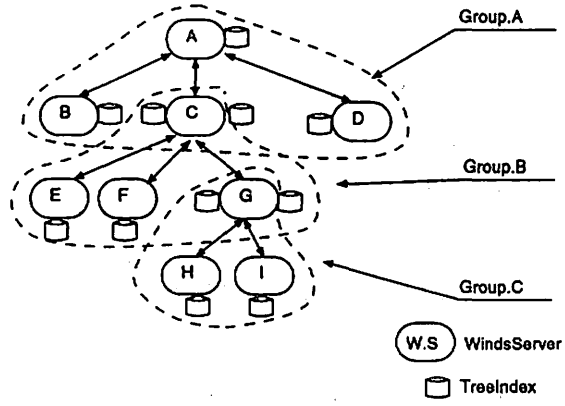

図 4: グループ構造

ザプロセスは直接サーバEに対し，情報の提供を求める．

3.5 動的データ収集

動的データを収集するには幾つかの方法がある．まず，専用の動的データを収集する専用のサーバを作動させ，そのサーバによって動的データを収集しデータベースに登録するシステムと，既にある動的データ収集システムを用いて動的データを収集する方法等がある．今回開発するシステムは，なるべくハードウエアに負荷をかけないようなシステムを目指しているので専用のサーバを新たに開発するのではなく，既存の動的データシステムを用いることにする．

この既存のシステムには，SNMP[5] という通信プロトコルを用いた動的データ収集システムがある．このSNMPは，数多くのハードウエア上で実装されており，利用は簡単なプロトコルによって行うことができる．しかも，UNIXシステム以外のネットワークルータ等のハードウエアにも実装されている．そこで，このSNMPを利用して動的データの収集をWINDSで用いることする．

動的データの提供はWINDSのインターフェースを使用して提供することによってより容易に動的データの提供をすることができる様になる．

4 WINDSの実装

前章で述べられた設計手法に基づき実装を行ったので、ここにその実装技術を報告する。実装したシステムの構成は図5のように表すことができる．以下，このシステムの構成図に沿って各々の部分について要素する．

4.1 システム構成

このシステムは，大きく分けて4つの部分に分かれている．データベース管理部，サーバ起動部，動的情報収集部，ユーザインターフェース部である．それぞれの機能について以下に説明する．

4.1.1 サーバ起動部

起動するべきデータベース管理部に対し，どのようなデータベースファイルを管理すべきかを指定し，データベース管理部を起動後終了する．

4.1.2 データベース管理部

データベース管理部は，すべてのすべての内部関数郡は，RPCプロシージャによって記述されている．データベース自身は，通常のテキストファイル及び，UC Berkeley で開発されたPosgresを用いることができる．また，1つのデータベース管理部によって複数のデータベースを管理することができる．また，サーバとして常

時作動している．

4.1.3 動的情報収集部

指定されたハードウエアの動的情報をSNMPを使って収集し，データベース管理部のRPCプロシージャを使ってデータベースに登録する．また，収集すべきハードウエアは，収集先指定ファイルによってどのような間隔で，どの情報を取得するのかを指定する．

4.1.4 ユーザインターフェース部

WINDSではサーバとのインターフェースは，ユーザプロセスの中に組み込まれる．つまり，ライブラリの形式で提供される．データベース管理部との通信は，RPCプロトコルによって通信を行う．

4.2 パケットの構造

通信におけるデータのやりとりは，データをパケット化して行う．このパケットには，以下にあげる情報を所有している．

1. 質問の種類
2. 質問の内容
3. アクセス制御用フラグ
4. 返答
5. キャッシュのためのデータの生存時間

このような情報をパケットとして通信に用いるために図6に示すパケット構造をとるようにする．

パケット内に写真データなどの，大きなデータが存在するときには，パケットのそのデータの部分に対し，データ圧縮を行って通信をする．パケットの通信量は，なるべく少ない方がよいのでこうするべきである．

4.3 通信方式

複雑なプロトコルを用いて通信を行うよりも，シンプルな通信方式を利用することにより通信方式によるプロトコルのオーバーヘッドを少くすることができる．また，通信のセキュリティについても，独自のセキュリティシステムを開発実装するよりも既存のセキュリティを使うことによってより完成度の高いシステムを構築することができる．容易に利用できる通信手段として，RPC(Remote Procedure Call)がある．このシステムは，現在多くのハードウエアに実装されており容易に利用することができる．また，このRPCには，セキュリティ対策が施されたSecure RPCがある．このSecure RPCを用いることによって容易にセキュリティ対策が施された通信を行うことができる．

そこで，開発するシステムの通信部分は全てSecure RPCを使うことによって行うことにする．このSecure RPCを用いることにより，開発するシステムは，セキュリティ対策が施された通信が容易に可能となる．また，通信におけるセキュリティ対策を施さなくても良いという点や，プログラミングを行う上で通信のためのプログラムが容易になるという点で，構造が簡素となる．

4.4 データベースのミラーリング

データベースの信頼性を上げるためには，サーバが停止してしまう事に対して何か対処しなければならない．このためには，データベースを管理するサーバのプロセスを複数違うハードウエアで起動すべきである．この補助的なサーバをセカンダリサーバと称して起動し，データベースの複製(ミラーリング)を作成する．

この時に問題になるのはデータは同じものを管理しているので，データをそれぞれ転送し合わなければ，データベースに不一致がおきる危険性を帯びている．このため，ダーティビットの手法を使ってデータベースのマージを行ない，各々のデータベースの不一致を防ぐ．

4.5 インターフェースの提供

WINDSに対するインターフェースは，ユーザに対して分かり易くて利用しやすくなければならない．このためには，インターフェースを全てライブラリの形で提供する．全ての提供関数は，ユーザーには通信を意識しなくてもよい

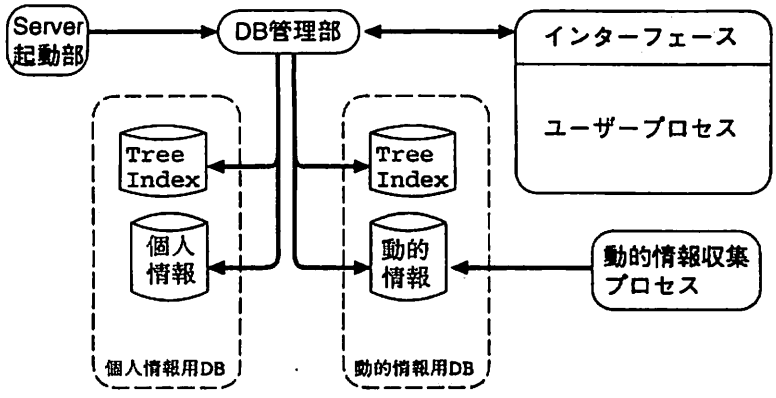

図 5: システム構造

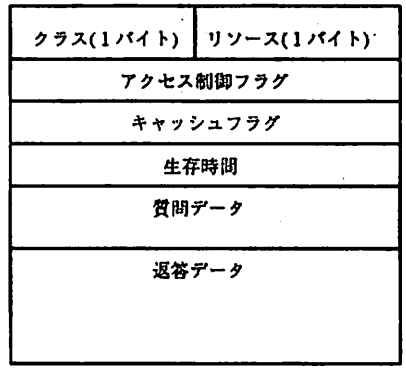

図 6: パケット構造

構造となっていて，その利用は容易にできる．

4.6 その他

情報を日本語のみとするよりも，マルチリンガルにした方が多くの利用が期待できる．そのため，流すデータのタイプとして言語のタイプを指定できる様にする．

5 実行例

図 7は，このシステムを立ち上げた後 xwinds という X-Window 用の WINDS クライアントプログラムを起動しその後，suguru というボタンを押したものである．図中の木構造は，名前空間を表している．この場合，WINDS の管理グループは 1 つであるが，サーバは，九州工業大学の計算機に 1 つ，大阪大学の計算機に 1 つおいてある．九州工業大学の計算機では，kyutech, ISCcenter の名前空間を管理し，大阪大学の計算機では Genesis-Project の名前空間を管理している． ボタンを押した後の表示は，実際にデータベースに登録されている情報を表示しているものである．

6 まとめ

今回開発したシステムを導入することにより分散環境での個人情報や計算機の動的情報について取得できるようになる．つまり，分散環境での管理者がマシンや個人を管理するための情

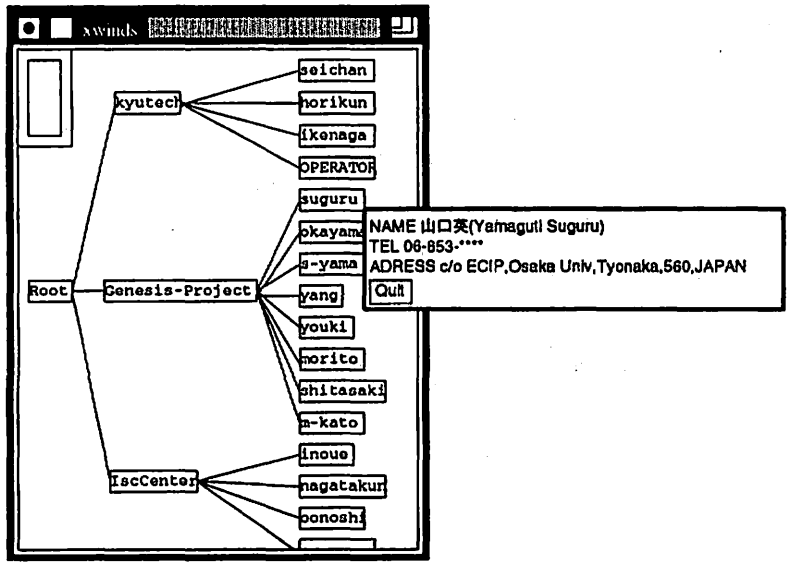

図 7: 実行例

報を提供することができる．このシステムは，管理面での大幅な労力削減と，さほど高度な技術がなくても管理を行うことができるようになる．

参考文献

[1] Mockapetris,P. Domain Names-Concepts and Facilities,RFC1034,1987,

[2] Mockapetris,P. Domain Names-Implementation and Specification, RFC1035, 1987.

[3] Stephen,P.Dyper. The Hesiod Name Server,Project Athena Technical Plan. MIT,1988

[4] CCITT. Infomation Processing System - Open Systems Interconnection - The Directory,International Standard 9594-1,1988

[5] Marshall T. Rose,The Simple Book: An Introduction to Management of TCP/IP-based Inrternets,Prentice-Hall, Inc, 1991.