3N−1

# ディレクトリシステムのＤＩＢ実装方式（1）
## ＤＩＢファイルのしくみ

増尾　洋† 池ノ谷　和幸†† 橋本　浩一††
†東芝ソフトウェアエンジニアリング（株）
††（株）東芝　情報処理・機器技術研究所

### 1．はじめに

ＩＳＯ／ＣＣＩＴＴで標準化が進められているＯＳＩ応用層のサービス要素の1つであるディレクトリ[1]は、通信に必要となる情報をディレクトリ情報としてデータベース等に保持，管理し、必要とする利用者が容易にこれらの情報を獲得する機能だけではなく、電話番号やネットワーク構成等の変更に伴うディレクトリ情報の追加変更機能をも提供するものである。本報告はディレクトリの核となるディレクトリ情報ベース（ＤＩＢ:Directory Information Base）の構築方法を提案する。

### 2．ディレクトリ情報ベース

ディレクトリが保持している情報の集合は、ＤＳＡ(Directory System Agent)が管理しているＤＩＢに蓄積されている。ＤＩＢに蓄積されている情報は、ディレクトリの操作によって読み出したり検索したりすることができる。属性と呼ばれるディレクトリの情報はエントリという属性の集合に格納される。エントリは、ＤＩＴ(Directory Information Tree)構造と呼ばれるトリー構造を形成し、トリーの根から葉に至る識別名と呼ばれる名前によって一意に識別できる。ただし基本標準等では利用者ーディレクトリ間の取り決めを規定しているだけでＤＩＢの実装についての詳細な規定はなく、作成は実装者に任されている。過去にＲＤＢや００ＤＢを使用したＤＩＢが報告されているが、事業所レベルの小規模なネットワークなどではコストの問題等から必ずしも汎用データベースがシステムに構築されていない場合も存在する。又、汎用データベースを組み込むとしてもディレクトリのシステムとしては膨大で高価なシステムになりかねない。そこで我々は比較的簡単なＵＮＩＸ[注1]ファイルを使用したＤＩＢを提案する。

### 3．ＤＩＢの構造

今回ＤＩＢは図1の様な3つのファイルより構成される。

- ＤＩＢに含まれるエントリや属性値の情報が入るＤＩＢファイル
- エントリが属するオブジェクトクラス間の上下関係を定義するＤＩＴファイル
- エントリに設定される属性の制約を定義するクラス属性ファイル

以上のファイルをＤＩＢハンドラ[2]と呼ばれるタスクが管理する。

### 3．1　ＤＩＢ分割について

ＤＩＢファイルはエントリや属性値の情報が設定されるファイルであり、ＤＩＢハンドラからディレクトリのオペレーションに従って検索や追加変更が行われる。又ＤＩＴファイルとクラス属性ファイルは、ディレクトリスキーマを定義するファイルである。ＤＩＴファイルではＤＩＴ構造定義が設定され、クラス属性ファイルではオブジェクトクラス定義，属性型定義，属性構文定義が情報として設定される。この2つのファイルは1度構築されると追加変更する事がほとんどない。

この事からＤＩＢハンドラから直接読み書きができるファイルと、読み出しだけ実行できればよいファイルの2種類に分類した事により、それぞれのファイルの大きさを縮小でき、特にファイルの読み書きを行うＤＩＢファイルでは読み書きの際の負荷が低減できる。

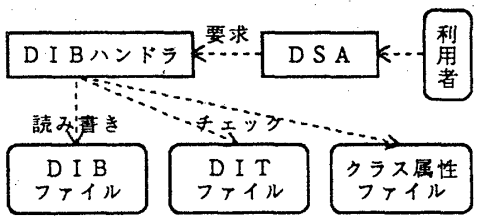


図1．ＤＩＢの構成

### 3．2　ＤＩＢファイルの構造

ＤＩＢファイルに設定されるエントリや属性の情報を以下のように規定した。

（1）ＤＩＢに設定される各エントリの種類やエントリ間のつながりを示す情報。
（2）各エントリに設定されている相対識別名や属性値など、情報の量が多く可変長なもの。

これらを同一の領域において管理するよりも、1つのファイル内の別々の領域において管理する方が望ましいと考えられる。従って、1つのファイルを以下の様に分割管理する。

- 固定領域：（1）の情報を設定する。
- 自由領域：（2）の情報を設定する。

又、自由領域は相対識別名や1つのエントリに複数存在する属性の情報などを設定する為、その全体量は莫大になると予想される。従って、ファイルアクセスの回数を極力軽減するなどの処置が必要になる。そこで、自由領域をある大きさごとのセグメントに更に分割することによって管理をすることとする。図2はＤＩＢファイルのファイル構成を示す。

ＤＩＢハンドラは、固定領域用のバッファと自由領域用のバッファをあらかじめ用意し、各セグメントを読み込んで処理をする。尚、固定領域は現在1つの大きさで済む程度のＤＩＢを考えており、ＤＩＢハンドラ実行開始時にバッファに読み込んでくるものとしているが、ＤＩＢを拡張したときの事を考えて固定領域のバッファ管理も設定している。自由領域に関しては、必要となった場合にその都度バッファに読み込んでくるものとする。

---

*Implementation of DIB for OSI Directory System (1)*
*Hiroshi Masuo[1],Kazuyuki Ikenoya[2], Koichi Hashimoto [2]*
*[1]Toshiba Software Engineering Corporation*
*[2] Information Systems Engineering Laboratory,Toshiba Corporation*

| 固定領域 | 自由領域１ | 自由領域２ | ・・ |
|---|---|---|---|

図２．ＤＩＢファイルの構成

固定領域にはエントリの情報を格納するエントリテーブルを設定する。エントリテーブルは、図３の様に他のエントリテーブルへのポインタ，同位エントリと呼ばれる同一のレベルの（直接下位）エントリへのポインタ，直接下位エントリへのポインタ，上位エントリへのポインタ及び自由領域に設定されている属性テーブルへのポインタと、そして別名エントリから指されている場合はその別名エントリへのポインタ等によって構成される。このような構造にした事により、検索回数の多いエントリテーブルは可変長に作成することなく固定領域で集中管理して、可変長なパラメタは自由領域に作成して空き領域の管理を行う事により無駄なテーブル領域がいらない等、資源の有効活用ができる。

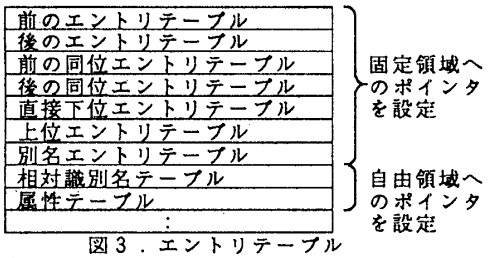


図３．エントリテーブル

固定領域はエントリの追加・削除が行われるごとに、又自由領域は属性や属性値の追加・削除が行われるごとに、それぞれ空き領域の管理が実行される。図４は固定領域のブロック図を示し、テーブル格納領域には使用済テーブル列と未使用テーブル列が設定される。エントリが追加されると未使用テーブル列にあるテーブルに新しいエントリが作成され、使用済テーブル列の最後にチェインされる。エントリが削除されると使用済テーブルにチェインされていたそのテーブルは未使用テーブルにチェインされる。

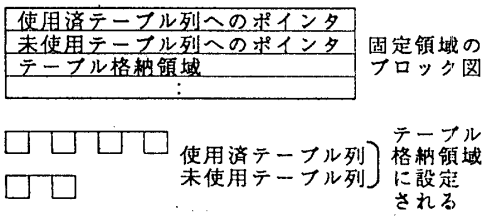


図４．固定領域の空き領域管理

## ３．３　エントリテーブルのリスト構造

固定領域においてエントリはＤＩＴ構造で表現されなければならない。従って、各エントリ間をチェインで結んでトリーを表現する。各エントリ間は図５の様な関係で結ばれており、あるエントリが保持する直接下位のエントリの情報は、直接下位エントリが複数あろうとも１つしか設定しない。そのエントリが葉のエントリの場合はＮＵＬＬが設定される。つまりエントリテーブルは自分の下位エントリに関する情報を、自分のテーブルの中に全て持つ必要がなく、また直接下位のエントリを獲得したい時はエントリテーブルに含まれる１つの直接下位エントリを調べ、その同位エントリを得ればよい。図５の上の図の様な木構造が存在し、エントリＡに存在する直接下位エントリのリスト（エントリＢ，Ｃ，Ｄ）を獲得したい時は、まず１つの直接下位エントリ（エントリＢ）を獲得し、そのエントリに同位エントリＣが存在するのでそのエントリＣを獲得し、更にエントリＣには同位エントリＤが存在するのでそのエントリＤを獲得すれば良い。こうする事でエントリＡが下位エントリを１つ獲得する度に再びエントリＡの下位エントリ獲得の処理に戻る事がなく、処理の簡略化と高速化がはかれる。

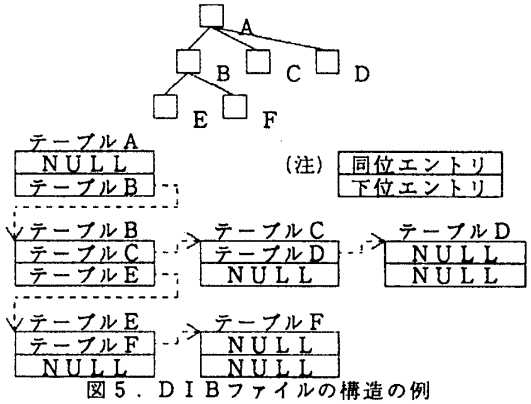


図５．ＤＩＢファイルの構造の例

## ３．４　テーブルの無制限設定

３．３で述べたリスト構造をとる事により、あるエントリが持つ事ができる下位エントリの数に制限を与える必要がない。この事は他のテーブルにも同様で、属性や属性値の個数に制限を与えないで済む。又それらの設定の為に無駄な領域を確保する必要がない。

## ４．あとがき

本稿ではＯＳＩディレクトリで使用するＤＩＢの構築方法について報告した。ＤＩＢへのアクセスに適したエントリのトリー構造や、属性などのデータ設定方法を考え、ＤＩＢファイルとして設計した。更に実エントリテーブルに別名エントリへのポインタテーブルを作成するなど、９２年版へ向けての対応にも一部考慮したＤＩＢファイルとなっている。ＵＮＩＸファイルを使用したＤＩＢを構築した事により、小規模なシステムでも使用できるディレクトリシステムが作成できた。現在、このＤＩＢファイルを用いたディレクトリシステムを試作中である。またこの他にオブジェクト指向型データベースを用いたＤＩＢも同時に作成している。今後はこの２つのＤＩＢを比較，検討し、更に効率の良いＤＩＢを作成していく予定である。

## 参考文献

[1] ISO/IEC 9594-1～8／CCITT X.500シリーズ
[2] 池ノ谷他，"ディレクトリシステムのDIB実装方式(2)DIBファイルの操作"，情報処理学会第46回全国大会，1993

[注1] UNIXはUNIX System Laboratries,Inc が開発しライセンスしています。