ディレクトリサービス概説、および今後の展望について

技術本部 技術研究室 新谷敏之

1．ディレクトリサービスの定義

ディレクトリサービスは、人、コンピュータ、ネットワークサービスといった、いわゆるネットワークリソースに関する情報提供サービスとして定義される。ディレクトリサービスの標準であるX.500では

・通信手段（電話や電子メールなど）を効率化するための情報を提供すること
・単独の通信手段でなく複数の通信手段に対して汎用性を持つこと
・世界規模のサービスであること

をその定義の中で強調しているが、最近の傾向としては、通信手段にこだわらずあらゆるリソースに関する情報リポジトリを指向する、世界規模ということにこだわることなく、企業内で利用できる柔軟なアーキテクチャを採用するなど、現実的な解として採用しやすい環境が整いつつある。

本稿では、現実の解となりつつあるディレクトリサービスについて、前半でそのアーキテクチャに関する基本的な考え方を、後半でその応用モデルを解説し、ディレクトリサービスの有用性と今後の可能性について考えていく。

2．ディレクトリサービスのアーキテクチャ

ディレクトリサービスを電話帳、104番号案内、DNS（Domain Name System）といったものに例えている解説書や雑誌をしばしば見かけるが、それよりは、各種の情報を格納し、要求に応じてそれを提供するという点に注目し、ディレクトリサービスを一種のデータベースサービスとして捉えたほうがそのアーキテクチャを理解しやすい。ディレクトリサービスの特徴を一般的なデータベースサービスと対比させたものを表1にまとめる。

2.1 構成要素

ディレクトリサービスに格納される情報の基本単位はエントリと呼ばれる。エントリはそれぞれ現実世界のエンティティを抽象化するものであり、オブジェクトクラスと呼ばれる型によって型付けされ[*1]、1つ以上の属性から構成される。

たとえば現実世界の“新谷敏之”という人物は、person[*2]というオブジェクトクラスで型付けされたエントリによって抽象化され、そのエントリはcommonName属性に“新

表1 ディレクトリサービスの特徴

	ディレクトリサービス	データベースサービス
情報の基本単位	エントリ	レコード
情報の格納構造	ツリー構造	表構造
情報へのアクセスメソッド	DAP、LDAP（プロトコル）	SQL（言語）
アーキテクチャ	クライアント／サーバー	クライアント／サーバー
分散メカニズム	Chaining、Referral、Shadowing	レプリケーション、2相コミットなど
重視される性能要件	読み出し性能	トランザクション性能

＊1）「型付けされる」とは、持たなければならない必須属性、持つことが許される任意属性が決まるということである。

谷 敏 之 ”、telephoneNumber 属 性 に “ ＋81－３－1234－5678 ”、mail 属性に “ shin@cac.co.jp ”、という値を持つといった具合になる。オブジェクト指向におけるクラス、オブジェクト（インスタンス）、属性の関係とほぼ同じと考えてよいだろう。

なお、ディレクトリサービスの相互運用を図るという目的で、標準的な属性は X.520、標準的なオブジェクトクラスは X.521で既に定義されている。いくつか例を示す。

・X.520で定義される標準的な属性の例

一般名（commonName、cn と省略される）
姓（surname、sn と省略される）
電話番号（telephoneNumber）
ユーザーパスワード（userPassword）
ユーザー証明書（userCertificate）

・X.521で定義される標準的なオブジェクトクラスの例

組織（organization、o と省略される）
組織単位（organizationalUnit、ou と省略される）
人（person）
組織人（organizationalPerson）
証明機関（certificationAuthority）

2.2 情報モデル

ディレクトリサービスの特徴の１つが、エントリをツリー状に配置するという点である。各エントリは、ディレクトリツリーのどこかのノード、もしくはリーフとして一意に配置される。この配置関係を示すのが DN（Distinguished Name）と呼ばれる特殊な属性である。DN はディレクトリツリーのルートから目的のエントリまでを右から左に順に並べたものであり、ちょうどファイルシステムにおける絶対パス名のような役割を果たす。

ディレクトリサービスを導入する際、ディレクトリツリーをどのように設計するかが重要なポイントである。X.500の制定当時[*3]は、組織構造をそのままディレクトリツリーに反映させるような設計が一般的であったようだが、最近の傾向としては組織構造の変更に伴うディレクトリツリーの変更を嫌い、リソース別にサブツリーを形成するといった設計も多いようである。もちろんいずれも長短があり、前者は、組織そのものと組織の権限構造を比較的素直に表現できる反面、組織構造の変更によってディレクトリツリーの変更を余儀なくされる。後者は組織構造の変更の影響を受けにくいが、ディレクトリに格納される情報に対するアクセスコントロールが難しくなる可能性が指摘されている。組織との関連においてエントリをどう配置するか、また、アクセスコントロールをどう実現するかがディレクトリツリーを設計する際のポイントになる。

2.3 機能モデル

ディレクトリサービスは典型的なクライアント／サーバー・アーキテクチャを採用しており、１台のサーバーと１台のクライアントがあれば最小構成の環境が成立する。なお、特にサーバーは DSA（Directory System Agent）、クライアントは DUA（Directory User Agent）と呼ばれる（図１）。

DUA は DSA に対して情報の提供を要求し、DSA は自分の所有するデータベースを照合し、適切と思われる情報を DUA に結果として返す。この要求と結果のやりとりに用いられるプロトコルは X.500において、DAP（Directory Access Protocol）と呼ばれ、次の12のオペレーションか

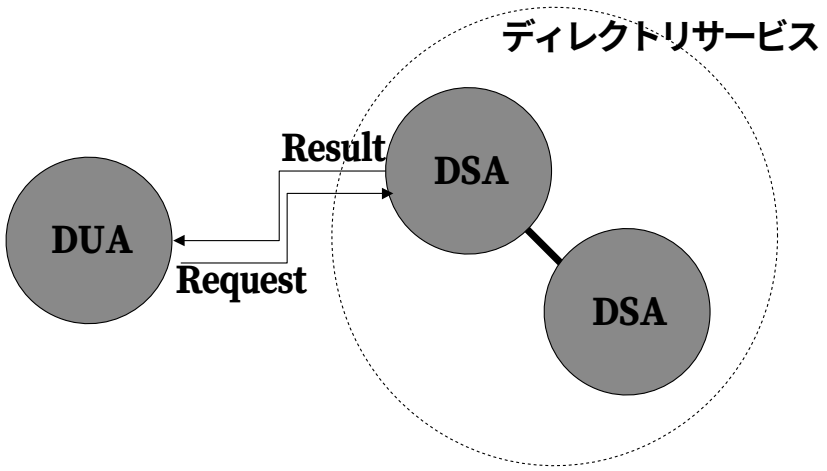

図１　ディレクトリサービスの基本アーキテクチャ

＊２）X.500の慣例として属性の名前やオブジェクトクラスの名前は原則小文字、２つ以上の単語を連結して名前とする場合は、区切りとして２番目以降の単語の先頭文字のみ大文字とする。

＊３）第１版が1988年、第２版が1993年に勧告されている。

ら定義されている。

bind .. DSA へのログイン
unbind .. DSA からのログアウト
read .. DN を指定した１エントリのみの読み出し
compare .. 属性値の比較
abandon .. 先行オペレーションのキャンセル
list .. 注目するエントリの直接下位エントリの一覧の取得
search .. 注目するエントリの下位ツリーから条件に合致するエントリを検索
addEntry .. エントリの追加
removeEntry .. エントリの削除
modifyEntry .. エントリの一般属性の変更
modifyDN .. DN の変更、つまりツリー構造の変更
modifyRDN .. RDN（Relative DN、DN の最左要素）の変更

DAP は OSI のアプリケーション層のプロトコルであるが、実装要件が重いため、これを TCP／IP 上に直接実装するようなプロトコルが考案された。これが LDAP(Light-weight DAP）であり、名前のとおり、実装コストが低く、read オペレーションと list オペレーションが省略されるなどプロトコル自身も若干簡略化されている。現時点におけるディレクトリアクセスの標準プロトコルである。

2.4 分散モデル

DUA から DSA へのアクセスコストを下げるため、あるいは管理を分散させるためなどの理由で、DSA を複数化し複数の DSA によって論理的に１つのディレクトリツリーを構築、保守するための仕組みがディレクトリの分散機能である。分散機能としては、

・Chaining
・Referral
・Shadowing

の３つがある（図２）。

Chaining は、DUA からのリクエストに対して DSA が答えられない場合、そのリクエストに答えることができる別の DSA にリクエストをフォワードするというものである。結果は、リクエストとは逆のルートを通り、DUA に返される。DSA－DSA 間のメッセージ量が多くなる可能性はあるが、DUA 側で DSA が分散していることを考慮した特別な実装を必要としない、かつ、データを重複して格納する必要がないといった優位性がある。

Referral では DSA が答えられない場合、リクエストに答えることができる DSA のアドレスを DUA に返し[*4]、そのアドレスに対して DUA が再度リクエストを発行し直す。データを重複して格納する必要がないものの、DUA 側に Referral を処理できるような実装が必要になる。

Shadowing は、DSA 間で事前にディレクトリツリーの一部、もしくは全部を複製しあうという手法である。データベースを分散させる際に用いられるレプリケーション手法と同じと考えてよい。DUA に特別な実装は必要なく、DUA－DSA 間のアクセスコストも小さいが、データを複製することになるため、整合性を維持するためのコストが発生する。また、大量データの初期同期にコストがかかる可能性がある。

それぞれの長短は表２で示すとおりである。コスト的な

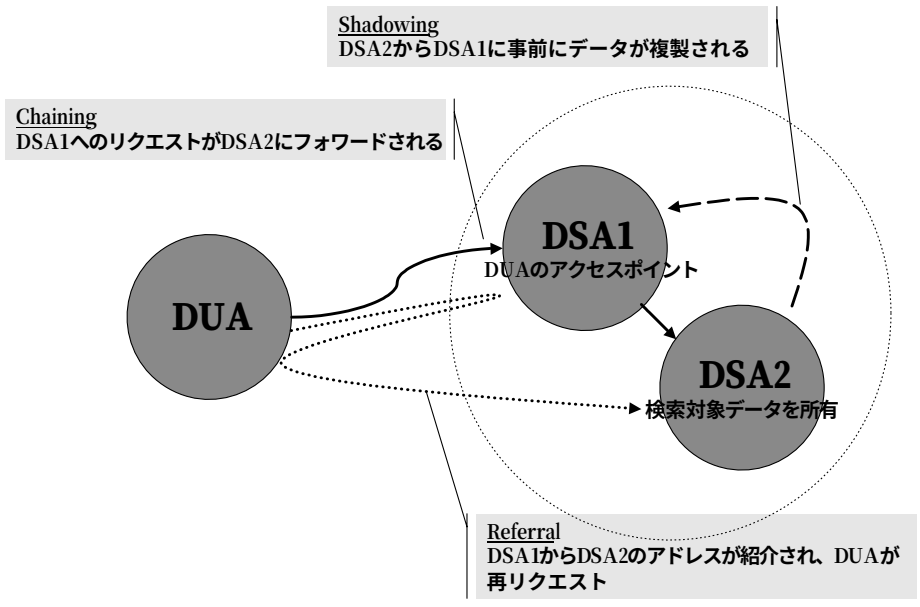

図２　ディレクトリサービスの分散機構

＊４）ldap：//で始まる URL が返される。

表2　各分散機構の比較

	Chaining	Referral	Shadowing
アクセスコスト	中	大	小
整合性維持コスト	小	小	大
クライアント実装コスト	小	大	小
設定コスト	小	小	中

面では Chaining が理想であるが、Chaining を完全な形で実装した製品が少ないのも事実である。現時点で、どの分散機能を採用するかは製品仕様に大きく影響されることになるが、最終的にはアクセスコストや管理の複雑さなどを最小化しながら、かつ、バランスをとることがポイントとなる。

3．ディレクトリサービスの応用モデル

ディレクトリサービスの最も典型的な利用方法は、ディレクトリサービスをアドレス帳として利用する方法である。すでにいくつかの電子メールソフト[*5]が対応しているが、アドレス帳を自分自身で保守するのでなく、公開された共有のアドレス帳として、ディレクトリサービスを利用することができれば、エンドユーザーをアドレス帳保守の煩わしさから解放することができ、電子メールソフトの乗り換えも容易になるなど便利である。

このほかにも、主要ベンダーからディレクトリサービスの応用モデルが提案されているので、興味深いものをいくつか紹介する。

3.1　Mission Control

企業内に存在する多くのアプリケーションは、ユーザーやグループといったコンセプトを持ち、ユーザーを認証したり、ユーザーに応じて情報へのアクセスをコントロールしたりする機能を持つ。しかし、現状の企業システムでは、アプリケーションごとに独立してユーザーやグループが管理されていることが多く、メンテナンスの増大といったコスト的な問題を発生させているばかりか、情報の冗長性によるセキュリティ上の脅威も増大させてしまっている。

Netscape はこのような状況を Application Islands[*6]と表現し、その解決策として、Mission Control と呼ばれるコンセプトとそれを具体化する６つの製品を提供している[*7]。このコンセプトはユーザー、グループ、パスワード、証明書、プロファイル情報、アクセスコントロール情報などアプリケーション間で共有されるべき情報をディレクトリサービスに集中的に格納し、それをすべてのアプリケーションで共有しようとするものである。Netscape では、これによって企業情報システムを構成するアプリケーション全般にアカウント管理の統一的な枠組みが提供され、同時に、メンテナンスおよびアプリケーション開発のコスト低減が可能になるとしている。

3.2　Service Publication

Microsoft は、Windows2000で Active Directory と呼ばれるディレクトリサービスの機能を提供する予定である。Service Publication は Active Directory の利用方法の１つであり、ネットワークサービスのロケーション情報をアプリケーションに動的に取得させることを可能にしている。

たとえば、現在のデータベース・アプリケーションの多くは、アクセスするデータベースの名前をローカルレジストリや INI ファイル中に格納しているケースが多い。データベースが利用不可能になった場合、システム管理者は、代替データベースのアドレスをユーザーに通知し、アプリケーションの構成情報を変更するようユーザーに指示を与えなければならない。これはユーザー数が多い場合、現実的な方法とはいえない。一方、スタンバイ状態にしておいた同じ構成のデータベースを同じ名前、同じ IP アドレスに変更したうえで再起動させ、クライアント側への影響を回避するという方法もあるが、トリッキーな方法であることは否定できない。

これに対し、Service Publication 対応のアプリケーションは、その初期化時にディレクトリサービスから、自分がアクセスするべきデータベース名を取得する。このため、

＊５）Netscape Messenger の Address Book 機能、Microsoft Outlook Express のアドレス帳機能など。

＊６）.. each application forms an independent island of administration, that must be separately, and often manually managed by an organization. These islands of administration create a huge administration burden for large companies...(Netscape Mission Control White Paper より抜粋)

＊７）Directory Server、Certificate Server、Admin Server、Mission Control Desktop、Java Console & Commands、Directory SDK for Java／C の６製品。

データベースが利用不可能になった場合、システム管理者はディレクトリサービスの格納情報を変更し、アプリケーションの再起動もしくは再初期化をユーザーに指示するだけでよい。

3.3 DEN（Directory Enabled Network）

DENは1997年にCiscoとMicrosoftが共同で発表したコンセプトであり、現在は仕様の確定がDMTF（Desktop Management Task Force）によって行われている。DENでは、ネットワーク機器やネットワークサービスの構成情報がディレクトリサービスに一元的に格納されると同時に、各機器やサービスの構成をディレクトリサービスを通じて動的に変更することができる。

たとえばCiscoが1998年のNetwork+Interopで行ったケーブルTV会社のデモでは、エンドユーザーがケーブルTV会社のWebサーバーに接続し、ビデオ・オン・デマンドのサービスを要求する。ケーブルTV会社のWebサーバーのバックエンドでは、ディレクトリサービスに格納された機器の構成情報をベースに帯域確保プランが作成され、最終的にはディレクトリサービスを介して、各機器に帯域確保が通知される。

Cisco、Microsoftのほかには、ネットワーク機器ベンダーとしてはBay Networks、Lucent Technology、ディレクトリ製品ベンダーとしてはNetscape、NovellがDEN対応製品の開発意向を表明している。

3.4 電子証明書の格納と公開

もともと、ディレクトリサービスでは電子証明書をbindオペレーションの１オプションとして利用していた。これは、DUAがDSAにログインする際、ユーザー名とパスワードではなく、より厳密な認証を可能にする電子証明書を提示するという方法である。その後ディレクトリサービスは、電子証明書を利用するだけではなく、電子証明書の体系的な格納と配付の機能を提供するものとしてその役割を変化させている。関連する標準としては、電子証明書の標準的なフォーマットを定めたX.509、電子証明書をディレクトリサービスに格納するために必要な属性を定めたX.520がある。

４．ディレクトリサービスの今後

システム・インフラストラクチャとのかかわりの中で、ディレクトリサービスの２つの機能性が、今後、より重要性を増していくのではないかと考える。

１つは企業情報システムを利用するため、もしくは管理するためのナビゲータとしての機能である。Mission ControlやService Publicationが示唆するように、ディレクトリサービスはあらゆるネットワークリソースの構成情報を格納するためのリポジトリとなるだろう。このとき、ディレクトリサービスは、エンドユーザーにとっては公開アドレス帳として、アプリケーション開発者にとってはアプリケーション構成情報の一元的なリポジトリとして、セキュリティ管理者にとってはアカウントデータベースとして、そして、ネットワーク管理者にとってはネットワーク機器データベースとして映るだろう。つまり、ディレクトリサービスこそが企業情報システム、言い換えればネットワークへの入口である。ディレクトリサービスを参照することで、企業情報システムを把握することができ、ディレクトリサービスを介することで、あらゆるネットワークリソースとコミュニケーションできるようなインフラストラクチャが、今後、構築可能になるのではないかと考えられる。

もう１つは、次世代セキュリティ・インフラストラクチャの主要構成要素としての機能である。現行のセキュリティは、ユーザー名とパスワードによる本人認証をベースにしているが、パスワードは失念そして漏洩しやすく、また、利用者のモラルに強く依存するといった面でその脆弱さが表面化しつつある。これに替わるより安全で確実なセキュリティ技術として、公開鍵暗号技術をベースにした本人認証、アクセスコントロール、暗号・復号システムなどの研究と製品化が進んでいる。公開鍵暗号技術をベースにしたインフラストラクチャは一般にPKI（Public Key Infrastructure）と呼ばれるが、ディレクトリサービスはこの中で電子証明書の格納と公開という中心的な役割を果たすものとして注目される。

５．おわりに

ディレクトリサービスはとりたてて新しい考え方ではない。従来のアプリケーションでも、ユーザー管理、グループ管理といったディレクトリサービスに類する何らかの機能を有しており、標準であるX.500の制定も1988年と比較的古い。にもかかわらず、ディレクトリサービスが最近になって特に注目されるのはなぜだろうか？１つにはネットワークリソースが多様化し、それを効率的に利用したり管理したりするためのナビゲーション機能が必要になったためと思われる。もう１つはより安全で強固なセキュリティ・インフラストラクチャへの渇望であろう。これらはインターネットといったオープンな場でも、企業情報システムといったクローズな場においても同様なことがいえる。

今後、インターネットや企業情報システムにおいてディレクトリサービスが前節で述べたような役割を果たすべく成長していくのかどうかを、その技術動向および製品動向とともに注目していきたい。