# SuperSQLを用いたLDAPデータの自動生成

4ZA−02

古思 望† 遠山 元道†

†慶應義塾大学 理工学部 情報工学科

## 1 はじめに

現在インターネットの爆発的な普及によりある組織にいる人々のメールや電話番号などをディレクトリ構造に蓄えることが多くなってきている。そのディレクトリサービスの中でも最近汎用的に利用されているのがLDAP(Lightweight Directory Access Protocol)[2]である。しかし、LDAPには関係データベース(以下RDB)が行える大量の更新やトランザクション機構、SQLを用いたjoinして欲しい情報を得るなどのリレーショナル処理がサポートされていない。また、LDAPを利用するためのAPI(Application Programming Interface)が開発されているが、どれもRDBからLDAPデータに変換してくれるものはない。そこで本研究では質問文の結果から様々な媒体への出力が可能なSuperSQL[1]を用いてRDBからLDAPデータを自動生成することを提案する。

## 2 LDAP

LDAPとはディレクトリサービスを汎用的に扱えるようにプロトコルを定義している。

### 2.1 情報モデル

LDAPはディレクトリ構造をとっていて、1つ1つのエントリー(ノード)は現実の世界に存在するオブジェクト概念(例えば人や組織など)であり、それらに関する名前、mailなどの情報保を持している。エントリーは名前、mailなどの情報を属性(atrribute)として持ち、各属性には1つ以上の値(value)がある。属性の中でobjectclassは、エントリーがどのような属性を持たなくてはいけないか、どのような属性を持つことができるか、とういことを表すための特別の属性名である。

Automatic generation of LDAP data using SuperSQL
KOSHI Nozomu†, TOYAMA Motomichi†
†Department of Information and Computer Science, Faculty of Science and Technology, Keio University.

### 2.2 ネーミングモデル

エントリーを1つ1つ識別するためにそれぞれのエントリーには識別名(DN)が付いてる。識別名はDNSと同じように階層構造のトップからツリーをたどるごとに後ろから表現していく。例えば「cn=ito,o=Koshi.com」の識別名はKoshi.comの下にいる伊藤を表す。識別名に用いられる属性名はそのエントリーの中にある属性ならどれでもよい。

### 2.3 LDIF

エントリーの集合を表すディレクトリ全体の情報をテキストで表現する記述方法をLDIF(LDAP Data Interchange Format)と言う。例えば、ある会社Koshi.comの支店であるKawasakiエントリーのLDIFは以下のようになる

```
dn:ou=Kawasaki,o=Koshi.com
objectclass:organizationalUnit
obectclass:top
ou:Kawasaki
```

## 3 SuperSQLの演算子と装飾情報

SuperSQL質問文では横、縦、深度の各次元の結合子を「,」,「!」,「%」で表現している。またインスタンスがある限り対応する次元方向に繰り返しをする反復子を「[]」で表す。また質問文の中で多彩な装飾指定を行う事ができ、これは質問文の属性の後に「@装飾情報」を記述することで実現される。LDAPにおいてそれらがどのように対応するかを説明する。

### 3.1 LDAPにおける演算子の対応

水平連結子「,」は1つのエントリーにある属性を続けてLDIFに表示させる。垂直連結子「!」は同じ親を持つエントリー同士を結合させる。深度連結子「%」は親子関係にあるエントリー同士を結合させ、

親の識別名を継承する。

### 3.2 LDAPにおける装飾情報

LDAPにおける装飾情報は4つある。1つ目はobjectclassでobjectclassの属性を指定する。2つ目はLDAPでは属性の名前が決められているので、attでRDBの属性名からLDAPの属性名に変換する。3つ目はエントリーがaliasオブジェクトの場合はatt-aliasでどこにaliasが張られているかを記述する。4つ目は親が記述されていないとき親の識別名が継承されないのでdnで親の識別名を指定する。次の章で質問文の例を見せる。

## 4 SuperSQLによるLDAPの実装

ある会社Koshi.comには、支店がありその支店には売場がある。そして売場で働ている従業員がいる。それらの情報が格納されているRDBからSuperSQLを用いてLDAPデータに変換するとき、質問文は以下のようになる。

```
GENERATE ldap [
  {c.name@{att=o}}
  @{objectclass=organization
            ,objectclass=top}%
  {{[{s.city@{att=ou}}
     @{objectclass=organizationalUnit
                ,objectclass=top}%
  [{d.name@{att=ou}}
    @{objectclass=organizationalUnit
               ,objectclass=top}%
  [{e.name@{att=aliasedObjectName
           ,att-alias=o=Koshi.com}
     }@{objectclass=alias}
  ]!]!]!}!
   [{e.name@{att=cn},e.name@{att=sn}
                ,e.mail@{att=mail}}
  @{objectclass=top,objectclass=person
    ,objectclass=inetorgperson
    ,objectclass=organizationalPerson}]!}
              ]!

FROM company c,store s,dept d,employee e
Where s.company_id=c.num and
     d.store_id=s.num and e.dept_id=d.num;
```

図1: 質問文

この結果であるLDIFは次のように出力される。

```
dn:o=Koshi.com
objectclass:organization
objectclass:top
o:Koshi.com

dn:ou=Kawasaki,o=Koshi.com
objectclass:organizationalUnit
objectclass:top
ou:Kawasaki

dn:ou=book,ou=Kawasaki,o=Koshi.com
objectclass:organizationalUnit
objectclass:top
ou:book
........
```

そしてこのディレクトリ構造は図2のようになる。

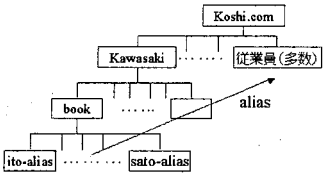


図2: ディレクトリ構造

## 5 結論・検討

SuperSQLを利用してRDBから一貫性のあるLDAPデータができることが実現された。これによりRDB側ですべての情報を管理して更新などを行い、それらの情報を本システムを用いてLDAPデータに変換しLDAPサーバーに格納することができる。LDAPの弱点である更新などをRDB側に任せることができ効率的である。例えばLDAPにおいてあるエントリーが変更されたときそのエントリーがあるsubtreeから削除して、別のsubtreeに付け加えなくてはいけないがRDBではそのエントリーの属性を変えるだけでよいのである。現在のところRDBにあるすべてのデータの一括生成しかできないので今後の課題としてはRDB側で更新されたデータのみをLDAPサーバー側で更新する差分更新の実現である。また図1の通りクエリーが複雑となるのでユーザーにとってわかりやすいGUIの作成が必要である。

## 参考文献

[1] Motomichi Toyama, "SupreSQL: An Extended SQL for Database Publishing and Presentation," *Preceedings of ACM SIGMOD '98 International Conference on Management of Data*, pp. 584-586, 1998

[2] Sihem Amer-Yahia,H.V.Jagadish,Laks V.S. Lakshmanan and Divesh Srivastava "On Bounding-Schemas for LDAP Directories" *EDBT2000*, pp.287-301, 2000