

## XPath を用いた LDAP ツリーの検索

山本 剛士 塚本 享治

東京工科大学大学院バイオ・情報メディア研究科メディアサイエンス専攻

## 1. はじめに

LDAP[1]とはディレクトリサービスにアクセスするためのプロトコルである。しかし、LDAP を利用するための「言語」が存在しないために、今まではプログラミング言語に依存する部分が大きく、LDAP を利用するためのコストが高かった。

そこで、本研究では LDAP のデータ構造が XML の木構造と類似していることに着目し、XPath[2]による LDAP ツリー(以下 DIT)の検索言語を実現した。

## 2. XPath による DIT 検索の実現

## 2.1 基本方針

XPath に特別な拡張を施さず、XML ドキュメントと DIT の違いを意識せず利用できるようにするという方針を立てた。

これは、XPath の利便性の高さを崩さない、差分習得のための手間をかけさせないということだけでなく、XSLT[3]や XUpdate[4]などの XML 関連技術の利用を念頭においているからである。

XPath をそのまま利用するため、XML ドキュメント用に書いた XSLT などのコードは、ほとんど修正をすることなく DIT に適応することが可能となる。

## 2.2 構造間のマッピング

XPath は XML のために創出された言語である。よって DIT で XPath を利用するためには、DIT を XML とみなすこと、つまり DIT と XML の構造をマッピングする必要がある。

## 2.2.1 XML の構造

XML はひとつのルートから広がる木構造であり、各要素はタグで表される。各要素は、子要素を持つことが可能であり(木構造)、一般的に

タグの値は、そのタグのテキスト子要素、ないしは属性である。タグに対する属性および子要素の追加は任意であり、一般的には DTD などのスキーマ記述言語によって制限される。

## 2.2.2 DIT の構造

DIT は XML と同様に木構造であり、各要素は属性名=属性値という形となる。属性値は省略することができず、また、各サブツリーの同一のネストレベルの中では、ユニークな属性値を持つ要素(RDN)だけが、子要素を持つことができる。

## 2.2.3 要素のマッピング

LDAP の要素は、属性名=属性値という形で表される。"dc=jp"を"<dc>jp</dc>"のタグにマッピングすると、RDN が子要素を持っていた時に奇妙な形になってしまう。

```
<dc>
  jp #ルート要素 dc の属性値
  <dc>example</dc> #子要素
</dc>
```

そこで、LDAP の要素における属性値は、次のようにタグの属性として付与することにした。

```
<dc value="jp">
  <dc value="example" />
</dc>
```

タグ名に value という属性を追加しているが、dc という名に対して値が value である、という点からも直感的である。その結果

"dn: dc=example, dc=jp"は XPath では次のように表記することができる。

```
/dc[@value='jp']/dc[@value='example']
```

以上のことからわかるとおり、DIT は制限された XML ドキュメント、つまり XML の完全なサブセットとみなすことが出来る。

以下にマッピングの様子を図式化したものを示す。

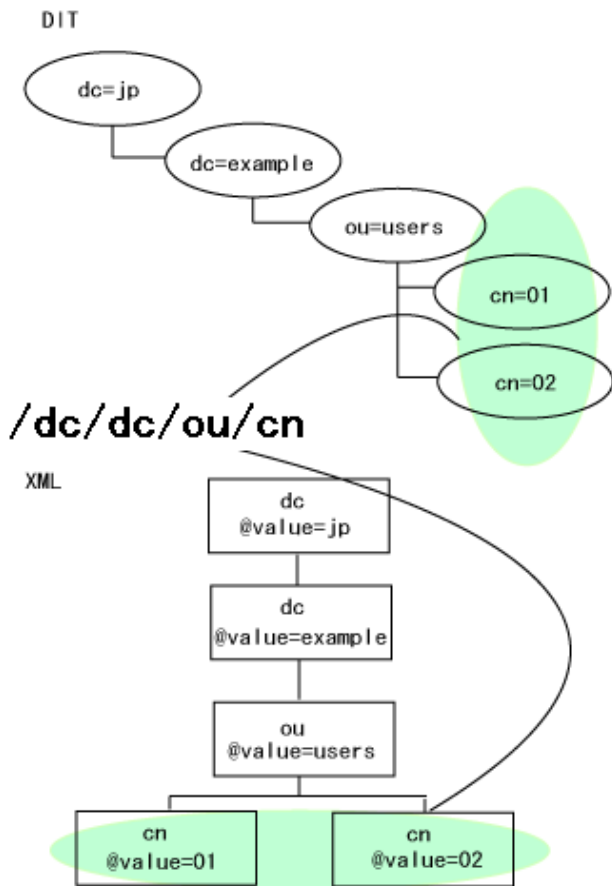


図 1 DITとXMLのマッピングの様子

### 2.3 機能間のマッピング

LDAP における検索は、ベースサフィックスと検索フィルタによって表される。ベースサフィックスには、特定の DN を指定することで、それより上位すべてのエントリを無視するようになる。(つまり特定の DN を示すことで、それ以下のサブツリーしか検索しなくなる)。検索フィルタは、ワイルドカードの利用、比較などによって、候補エントリを絞り込むものである。

XPath にベースサフィックスという概念は存在しないが、ベースサフィックスの機能を満たす記述をすることは可能である。

`dc=example, dc=jp`

というベースサフィックスは

`/dc[@value='jp']/dc[@value='example']`

という記述を行うことが可能である。

検索フィルタについても、XPath の比較の構文を用いて LDAP の検索フィルタはほとんどそのまま対応させることができる。ただし、LDAP の検索フィルタにおける「ほぼ等しい」という比較機能は XPath には存在しない。

この機能を実現するためには XPath の文法レ

ベルでの拡張が必要である。比較演算子が一つ増えるだけであるが、XML と完全に等価ではなく、互換性が無くなってしまふ。

逆に、XPath を拡張しないのであれば、LDAP が本来保持する検索機能の一部を削ることとなる。どちらにするかはトレードオフであるが、本研究では XML との互換性を重視して、XPath の拡張関数を作成してこの機能を実現した。

### 3. インプリメンテーションと評価実験

動作確認、評価のため、Java 言語を用いて OpenLDAP にアクセスする XPath ライブラリのインプリメンテーションを行った。

インプリメンテーションの結果、正しく動作させることができることが確認できた。

しかし、まだ残されている課題は多い。一つには、XPath と LDAP の検索機能の仕様の違いから十分な最適化が行えておらず、余計なオーバーヘッドが発生している点である。これは、XPath が、LDAP の検索機能よりはるかに多くの仕様を持っているためである。

また、複数のサイトにまたがる分散ディレクトリ構造を持った DIT に対しても十分な動作検証ができていない。検索だけを考慮するならばそれほど大きな問題にはなりえないが、XUpdate を利用したディレクトリの更新時などに、問題が発生する可能性がある。

これらの問題を解決するために、設計の見直しと、インプリメンテーションの改良を行っていく必要がある。

### 4. おわりに

XPath を用いた LDAP ツリーの検索が出来るようになったが、XPath を Java 言語からしか利用できないのでは、結局のところプログラミング言語に依存することになってしまう。この XPath における LDAP ツリーの検索を、XSLT などの XML 関連技術と組み合わせることで、LDAP はより簡単に扱えるものとなる。

#### 参考文献

- [1]RFC2251 " Lightweight Directory Access Protocol (v3)" <ftp://ftp.isi.edu/in-notes/rfc2251.txt> 1997
- [2]W3C, "XML Path Language(XPath) Version1.0" <http://www.w3.org/TR/xpath>, 1999
- [3]W3C "XSL Transformations(XSLT) Version1.0" <http://www.w3.org/TR/xslt>, 1999
- [4]XMLDB "XUpdate Working Draft" <http://xmldb-org.sourceforge.net/xupdate/xupdate-wd.html>, 2000