Simple Web Interface to LDAP Directories

J. Sodnik and S. Tomazic
University of Ljubljana, Faculty of Electrical Engineering
Trzaska 25, Ljubljana, Slovenia
jaka.sodnik@fe.uni-lj.si

Abstract— Lightweight Directory Access Protocol is a communication protocol which enables the access to online directories. As it is the case with the majority of free client software, the user is merely able to explore the directory, but can not change its content. The necessary software to administrate LDAP directory can be bought, but it is usually not compliant with the needs of users. Our interface module runs on ASP server and works as a converter between HTML and LDAP protocol. With the use of our model the user can access his directory, administrate it or change its content merely by using a standard internet browser. It was developed to simplify the use of LDAP protocol and the access to LDAP directory for various users.

Index Terms— Lightweight Directory Access Protocol, directory, web, interface.

1 Introduction

The information society and business infrastructure of today mostly dependon distributed computer systems and networks which serve as a platform for different applications. Network applications depend on interactions between computers which are part of Local Area Networks (LANs) or wide networks such as the Internet. Different types of information about the users, applications, data files, printers, etc., are usually stored in special databases called directories. Several different types of directories are available. In order to enable uniform data access to these directories, a special protocol, the so-called Lightweight Directory Access Protocol (LDAP), was developed.

LDAP is the standardized protocol enabling the user to access and manage directory data. It is optimized and highly adapted for reading and searching directory contents and less adapted for directory writing or modifying. LDAP is easy for implementation and highly efficient. It is based on client/server interactions. LDAP client connects to LDAP server and requests or sends specific data, depending on the current operation.

LDAP directory can be dealt with as a specific database, adapted to specific data. LDAP directory and classic databases, such as SQL or Oracle, do not differentiate much, until they are deployed in specific system and filled with data. The main differences, however, are the following:

- read to write ratio: directories are optimized for reading (writing is usually limited to administrators),
- directories do not support transactions,
- strict consistency in directories is not required,
- special directory access protocol (LDAP) is used in directories instead of simple query language (SQL) used in databases,
- distribution of data (one database and many physical servers in directories) , and
- high efficiency of directories (more operations per second).

LDAP directory and protocol can be described on the basis of four LDAP models which enable the compatibility for different LDAP versions. The following four models also enable personification and modification of specific directories to individual users [5], [6].

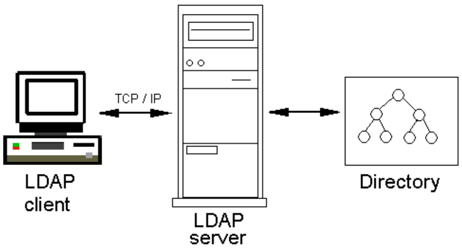


Fig. 1. LDAP server with directory

The basic LDAP functions are derived from x.500 standard, but they are much simplified. Many restrictions and rules typicial for x.500 directories are abandoned. The four described models are:

- LDAP information model
- LDAP naming model
- LDAP functional model
- LDAP security model

LDAP models serve as basis for directory developers and administrators.

1.1 LDAP information model

LDAP information model specifies the data type or the basic information unit, which can be stored in LDAP directory and can be operated with. The information model describes the main directory components or building blocks.

The basic information unit in the directory is an entry, which describes all the important characteristic of specific object: person, building, hardware, software, etc. The entry consists of a group of attributes; each attribute describes a specific detail of the object. Further, each attribute has its type and one or many values. The type defines the sort of information contained in value fields.

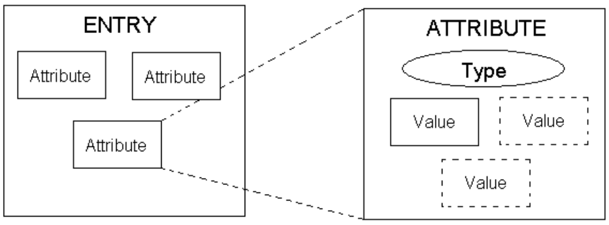


Fig. 2. Relations between entries, attributes, types and values

1.2 LDAP naming model

The LDAP naming model defines the organization of data in a specific directory. It describes the structure types which can be used to build customized directory entries and structures. The naming model also defines the relations between different parts of directory contents. The directory content is usually organized as a tree (Directory Information Tree – DIT). Each entry has its unique name, the so-called Distinguished name (DN), by which it can be addressed or accessed directly. DN consists of a sequence of the so-called Relative Distinguished Names (RDNs) up to the root of the tree. RDN is the left or the beginning part of DN. It is assigned to a directory entry. The combination of all RDNs composes a unique DN for each single entry. RDN is also one of attributes in the entry. When addressing an entry, the abbreviations of attribute names are used. Some of the most common attributes are shown in Tab. 1.

TABLE 1
LIST OF SOME COMMON ATTRIBUTES

Attribute type	Abbreviation
CommonName	CN
OrganizationalUnit-Name	OU
CountryName	C
UserId	UID

The following figure shows an example of a LDAP tree.

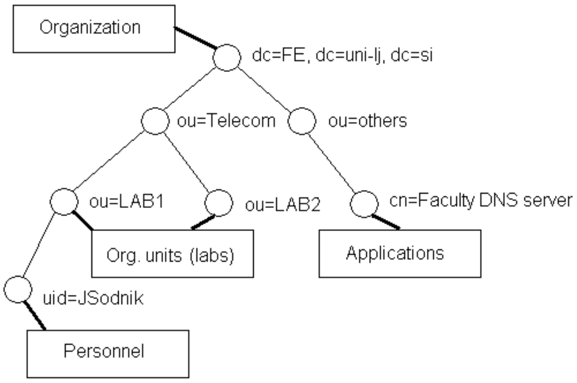


Fig. 3. LDAP tree structure

The directory address or DN for Mr. Sodnik would be:
uid=JSodnik, ou=LAB1, ou=Telecom, dc=FE, dc=uni-lj, dc=si

1.3 LDAP functional model

The LDAP functional model defines the directory protocol operations which are used to access or modify the directory content. It contains nine basic operations which can be divided into three main categories:

- search (search, compare)
- modify (add, remove, modify, modify DN)
- authenticate (bind, unbind, cancel)

With different combinations of these basic operations, a LDAP client can perform advanced request on the server.

1.4 LDAP security model

LDAP is a connection-oriented protocol, which is an important fact for the security model. A LDAP client creates the connection to server and hands in its request. At the time of the binding process, the authentication process is performed. Based on the authentication, the corresponding server assigns privileges and restrictions for directory browsing and modifying. Simple authentication and Security Layer (SASL) or Secure Socket Layer (SSL) can be used to ensure greater security. Some developers came up with the idea of enabling LDAP administration via Web interface [10]. In such a way, the administration is not limited to specific computers and is thus available anywhere [9].

In our laboratory, we developed an application which enables simple LDAP directory administrations (reading, writing, modifying, etc). We use a LDAP directory for mail addresses of users and their x.509 certificates which are used for mail encryption. The application is a program module which enables access to LDAP directory and its administration via Internet browser. It consists of three main modules, working on different layers.

2 LDAP DIRECTORY WEB ADMINISTRATOR

An important issue of the LDAP directory use is its administration. The market offers many applications which enable the administration of LDAP directories. The majority of the mail clients, organizers and other similar programs can read from LDAP directories directly, but can usually not modify them. Administration software is usu-

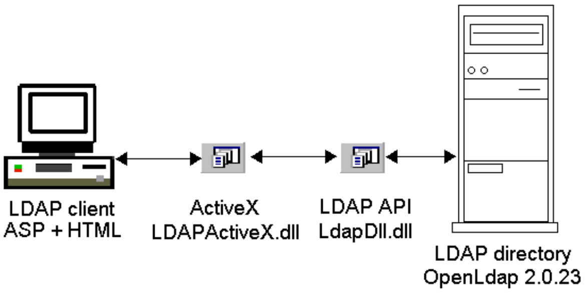


Fig. 4. Main application modules

ally not free, except with some exceptions [8], or it has limited functionality. Usually administration software has to be installed to all administration computers.

2.1 OpenLdap 2.0.23 server implementation

OpenLdap 2.0.23 is a version of the 3 LDAP server which can be downloaded for free. The server software is made primarily for Linux or Unix operating system, but can be modified to operate in Windows environment [FiveSight]. The source code written in C++ programming language has to be modified for the Windows system and then compiled. For successful compilation, the source code of data Sleepycat's Berkly DB model has to be provided.

After the successful installation of LDAP server, it can be managed through special configuration files with basic features and characteristics. The most important configuration file slapd.conf (Fig. 5) contais the following parameters:

- database: database type used as the basis for the directory tree
- suffix: the root of the directory tree
- rootdn: administrator of the directory
- rootpw: administration password
- etc.

```
#######################################################
# ldbm database definitions
######################################################


database ldbm
suffix          "dc=uni-lj,dc=si"
rootdn          "cn=Admin,dc=fe,dc=uni-lj,dc=si"
```

Fig. 5 Some basic database definitions (slapd.conf file)

LDAP server usually runs as service (in the background).

2.2 LDAPdll module

OpenLdap source code also includes the Application Programming Interface (API), which enables the development of arbitrary LDAP applications. API set consists of many basic functions which can be used to work with LDAP directory. The user can:

- add data to directory
- search the directory
- delete specific entries
- modify specific entries
- etc.

The use of API functions requires a good knowledge of C++ programming language [Prtenjak] and also a LDAP protocol. Our idea was to develop a universal interface which converts complicated protocol requests to a few simple basic operations.

LDAPDll module is a Dynamic-Link Librry (DLL) file. DLLs are typical for Windows operating systems. They consist of public functions and their parameters, which can be used and called in many different applications. They are usually used in applications developed in more advanced programming languages to access some files at a lower level.

LDAPDll includes three public functions: Search, Add and Delete:

- the Search function returns a data set consisting of specific entries and their parameters
- the Add and Delete functions return only the descriptive information about a successful operation or possible error.

2.3 LDAPActiveX module

LDAPActiveX is the second module of our project, developed in MS Visual Basic 6.0 programming language. This component works on a higher level and uses LDAPDll component to access LDAP directory. It can be accessed and used from HTML or ASP pages [Powers]. ActiveX is Microsoft technology which enables the development of interactive and powerful ASP pages. ActiveX components also have "dll" extensions and contain definitions of objects and classes, together with their properties and methods. They can be developed and used in many different programming languages.

The ActiveX component is usually used as a sort of a "black box", knowing only its properties and methods. In our case, LDAPActiveX module's properties describe the server's IP address, port number, username, password, search filters, etc. The methods enable the work with LDAP directory, using public functions in LDAPDll module. LDAPActiveX module has the following mehods:

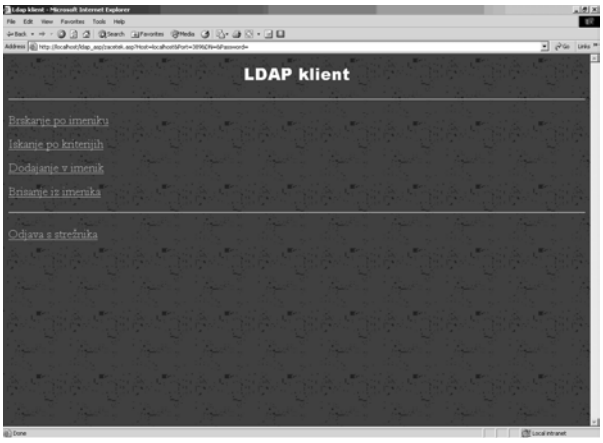
- LDAP_Search (search, based on specified search filter)
- LDAP_Result (details about returned entires)
- LDAP_Add (add entries into directory)
- LDAP_Delete (delete entries from directory)

LDAPActiveX component is not build into ASP application directly, but is installed on the WEB server.

2.4 LDAP web client

The LDAP WEB client is an ASP internet application which serves as a user interface for LDAP directory access. It is the third and "highest" module of our project. The important part is its functionality for simple directory access and modification which are enabled by the use of LDAPActiveX component (especially its properties and methods). The Web client can be used for browsing, searching, adding, deleting the entries in the directory, etc..

It is important to point out that such a Web application is just one possible use of the LDAPActiveX and LDAPDll components. These can be included and used also in other applications which require LDAP protocol and access to LDAP directories.


Ldap klient - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address
Go Links
Done
Local intranet

LDAP klient

Brskanje po imeniku

Iskanje po kriterijih

Dodajanje v imenik

Brisanje iz imenika

Odjava s strežnika

Fig. 6. LDAP web client interface (in Slovenian language)

3 Conclusion

The presented Web interface for browsing and administrating LDAP directories simplifies the use of LDAP directories and enables universal access from arbitrary locations. It can be described as a simple Web-LDAP converter. Our module is a sort of demonstrative application, adapted for the use in our laboratory and with our LDAP directory. Due to this fact, the converter has certain limitations. Some LDAP protocol functions are not supported, but all commonly used LDAP commands are available.

4 Acknowledgment

This work has been supported by the Ministry of Education, Science and Sport of Slovenia within the program: Algorithms and optimization methods in telecommunications.

References

[1] J. Sodnik, "Preprost protokol za dostop do imenika LDAP," Diplomsko delo univerzitetnega študija, Ljubljana, 2002. (Slovenian language)

[2] T. Howes, M.C. Smith, G.S. Good, T.A. Howes, "Understanding and Deploying Ldap Directory Services", Macmillan Technical Publishing, 1999.

[3] M. Wahl, T. Howes, S. Kille, "Lightweight Directory Access Protocol (v3) ", RFC 2251, 1997.

[4] T. Howes and M. Smith, "LDAP: Programming Di-rectory-Enabled Applica-tions with Lightweight Di-rectory Access Protocol", Macmillan, Indianapolis, Ind., 1997.

[5] H. Johner, L. Brown, F.S. Hinner, W. Reis, J. Westman, "Understanding LDAP", International Technical Support Organization, IBM, http://redbooks.ibm.com, 1998.

[6] H. Johner, M. Melot, H. Stranden, P. Widhiasta, "LDAP Implementation Cookbook", International Technical Support Organization, IBM, http://redbooks..ibm..com, 1999.

[7] OpenLDAP, http://openldap.org

[8] PhpLDAPAdmin, http://phpldapadmin.sourceforge.net/

[9] R. Drach, "Serving scientific data over the Web," Computing in Science & Engineering, vol. 2(6), pp. 14 – 18, 2000.

[10] C.S. Yang, C.Y. Liu, J.H. Chen, C.Y. Sung, "Design and implementation of secure Web-based LDAP management system", Proceedings. 15th Interna-tional Conference on Information Networking, pp. 259 - 264, 2001.

[11] FiveSight Technologies, "How to Port OpenLDAP to Windows", http://fivesight.com/downloads/openldap.asp

[12] M. Prtenjak, "C++ za velike in male", Izola: Desk, 1995 (Slovenian language).

[13] S. Powers, " Developing ASP components", Sebastopol, CA : O'Reilly, cop. 2001.

J. Sodnik received the B.Sc. degree in telecommunications from the University of Ljubljana, Slovenia, in 2002. He is currently working on his Ph.D. thesis. He is Junior Researcher at the Faculty of Electrical Engineering. His research interests are acoustics, signal processing and telecommunication networks.

S. Tomazic received the B.Sc. degree in 1979, the M.A. degree in 1981 and the Ph.D. degree in 1991, all from University of Ljubljana. He is a professor at the Faculty of Electrical Engineering, University of Ljubljana, where he teaches telecommunications and signal processing. He has already worked as the telecommunications advisor for the Ministry of Defence, information technology advisor for the Ministry of Education and the national coordinator of the telecommunications research in Slovenia. Currently he is the head of the Telecommunications Department and the head of the Laboratory of Communication Devices at the Faculty of Electrical Engineering.