

01 Jul 1993

## X.500 Directory Service Support for Electronic Mail

Mihai G. Sirbu

Fikret Erçal

Missouri University of Science and Technology, ercal@mst.edu

Follow this and additional works at: [https://scholarsmine.mst.edu/comsci\\_techreports](https://scholarsmine.mst.edu/comsci_techreports)



Part of the [Computer Sciences Commons](#)

---

### Recommended Citation

Sirbu, Mihai G. and Erçal, Fikret, "X.500 Directory Service Support for Electronic Mail" (1993). *Computer Science Technical Reports*. 45.

[https://scholarsmine.mst.edu/comsci\\_techreports/45](https://scholarsmine.mst.edu/comsci_techreports/45)

This Technical Report is brought to you for free and open access by Scholars' Mine. It has been accepted for inclusion in Computer Science Technical Reports by an authorized administrator of Scholars' Mine. This work is protected by U. S. Copyright Law. Unauthorized use including reproduction for redistribution requires the permission of the copyright holder. For more information, please contact [scholarsmine@mst.edu](mailto:scholarsmine@mst.edu).

X.500 DIRECTORY SERVICE SUPPORT

FOR ELECTRONIC MAIL

M. Sirbu\* and F. Ercal

Department of Computer Science

University of Missouri - Rolla

Rolla, MO 65401 (314)341-4491

\*This report is substantially the M.S. thesis of the first author, completed July 1993.

X.500 DIRECTORY SERVICE SUPPORT  
FOR ELECTRONIC MAIL

by

Mihai G. Sirbu, 1962 -

A THESIS

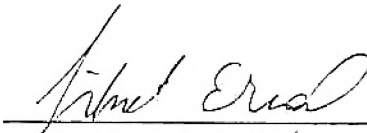
Presented to the Faculty of the Graduate School of the  
UNIVERSITY OF MISSOURI—ROLLA


In Partial Fulfillment of the Requirements for the Degree

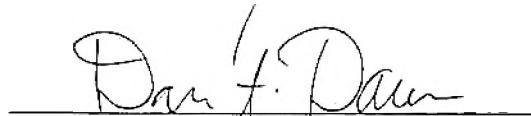
MASTER OF SCIENCE IN COMPUTER SCIENCE

1993

Approved by

  
\_\_\_\_\_  
Ekret Ercal, Advisor

  
\_\_\_\_\_  
Arlan R. DeKock

  
\_\_\_\_\_  
Darrow F. Dawson

## ABSTRACT

One of the difficult problems on the Internet is finding the electronic mail addresses of users. In practice, there are some indirect ways of finding these addresses such as the *finger* program in UNIX, but almost all of these methods require the user to know the exact host name of the destination. What is most desirable is an automated mechanism which provides the e-mail addresses of users if some minimal information about the destination site is known.

This thesis describes the design of such a directory service support system, based on the X.500 Series of CCITT Recommendation, for the *elm* electronic mail facility in the UNIX environment. The purpose of the system is to provide easy access to the electronic mail addresses of users, without requiring the direct use of the X.500 directory system. To facilitate the service, the user only needs to know the approximate name of the person and the location he/she works at. The query is done automatically and the returned e-mail address is used for the "To:" field in the header. In case of multiple hits, the sender has a few options: he can choose between the possible solutions offered by the system, ask for more information about a person or eliminate ambiguous references from the original request.

The system was designed and implemented on a Sun SparcStation. Tests were conducted to measure the response time of the system and the results are reported. Work is in progress to provide more functionality and a better user interface.

## ACKNOWLEDGMENTS

I would like to express my thanks to Dr. Fikret Ercal, my advisor. A true professor and friend, he was there every time I needed him during the research and writing of my thesis. Also, I would like to thank Dr. Arlan R. DeKock and Dr. Darrow F. Dawson for being on my committee.

The most special thanks go to my family, without whose constant and warm support this thesis would never have been completed.

I would like to thank Angela Lammers, Sanjay Nayar, and Matthias Mayer for their friendship, encouragement and intelligent solutions provided during this work.

Thank you to those who had the patience to check and correct the language and grammar for my thesis: Angela Lammers, Rick and Stella Rogers, Randie El-Amin.

The support provided by the Computer Science Department of the University of Missouri - Rolla through a half-time GTA position is gratefully acknowledged.

A special gratitude for the Computer Science Department system administrators team, which had the patience to answer a lot of strange questions. Dave Mentis, Eric Jui-Lin Lu, Chris Gorsuch, Robert Ziegler, thank you again.

## TABLE OF CONTENTS

	Page
ABSTRACT . . . . .	iii
ACKNOWLEDGMENTS . . . . .	iv
LIST OF ILLUSTRATIONS . . . . .	viii
LIST OF TABLES . . . . .	ix
SECTION	
I. INTRODUCTION . . . . .	1
A. LINKING E-MAIL WITH DIRECTORY INFORMATION . . .	1
B. THESIS OUTLINE . . . . .	2
II. THE X.500 DIRECTORY SERVICE STANDARD . . . . .	4
A. MODELS OF EXISTING DIRECTORY SERVICES . . . . .	4
1. Telephone System Directory . . . . .	4
2. WHOIS Database . . . . .	4
3. Domain Name Service . . . . .	5
B. THE X.500 APPROACH FOR THE DIRECTORY . . . . .	5
C. OVERVIEW OF THE DIRECTORY . . . . .	6
1. The Directory Model . . . . .	6
2. The Information Model . . . . .	7
3. Basic Services Provided by the Directory . . . . .	10
4. Directory Protocols . . . . .	11
D. DISTRIBUTED OPERATION OF THE DIRECTORY . . . . .	11
1. Chaining . . . . .	13
2. Referrals . . . . .	14

3.	Replication of Data . . . . .	16
III.	THE ISODE-QUIPU PACKAGE . . . . .	17
A.	ISODE AND THE DIRECTORY SERVICE . . . . .	17
1.	General Information . . . . .	17
2.	The White Pages Pilot Project . . . . .	17
3.	ISODE Availability . . . . .	18
B.	INSTALLATION . . . . .	19
1.	Starting Up a Quipu DSA . . . . .	20
2.	Management of the DSA . . . . .	21
C.	LINKING THE DSA TO THE WHITE PAGES PILOT . . . . .	21
IV.	CONNECTING E-MAIL TO THE DIRECTORY SERVICE . . . . .	23
A.	DESIGN ISSUES . . . . .	23
1.	elm . . . . .	24
2.	FRED . . . . .	25
B.	CONNECT DAEMON . . . . .	26
1.	Structure . . . . .	26
2.	Implementation . . . . .	28
3.	User Interface . . . . .	28
C.	PROJECT DEVELOPMENT . . . . .	29
V.	EXPERIMENTATION AND DISCUSSIONS . . . . .	31
A.	RESULTS . . . . .	31
1.	Functionality . . . . .	31
2.	Timings . . . . .	31

B. DISCUSSION . . . . .	33
C. FUTURE RESEARCH DIRECTIONS . . . . .	34
1. Short term research directions . . . . .	34
2. Long term research directions . . . . .	35
VI. CONCLUSIONS . . . . .	36
BIBLIOGRAPHY . . . . .	37
VITA . . . . .	40



## LIST OF ILLUSTRATIONS

Figure	Page
1. Model of the Directory and its users . . . . .	7
2. Structure of an entry in the Directory Information Base . . . . .	8
3. Portion of a hypothetical DIT . . . . .	9
4. User's View of the Directory . . . . .	10
5. Directory Protocols . . . . .	12
6. Global view of a typical "Search" request processing . . . . .	14
7. Generation and subsequent execution of a referral . . . . .	15
8. Downloading of the ISODE software . . . . .	19
9. Linking e-mail to the Directory Service . . . . .	23
10. The structure of the CONNECT System . . . . .	24
11. CONNECT Process Structure and Communication . . . . .	26

## LIST OF TABLES

Table	Page
I. FTP sites for the ISODE software . . . . .	18
II. US/Canada FTP sites for the elm software . . . . .	25
III. Timing results . . . . .	31

# I. INTRODUCTION

## A. LINKING E-MAIL WITH DIRECTORY INFORMATION

The Internet is now expanding at an exponential rate [19]. Each month thousands of new hosts, users and services are added to the already crowded network. Under these conditions, finding the e-mail address of a user or a specific service on the network is getting more and more difficult. There are a number of indirect ways available (such as the *finger* and the *whois* programs), but almost all of these methods require exact knowledge, for example the host name of the destination. The CCITT standards for the directory service (see [8] to [16]) are designed to provide guidelines for such general purpose utilities.

Models for the directory service are the original Telephone System Directory Service, the centralized WHOIS database and the Domain Name Service (DNS) database which is used for routing purposes in the Internet. A more detailed presentation of these can be found in Chapter II. Due to particular limitations, none of these systems is able to process the volume and diversity of information necessary for an Internet directory service.

The current X.500 directory service is a distributed database used to store information about the users on the Internet. The directory is an OSI application, so most of the implementations run on top of ISODE, a support structure that provides OSI compatibility with the existing Internet TCP/IP protocol. The database is organized as a tree, having different countries as children of the root. Information is stored and managed locally on different nodes in Directory System Agents (DSAs). A

user can query the system through an interface called Directory User Agent (DUA). When queried, the DUA binds to the nearest DSA, and then is referred to the one that has the required data. To improve the efficiency, a replicating mechanism (cache) for the information is provided. Among other capabilities, the system is able to make a search with minimal information, and returns a list of possible entries that match all the keywords provided.

Obviously, information about a user cannot be obtained if he/she is not registered in the directory. Currently, the directory system is developing at an accelerating rate. As of 1992, there are over 25 countries taking part in the project, mainly from Western Europe, North America, and Australia. The US White Pages Pilot Project has over 100 DSAs running, with more joining every month.

The purpose of this thesis is to automate the procedure of finding the e-mail address of a user. The user does not have to search for an address. The mailer does this operation mostly by itself. If no destination address is found, the mail is returned with "unknown address". If more than one "hit" is scored, the user has to decide in an interactive manner to whom the message will be sent.

## **B. THESIS OUTLINE**

The second chapter is a brief presentation of the X.500 Directory Service Standard. It includes the underlying structure, the protocols and the distributed operation of the directory.

The third chapter is devoted to the ISODE-Quipu package. At this point, the ISODE system is the most extensive free implementation of the OSI protocol stack. On top of it an OSI application can run over the existing TCP/IP internet. The

major part of the chapter discusses the ISODE implementation and how to set up a DSA under the Quipu application. The White Pages Pilot Project is introduced, together with the necessary procedure to link the DSA to the Pilot.

Chapter IV describes the implementation of the CONNECT application. Different design issues are analyzed and the advantages of the existing one are discussed. The two interface programs (elm as the e-mail system and FRED as a front end to the directory) are presented. The structure of CONNECT and implementation details are given.

Experimental results and the performance analysis of the system are reported in Chapter V. Advantages and existing problems are discussed. Future research directions are given at the end of the chapter.

Conclusions are in Chapter VI which is followed by the references used in the writing of the thesis.

## II. THE X.500 DIRECTORY SERVICE STANDARD

### A. MODELS OF EXISTING DIRECTORY SERVICES

To better understand the design decisions taken in the X.500 standards, three existing directory systems will be analyzed. The first is the telephone system, well-known to all its users. Two more systems which coexist on the Internet, the WHOIS database and the Domain Name Service will be presented. The advantages and the possible drawbacks of extending these types of services to a very dynamic Internet will be discussed.

1. Telephone System Directory. The worldwide telephone system could serve as a model for many applications of computer networks. All the stations (end user phones) are connected into a single, transparent network, and any pair can be linked in a conversation. If we have to pass local boundaries, an access code and the addition of some numbers to the dialed information (area code, country prefix, etc.) give us the right connection. If we need an unknown number and we have additional information about the user (name, address, etc.), we can dial the telephone directory service or look up the number in a printed directory. Although some systems provide additional functionality (e.g. Minitel-France), the scope of the Internet directory is much broader than just finding people's e-mail addresses. In the future, routing applications should be able to use the information from the Internet directory without the need to refer to present routing protocols.

2. WHOIS Database. The WHOIS database is centrally managed by the Defense Data Network – Network Information Center DDN NIC [18] and gives us information about services and people available on the network. The database is

updated manually. The processing delay can take up to two weeks. Advantages of the system are perfect coherence of the data (even if some information is outdated), and easy access. Some important drawbacks of the systems are slow answer time at peak loads, the potential for errors due to additional handling of the information, and poor reliability in case of link failure to the single information server.

3. Domain Name Service. The Domain Name Service (DNS) [18] forms the backbone of the routing system, by providing a correspondence between host and domain names and IP addresses. An unknown host is looked up on the nearest server, which eventually will pass the request if it cannot be fulfilled. Secondary nameservers can be identified for each domain so that an unreachable network will not cut off the service. The implementation is robust and its reliability has been proven by real-time usage. However, difficulties arise if we try to extend this approach to a directory service. The major drawbacks are a limited number of data types, a limited search capability, and that the data base is not easily expandable.

## B. THE X.500 APPROACH FOR THE DIRECTORY

After an extensive analysis of the advantages and disadvantages of the previous directory systems, some of which are outlined above, a distributed database structure was chosen for the OSI X.500 directory implementation by the CCITT commission. This structure directly offers the following features:

- decentralized maintenance
- powerful searching capabilities
- single global namespace

- structured information framework
- standard-based directory services

Particular implementations have added extra features, such as color photos in the JPEG format.

Problems in the use of the current directory system are:

- slow search over a big namespace
- slow search due to the network delays and poor algorithms
- tradeoff between performance and security issues

While security is not yet a major problem in the implementation of the directory, a discussion about confidentiality is in progress [20]. Not every organization and not every person desires unlimited access by strangers to the data stored in their directory database. With the widespread use of computer systems, such concerns will become more common.

### C. OVERVIEW OF THE DIRECTORY

Most of the elements in this part are credited to the excellent paper “Distributed Operation of the X.500 Directory” [1] by Bohdan Smetaniuc. This paper is highly recommended to any researcher who studies the X.500 directory.

1. The Directory Model. The directory is an on-line distributed system which holds information about objects and which provides users with services for accessing the information. It includes a set of cooperating systems, each of which contains some directory information and executes the directory protocols.



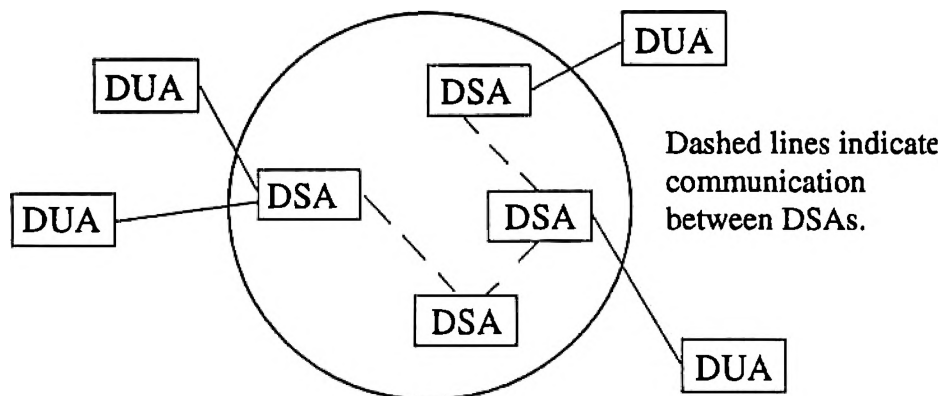


Figure 1. Model of the Directory and its users

The Directory System Agent (DSA) is an application process that represents the directory functionality.

The Directory User Agent (DUA) is an application process running in the user's domain. It provides an interface to the information stored in the DSAs, information that can be used either by people or computer programs (i.e. the X.400 electronic messaging system).

The structure of the system is given in Figure 1.

**2. The Information Model.** All the information stored in the directory about different objects is known as the **Directory Information Base (DIB)**. The information is stored in the form of entries. The structure of an entry is shown in Figure 2.

The DIB has the following particular characteristics:

- it is distributed
- the number of queries is much greater than the rate of the updates
- temporary inconsistencies are quite acceptable

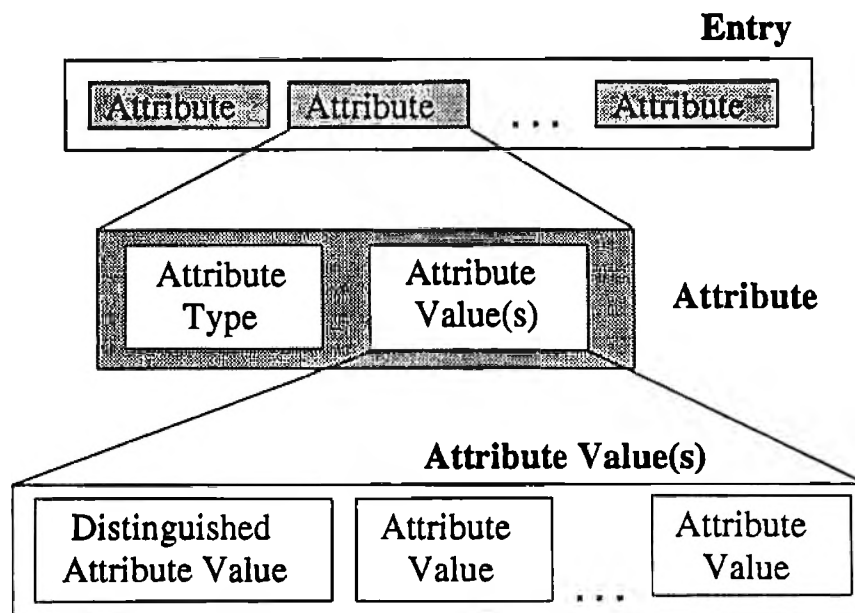


Figure 2. Structure of an entry in the Directory Information Base

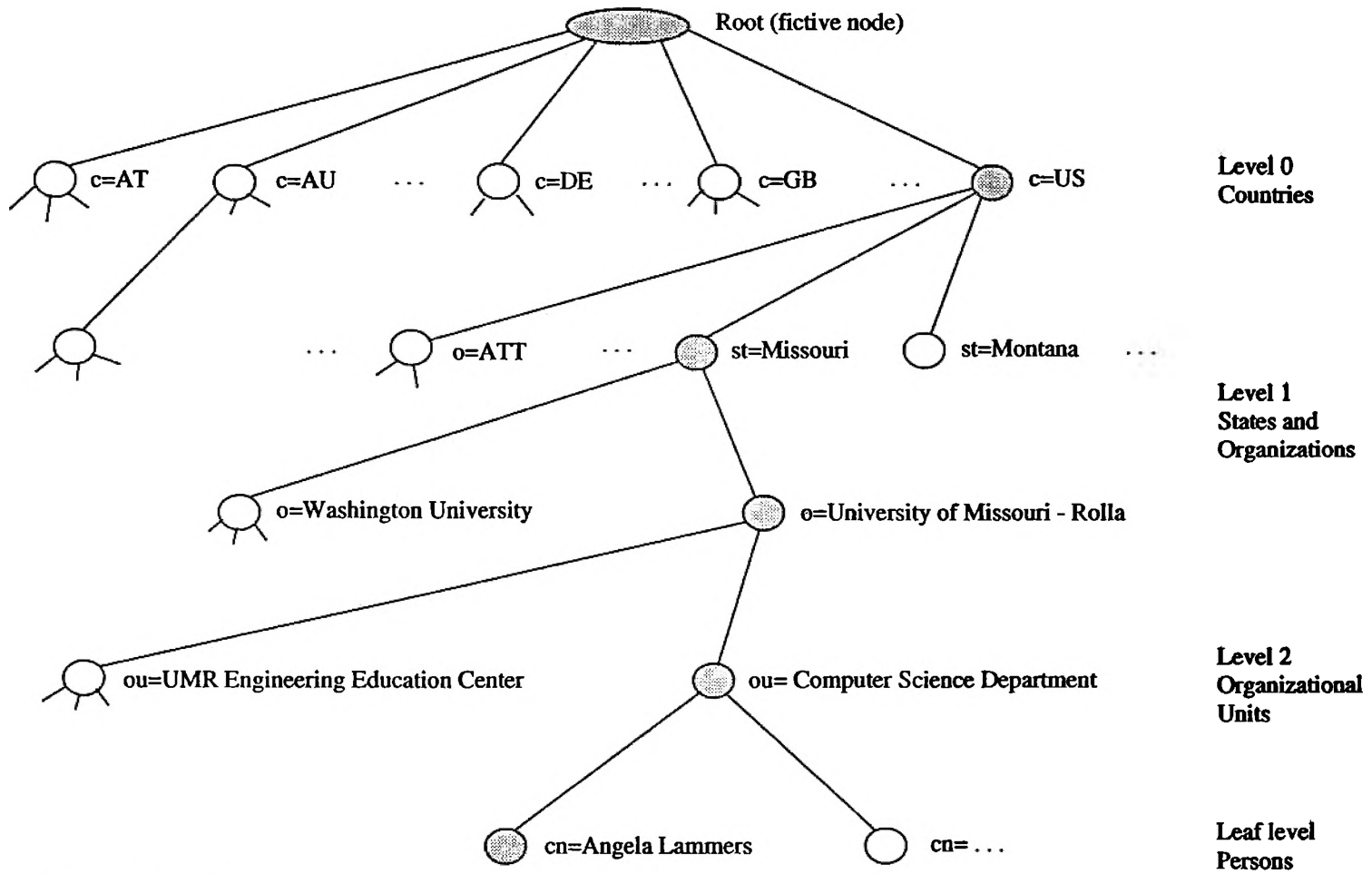
The entries in the DIB are arranged in the form of a tree known as the **Directory Information Tree (DIT)**, reflecting the natural hierarchical relations among the objects. A portion of a DIT is shown in Figure 3. A person works in a Department, which is part of an organization, which has its headquarters in some country. The root of the DIT is not an entry. This is the reason the symbol for the root in Figure 3 is different.

Given such a huge namespace, we have to distinguish any two objects that are stored in the DIB. Because of this, the directory provides a naming scheme that ensures a unique identity for each object. Following are some of the elements:

**Attribute Value Assertion (AVA):** is a pair (attribute type, attribute value), usually written as “a=b”, where “a” is the attribute type and “b” the attribute value.

**Relative Distinguished Name (RDN):** is one or more attribute value

Figure 3. Portion of a hypothetical DIT



Distinguished Name:

@c=US@st=Missouri@o=University of Missouri - Rolla@ou=Computer Science Department@cn=Angela Lammers

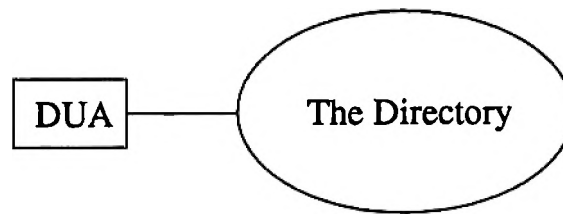


Figure 4. User's View of the Directory

assertions chosen when the entry was created and unique among the entries with the same superior node.

**Distinguished Name (DN):** the concatenation of the RDNs of all superior entries - starting at the hypothetical entry root - and the entry's own RDN. The DN of an object is the DN of its object entry.

Figure 3 shows the RDNs for each entry. The DN for one node is the list of RDNs for each node along the path from the DIT root to the current node. In our example, we concatenate the RDNs for the grey nodes. The DN for the marked leaf is given at the bottom of the figure.

**3. Basic Services Provided by the Directory.** From the user's point of view (Figure 4) the directory is an integrated system capable of performing various services. The two main service types are retrieval and modification of the DIB information.

The following operations are supported by the directory:

- Read
- Compare
- List
- Search

- Abandon
- AddEntry
- RemoveEntry
- ModifyEntry
- ModifyRDN

The user DUA sends a message to the directory requesting that one of the above operations be performed. The directory responds to each DUA request by returning either a result or an error. An error can occur in case of incorrect data, or an internal error in the system or because permission was denied to the user for this specific operation.

**4. Directory Protocols.** There are two main directory protocols as presented in Figure 5.

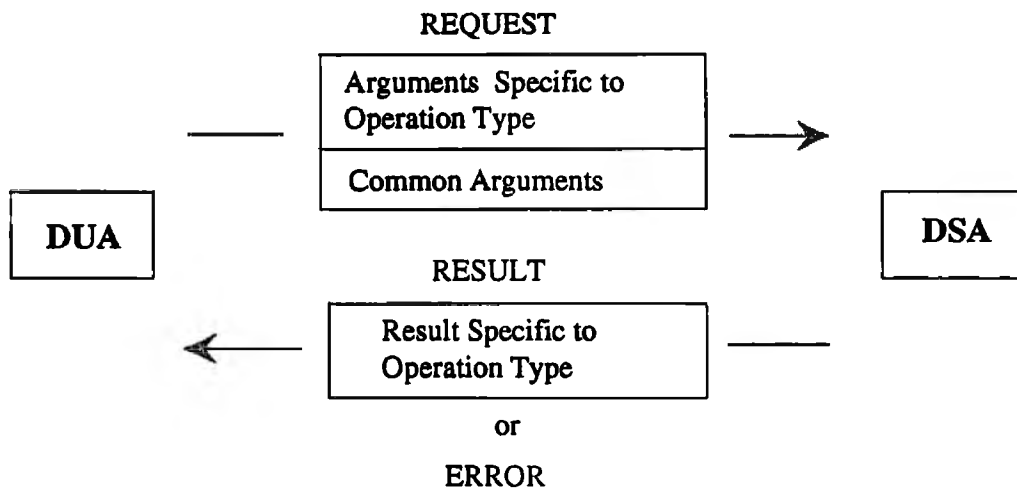
The **Directory Access Protocol (DAP)** defines the exchange of requests and responses between a DUA and DSA.

The **Directory System Protocol (DSP)** defines the exchange of requests and responses between two DSAs.

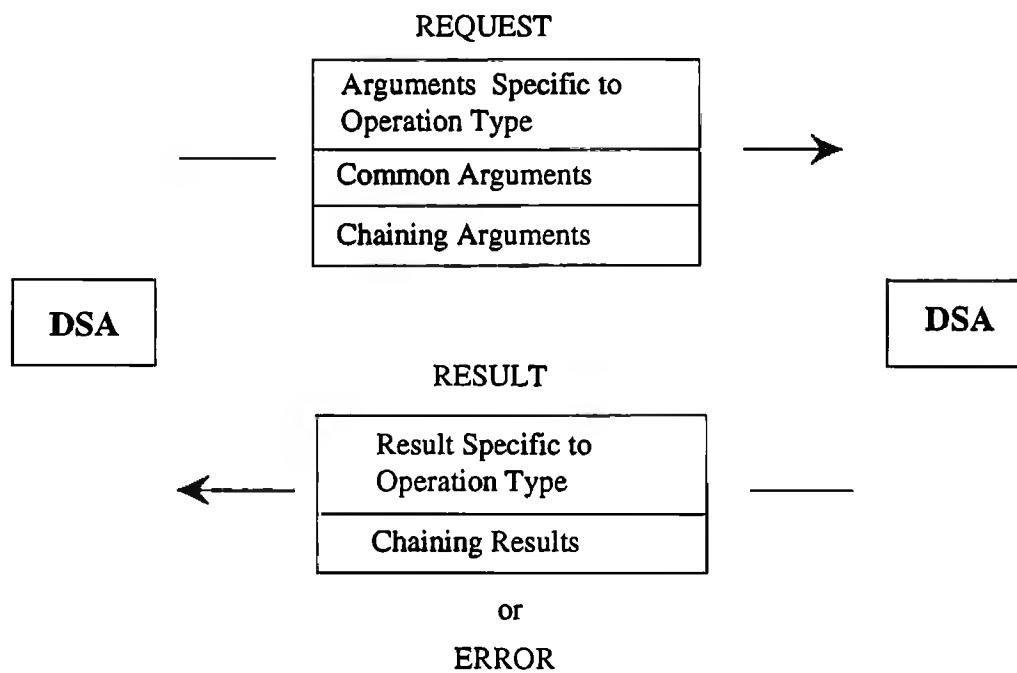
The protocols use the X.219 (Remote Operation Service - ROS) CCITT Recommendations [7].

#### **D. DISTRIBUTED OPERATION OF THE DIRECTORY**

From the processing point of view, the directory operations fall into one of the following categories:



a) DAP: Exchange protocol between a DUA and a DSA



b) DSP: Exchange protocol between two DSAs

Figure 5. Directory Protocols

- **Single-object operations:** these involve a single entry (Read, Compare, AddEntry, RemoveEntry, ModifyEntry and ModifyRDN). The evaluation is performed by the DSA that holds that entry.
- **Multiple-object requests:** these can involve multiple entries (ex. List and Search). Evaluation may need to be performed by several DSAs.
- **Abandon operation:** this does not involve any entry. The abandon request simply follows the path taken by the request to be cancelled, and will inform all the involved DSAs to cease processing the request.

In addition to the DSAs holding the entries which are subject to the request, other DSAs may be involved in the processing by routing requests to these DSAs.

Currently, the standard employs three different data distribution mechanisms: chaining, referrals, and replication [1, 5]. These mechanisms are discussed in the following subsections.

**1. Chaining.** Chaining is the main information search strategy in the directory. It involves a number of global steps. Processing of a request by a DSA begins when it is submitted by a DUA. If the receiving DSA cannot fully evaluate the request, it will accomplish evaluation by cooperating with several other DSAs. After the request is received, the DSA enters a global phase called Name Resolution. During this phase, the request is passed from DSA to DSA until the DSA holding the principal entry of the request is reached. Evaluation will start as the second global phase of request processing. For a multiple-object operation, the DSA sends – where necessary – subrequests to the DSAs which hold immediate subordinates of these entries. The process is recursive.

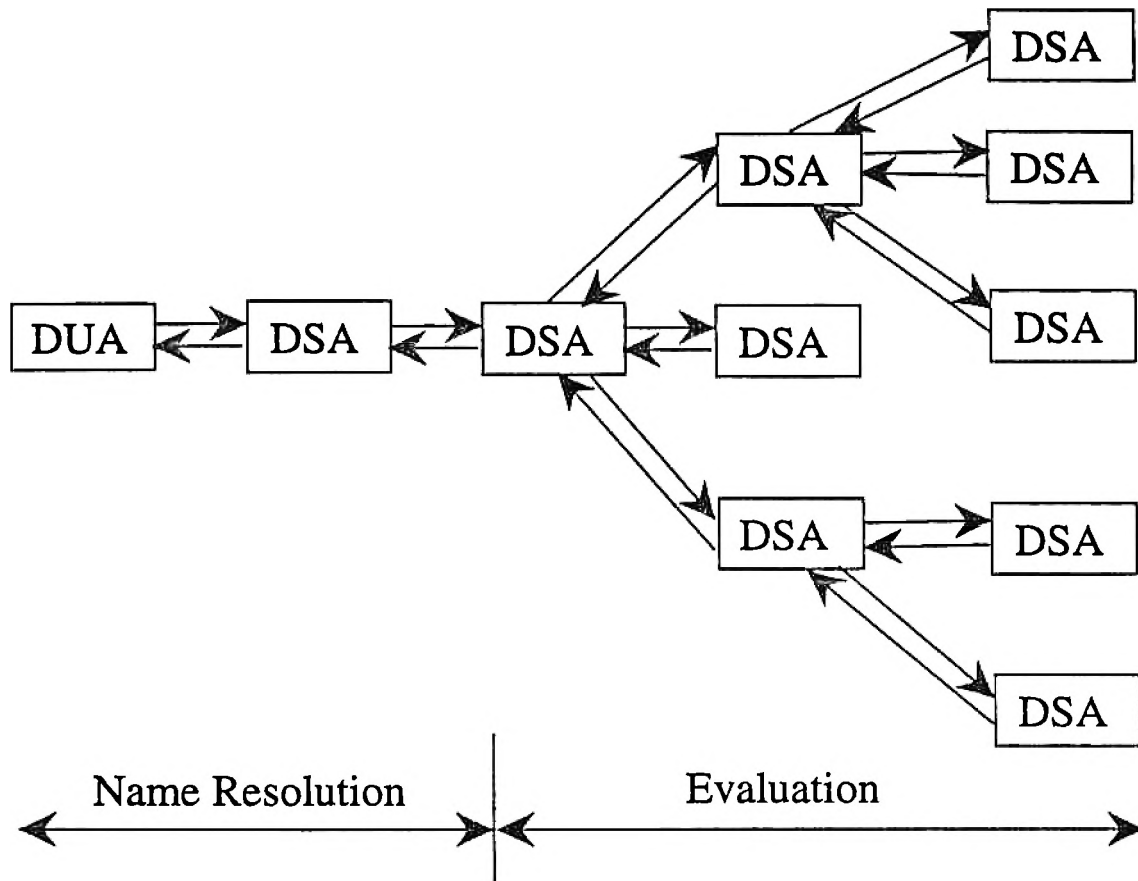


Figure 6. Global view of a typical "Search" request processing

Figure 6 shows the processing of a typical Search request. The full description of the procedure can be found in the X.518 Recommendation for Distributed Operation of the Directory [12].

2. Referrals. Figure 7 shows a different approach when evaluating a request. Sometimes, not all the chaining required for a full evaluation takes place. There can be several reasons for this:

- the DSA involved (for example C in Figure 6), is not willing to chain to another DSA



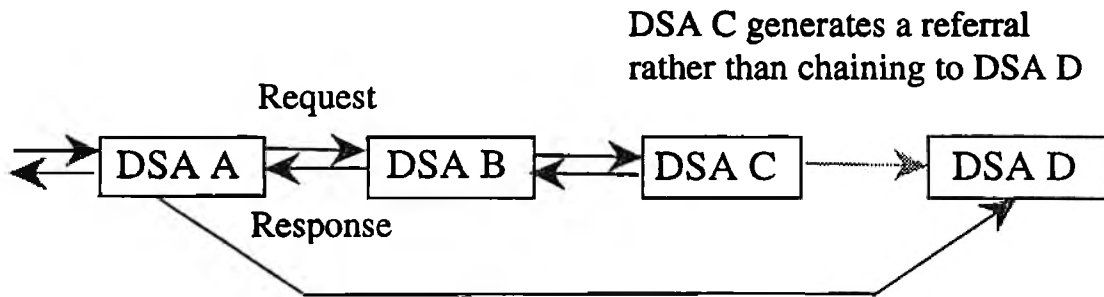


Figure 7. Generation and subsequent execution of a referral

- the DSA cannot chain to another DSA due to technical reasons (target off-line, network unreachable, etc.)
- the user requested no further chaining by setting the specific service control (“chainingProhibited”) to TRUE

Instead of connecting to the next DSA, the request accompanied by the name of the DSA to which it should have been sent is routed back on the calling path. Such an operation is called a Referral. Each DSA along the request path has the following courses of action available:

- it may try to execute the referral itself, since all the necessary information is available
- it can pass the referral to the calling DUA/DSA

Regarding Figure 7, let’s presume that the connection between DSA C and DSA D is not open due to one of the reasons outlined above. This would lead to the referral of the request back to DSA B. If the DSA B is on the same network as C, it will not be able to chain to D itself. DSA B will thus refer the problem to the caller, DSA A. In this case, a chaining to DSA D is possible, and the request will be

executed. The referral is transparent to the nodes upward of DSA A in the calling tree. Such nodes would not know the exact pattern of the execution.

**3. Replication of Data.** The replication mechanism is also referred to as “shadowing”. A server will receive a copy of the information from another DSA, followed by periodic updates of the information. There is no attempt to perform the updates synchronously, but the process maintains the replicated information reasonably up to date.

In [26], Kille discusses some of the needs for a stronger replication mechanism. Increase of performance, availability and reliability are the main reasons for its use. Additional replication issues are analyzed in [2].

Unfortunately, the X.500 standard does not provide a powerful replication mechanism. It is expected that the next versions will correct this omission.

### III. THE ISODE-QUIPU PACKAGE

#### A. ISODE AND THE DIRECTORY SERVICE

1. General Information. The ISODE package is a non-proprietary development tool used to test and implement OSI applications. Usually, it operates on top of the TCP-IP layers that currently control the Internet communication, but it can also work on pure OSI lower layers of software. The package has a wide range of functions and possibilities, and includes the work of many dedicated programmers. It is necessary to point out that although the programs are openly available, they are *not in the public domain*.

The main purpose of ISODE is to accelerate the creation, development, and use of true OSI applications. Experience has shown that the development of application level protocols take as long as or significantly longer than the development of lower level protocols [28]. Another purpose of ISODE is to make an OSI tool available to the US Internet community. Due to the success of the Internet protocols, US programmers were reluctant to get involved actively in experimenting with OSI protocols.

Most applications that implement the X.500 standard today are running on top of the ISODE platform.

2. The White Pages Pilot Project. Introducing the X.500 OSI service is a complicated task. A pilot project has been undertaken, both to demonstrate the viability of the new service and to provide this extended service to a widely distributed community.

Performance Systems International, Inc, is the sponsor of the White Pages Pilot Project for the United States. PSI manages the "@c=US" level zero DSA and

Hostname	IP Address	Programs
ftp.psi.com	38.146.10.5	ISODE, Quipu, DSA for US
cs.ucl.ac.uk	128.16.5.31	ISODE, Quipu

Table I. FTP sites for the ISODE software

provides registration for the local DSAs. Additional information is available to the sites that want to start a DSA and be part of a unified effort (see also section C in this chapter). Support for working with ISODE and QUIPU – the official software of the pilot – is given to sites that have joined the project. While it is possible to use other software packages that conform to the X.500 protocols, no assistance is given to users of such packages by the pilot managers.

**3. ISODE Availability.** The ISODE package can be downloaded from a series of anonymous FTP sites. Table I lists some of them. A numeric IP address is included in case the site is not visible (i.e., the ftp program does not recognize the host name as valid).

The ISODE distribution package is a big one. The source files - usually found under `isode-8.tar.Z` - are about 5.5MB in size, in tar and compressed form. There are also manuals available. A  $\LaTeX$  version can be found as `isode-7-doc.tar.Z` (3.5MB), and a PostScript version is available as `isode-7-ps.tar.Z` (4.3MB). The changes for version 8 in the manuals are available as an update release under number 8 from the same sites.

If a site plans to start a local DSA and join the pilot, an additional file is required: `wpp-addon.tar.Z`. The files and the information to start a US-White Pages compatible program are included in it. A sample session for downloading the files necessary to a US-based organization is given in Figure 8. The same approach can

```

msirbu(ercalisc-66) ftp ftp.psi.com
Connected to batcher2.psi.com.
220 batcher2.psi.com FTP server (SunOS 4.1) ready.
Name (ftp.psi.com:msirbu): anonymous
331 Guest login ok, send ident as password.
Password: guest
230 Guest login ok, access restrictions apply.
ftp> binary
200 Type set to I.
ftp> cd wp/src
ftp> get isode-pilot.tar.Z
ftp> get wpp-addon.tar.Z
ftp> bye
221 Goodbye.
msirbu(ercalisc-67) uncompress < isode-pilot.tar.Z | tar xfbp -
msirbu(ercalisc-68) uncompress < wpp-addon.tar.Z | tar xfbp -
msirbu(ercalisc-68) rm isode-pilot.tar.Z wpp-addon.tar.Z

```

Figure 8. Downloading of the ISODE software

be used to obtain the ISODE manuals.

## B. INSTALLATION

The process of installing ISODE is relatively straightforward, but there are some important elements:

- To install the programs, superuser permission is necessary.
- If all ISODE applications are to be installed, about 60 MB of free disk space is needed. If source files are not kept after the installation, 7 MB of disk space will be sufficient.
- If you plan to join the White Pages Pilot Project, you should follow the rules in [29]. Otherwise, follow the ISODE manuals [28] and the instructions found in the software package.

The first step is to configure ISODE for the target system. This involves copying the proper ISODE files into configuration files and setting up some directories. Scripts have to be started to compile and install the system. Since the process is long (a few hours), it is good practice to redirect the output to a file.

To start and keep ISODE operational, you need to [29]:

1. add a “tsapd” entry to your `rc.local` file.
2. add the “tsap” to the `/etc/services` file.
3. add an entry to the `crontab` file to clean the log files daily.
4. start the “tsapd daemon” by hand (only the first time).

Since the ISODE system runs without problems on a wide range of UNIX systems, there should be no problems when starting it. If something goes wrong, debugging hints are given in the documentation. In case they do not work, the user is instructed to contact one of the ISODE mailing lists.

1. Starting Up a Quipu DSA. Before the DSA start is attempted, a DUA should be operational. Of course, this means that the ISODE package is running error free on the system. The user should test the DUA by trying to connect to a well-known DSA. The “@c=US” directory servers can be used for this purpose, since their addresses are in the `tailor` file for all Quipu DUAs, and their availability is among the highest on the Internet. Then the DSA is started.

The first step is to choose a name for the DSA. It is the Quipu convention that every DSA should be named after endangered South American wildlife<sup>1</sup>. You have to

---

<sup>1</sup>For UMR, the DSA name is “Tinamou”. The Distinguished Name is “@c=US@cn=Tinamou”.

check that no other DSA exists by that name. Since it is a part of the Distinguished Name, there cannot be two DSAs with the same name.

The DSA configuration file has to be changed as above, and then the initialization script can be run.

Adding, removing entries, changing the values of attributes and adding new attributes are some of the test procedures that have to be employed to make sure the DSA is fully operational.

**2. Management of the DSA.** There are a series of services that have to be configured to allow maximum access to your directory server. Possible services include:

- White Pages via the DA-service;
- White Pages via TELNET;
- White Pages via WHOIS;
- White Pages via mail.

Discussion of these services is beyond the scope of this thesis. More information can be found in [28, 29].

### C. LINKING THE DSA TO THE WHITE PAGES PILOT

Once the DSA is running correctly for a number of days, it is time to join the Pilot Project. E-mail to the project manager informs him of the newly available DSA. He will run some tests to see if your DSA is visible from other sites on the Internet. Once satisfied that the server is running correctly, the manager will include your entry in the level-0 server and give you some connecting information. With this

action, and the subsequent updates of your directory tree, the DSA is part of the White Pages Pilot Project, and is visible from all the DSAs in the world.

It is important to note that you do not need to have an operational DSA or to be a part of the White Pages Pilot to access the directory. Any running DUA that can connect to a DSA can do the job. But if you have a DSA that is not linked to the project, information from your site will not be available to the servers outside your organization.



## IV. CONNECTING E-MAIL TO THE DIRECTORY SERVICE

### A. DESIGN ISSUES

There are a number of ways to allow direct access by an e-mail system to the information stored in the directory service. As seen in Figure 9, any such application needs two interfaces: one with the directory system and one with the electronic mailing system.

There are a number of options available when building such a system. The one adopted for this project was to take two existing interfaces and link them together by the use of interprocess communication. Such an application requires an extensive knowledge of the components. Still, the solution is straightforward and time-efficient.

Other valid options include building the system from scratch or taking an existing interface (i.e. an e-mail program) and incorporating the functionality of the second interface (i.e. a DUA).

The building blocks of the **CONNECT** system are (see Figure 10):

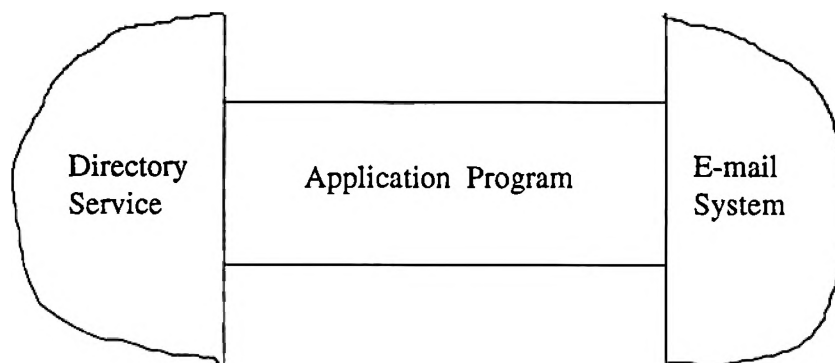


Figure 9. Linking e-mail to the Directory Service

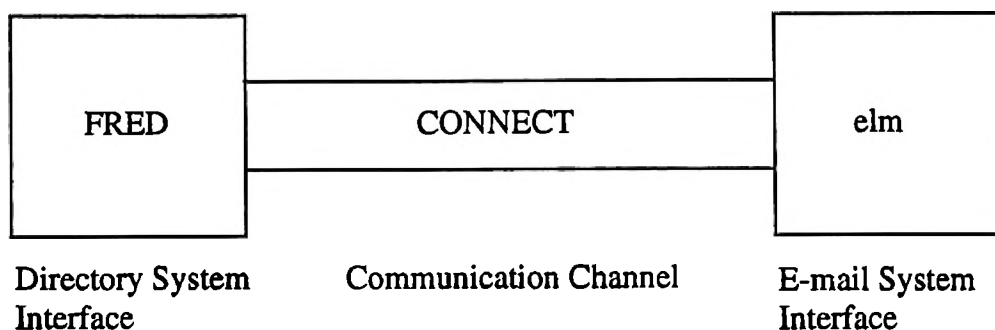


Figure 10. The structure of the **CONNECT** System

- **FRED**: a user-friendly DUA used as an X.500 directory system interface. **FRED** is an ISODE-QUIPU component.
- **elm** : a public domain electronic mail program for the UNIX environment, and also as one of the few interactive, screen-oriented mailers. The **elm** program was a clear choice for the project.
- The **CONNECT** daemon links these two interfaces to create a fully functional system.

There is an additional advantage to this approach. It is very simple to extend the system to a different pair of interfaces. The connection element and the second interface are already operational. After studying the new interface, only a link to the existing parts is required.

1. **elm**. Elm is a full screen oriented mailer for the UNIX operating system that is part of the public domain. It has most of the features of `/bin/mail` and Berkeley Mail.

The **elm** distribution package can be downloaded by *anonymous* ftp from the sites given in Table II. It consists of a compressed tar file. Upon unpacking, the user

Hostname	IP Address	path
wuarchive.wustl.edu	128.252.135.4	/mirrors/elm
ftp.uu.net	137.39.1.9 192.48.96.9	/networking/mail/elm

Table II. US/Canada FTP sites for the elm software

has to create and tailor the configuration files. The process is guided by a script and it is straightforward. After configuring the system, the program can be compiled. The executable code is created in the local `bin` directory.

Additional information about elm can be obtained by reading the references [30] to [34], or from the *comp.mail.elm* newsgroup on the Usenet.

2. **FRED.** Our selection for a directory service interface is FRED, a user-friendly DUA that is part of the ISODE distribution (see Chapter III).

The QUIPU package contains a number of DUAs that can be used for different purposes. The most complex and powerful DUA is DISH (DIrectory SHell). Working with DISH can be difficult. While it is well suited for managing an entire system, DISH is relatively difficult for an inexperienced operator to use. FRED is designed as a simple, straightforward interface to the directory system. FRED is not a stand-alone DUA. It first links to DISH (actually, FRED stands for FRont End to Dish), and executes its queries with the help of the more advanced DUA.

The simplicity and user-friendliness of FRED were the main reasons for choosing it as the directory service interface in our implementation.

The sources for FRED can be found in the `isode/others/quipu/uips/fred` directory. Usually, the binary is generated in the binary directory (`/usr/local/bin` on the “ercalisc.isc.umr.edu” system).

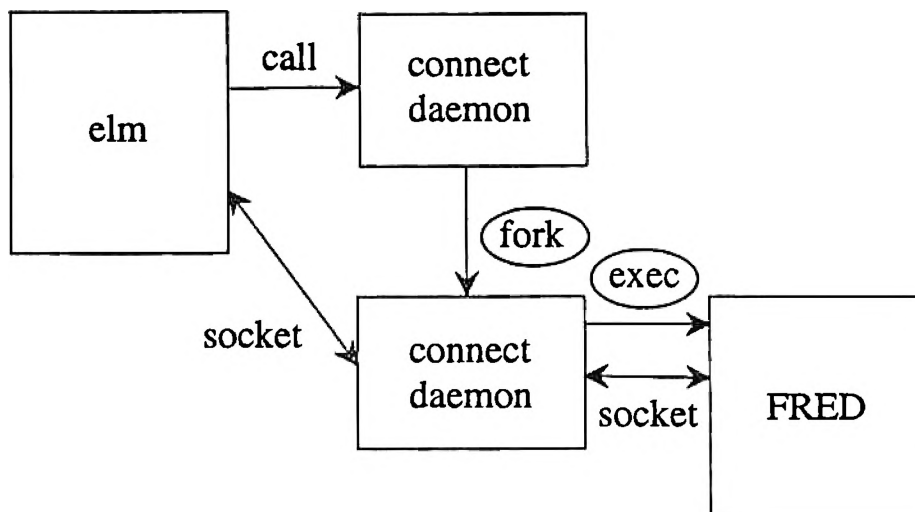


Figure 11. CONNECT Process Structure and Communication

## B. CONNECT DAEMON

1. Structure. The process structure of the **CONNECT** application is given in Figure 11. We call the computer which is running the DSA server for that particular site as the “host system”. In case no DSA is installed there, we need to have at least knowledge of an operational DSA to which our DUA will be able to link. On the same system we will install the connect daemon. The program will detach, will bind a socket and will wait for a communication request from an e-mail program.

The moment a user starts the mailer, it will request a link to the daemon. As shown in Figure 11, the mailer does not have to be on the same machine as the directory server. It will request the service via an Internet socket, so the only necessary information will be the host system name (or IP address) and the port number.

Once the daemon receives a “call”, it forks to process the connection, and the parent will continue to wait in the background for the next request. The child

starts FRED (fork + exec) and opens up a communication line to it. In its turn, FRED contacts the nearest DISH and completes the communication chain between the e-mail system and the directory service.

At this point, the communication link waits for a directory request from the mailer. When a request is received, FRED will change its local "org" string to that of the requested directory address. Then it will activate a WHOIS query with the appropriate parameters. The following outcomes are possible:

1. A single entry is returned: the e-mail address of the user is parsed by the CONNECT daemon and passed to the mailer to fulfill the request.
2. Multiple entries are found: the original search was ambiguous. Detailed information about each target is passed back to the mailer. In this case, an interactive process is started. The user has the following options:
  - accept one of the proposed entries;
  - accept all proposed entries for a group mailing;
  - request full information on one or more of the entries;
  - submit additional address information and request another search;
  - abandon the search process.

At the end, we will either have a single address that matches our initial request, a group of addresses or the command to quit the current mailing.

3. No entry is found: there is no user with the given characteristics, or he/she is not registered with the directory service. We can abandon the mail or submit a new address.

4. An error is reported: the communication or the directory system is broken. The mailer may close the communication line and try to reconnect a number of times, or report the failure and mark the directory service extension as unavailable at the present.

2. Implementation. The CONNECT daemon is implemented along the lines of a typical UNIX server. When it is started, the daemon creates an Internet socket and waits for a client to access it.

The decision to use an Internet socket was taken so that the service would be available on machines other than the directory service host. In this way, we would need only one CONNECT program running in the background for every site. The mailers can run on every machine that has access to the designated host. Because of the extra load, it would be advisable to use the connection only over high-performance networks, like LAN-s.

3. User Interface. To be a useful service, the user interface needs to be simple and straightforward. To mark an X.500 type address in the "To:" line of the e-mail message, the user will bracket it by << and >>. He can then request a Directory search for the given person. If only one hit is encountered, the internet e-mail address of the user is substituted in the "To:" field and the mailing process continues as usual. If more than one hit is present, the names are printed on the screen in the elm mailbox format. The user can accept one of them, abandon the request, select one to see more information related to that entry, or change the request to find a reduced number of hits. Also, the user can add other recipients to the list.

Basically, if the address search is a success, the mailing process continues. If not, the user is given more information to make a choice, or the entire message is

abandoned.

Note that it is important that the directory search is done at the beginning of the mailing process. It would be very frustrating to write a two-page message only to discover that your correspondent does not have his/her directory information registered with an X.500 service.

### C. PROJECT DEVELOPMENT

The following steps were performed in order to implement the **CONNECT** program:

- The X.500 Recommendations Series was obtained and examined thoroughly.
- The complete ISODE package (version 7, later replaced by version 8) was downloaded, compiled and installed on the "ercalisc.isc.umr.edu" SUN workstation.
- A local DSA with about 20 persons (nodes) was created under two organizational units (Computer Science Department and UMR Engineering Education Center - see Figure 3).
- UMR has been linked to the White Pages Pilot Project. The local DSA was started under the Distinguished Name "@c=us@cn=tinamou". At this point, it is the second DSA registered in the state of Missouri, and the only one currently available.
- The structure of the **CONNECT** daemon and the communication protocol was designed, and coded.
- FRED was modified to be able to communicate with the daemon. The connection and standalone operation were tested.

- elm was modified to allow access to the directory.
- The complete system was integrated and tests were conducted. The results are presented in the next chapter.

Effort is currently being made to adapt the user interface to the elm implicit guidelines.



## V. EXPERIMENTATION AND DISCUSSIONS

### A. RESULTS

1. **Functionality.** The integration of e-mail and the directory system was successfully implemented. We have a system available with which one can automate a directory query and use the given address in the same step.

Because of the particular design structure, multiprocessing is also available. Multiple requests can be handled by the **CONNECT** daemon by forking a child process for each user. The initial connection is made over an Internet Socket. Programs on an entire LAN can access the same daemon, reducing the need for multiple copies of DUAs on the local network.

2. **Timings.** A number of tests are conducted to study the timing response of the system. The results are given in Table III.

The system start-up time is the time needed for the **CONNECT** daemon to accept the request, to fork a child process, to start a local FRED DUA and to

Timing results in seconds				
Operation or Query	Average		Maximum	
	Network Load		Network Load	
	Low	High	Low	High
System start-up	8	8	10	10
local 1 person query	1	1	2	2
remote n persons, no specific match [st=California, o=Inform...]	2	3	5	5
remote 1 person (Europe)	8	17	10	25

Table III. Timing results

connect to the DISH program. Most of the time is taken by the FRED – DISH communication.

The low network load has been tested at around 10 p.m., central time. This is equivalent to 4 a.m. G.M.T., or 6 a.m. in Eastern Europe. The load on the local network, as well as the load on the intercontinental Internet are relatively low at that hour compared to the daytime hours when the usage is at its peak. The high for the network load was measured at around 9 a.m. central time, with a high level of traffic on both sides of the Atlantic.

For each test, a number of trials were made. The timings in Table III give the average values and the worst result for each test.

A different name was queried each time, because DISH would not initiate a new network request for a previously used name. Instead, DISH would get the information directly from the local cache. While in normal operation a cache hit would improve the performance, and therefore would be desirable, for our purposes it would lead to wrong results and is not desired.

The first row gives the time necessary to find a person registered in the local DSA. This time is important if local e-mail routing is desired in the future.

The second query was to look for all the “John”s at a California Research Institute. In this test, it is assumed that we did not have the exact name or the e-mail address of the person, but we knew his/her company.

For the third test, a specific European person was considered. The complete information including name and organization was provided in this case.

## B. DISCUSSION

As expected, the US Internet structure provided a good directory response time, even for long-distance requests. The Internet network was developed in the US and has been operational for a long time. Thus, the hardware has high performance and provides sufficient bandwidth even for "long-distance" communication. This is the major reason for the good results in row number three.

During peak traffic, the search in Europe took a relatively longer time. The weak spot in the system is the intercontinental link, that cannot handle the requested load efficiently. In time, the available bandwidth for this network segment is expected to increase, hence resulting in a better response time.

In all cases, the query times are within reasonable limits. It is not a high-speed process, and thus not yet appropriate for routing applications. It is slow because of the OSI application and service running on top of a TCP-IP network. If, in the future, the network protocol is switched to the OSI protocol, and the routing is performed directly by the directory service, thus eliminating the need for Internet addresses, a significant increase in efficiency of the system is expected.

The directory responds to a growing need of the Internet community. One of its major implementation advantages is that it already provides a network standard which is ready to be used. As such, most of the current implementations are protocol-compatible, and it is expected that this trend will continue.

The directory is currently a hot research item in network and distributed programming. Nine out of the last 40 RFCs (Request for Comments) on the Internet deal with directory problems ( [17] to [27] ), ranging from technical to legal issues [20]. A new standard is due this year.

Currently, Europe, North America and Australia have X.500 services with over 300 sites, and the numbers are growing daily (see [3] to [6] ). Some universities, such as University of Michigan and Dartmouth College are already using directory servers for routing purposes.

There are a number of open research and implementation issues. New DSAs and DUAs, with increased capabilities are expected. The security issues are not fully solved yet, and the current data replication mechanism leaves room for further analysis and design. Elements of the distributed database theory can be successfully applied to solve certain problems in this area.

Currently, there are two major problems associated with using a direct link to the directory service from e-mail programs. Both of them are due to the X.500 service and not to the link:

1. The speed of the query is relatively slow.
2. Not all the sites and users are recorded in the directory. Some portions of the directory tree are voluntarily omitted by users, since they do not want the information to be available to others. With these limitations, the efficiency of the e-mail searches drops due to the "not available" queries.

### C. FUTURE RESEARCH DIRECTIONS

We may list the possible research directions in two categories: short-term and long-term goals.

1. Short term research directions. In the short term, we can attempt to link other mailers to the directory. Also, we can develop a separate, dedicated

system, instead of linking two independent applications together. We can test its performance and see if there is any significant improvement over the existing system.

Another possible research direction is to provide an e-mail distribution system within an organization based on the local directory service. For example, at UMR, there are a number of people that have accounts on different machines, or people who change their departments and thus change their e-mail addresses. Instead of propagating these changes to all their correspondents, a system could be developed which accepts mail sent to **FirstName.LastName@umr.edu**, queries the directory for the current e-mail address of the recipient, and internally routes the message to the targeted person. Currently, there is such research going on in the Computer Center at UMR, but their efforts are based on a self-developed database format and not on the X.500 system.

**2. Long term research directions.** Since the overall performance of the system depends highly on the performance of the directory service in general, any global improvement made in the standard or implementation of it will benefit an e-mail facility in general. One of the most promising research topics to date is using X.500 name information for routing packages on the network. Instead of querying a nameserver for the destination hostname, we could access a DSA for the same purpose. Of course, to rely on X.500 information for routing purposes, the current performance of the directory service needs to be improved by at least two orders of magnitude. Maybe it does not sound very optimistic, but unless major improvements are made to the system, it will not be considered more than an experimental tool. Obviously, the ultimate goal is to make the directory service a useful and a functional tool in the future networking environment.

## VI. CONCLUSIONS

The directory service is a powerful and a useful tool considering the recent increase in the size of the international networking community. Integrating it directly with e-mail programs to provide directory support will eventually spread its use among the users.

The `CONNECT` application developed in this thesis helps users to send e-mail messages by using X.500 type addresses.

Most of the drawbacks mentioned in Chapter V are inherent to the current structure of the X.500 directory service. By improving the service, all the applications built on top of it are expected to have a better performance.

The integration of the e-mail facility and the directory services is a step forward. It gives the users the flexibility of using naturally defined e-mail addresses. Furthermore, by providing a completely new service, the number of users learning about it, and the number of professionals involved in design and implementation of applications will increase. In time, the increase in the quality and the range of the OSI services will hopefully lead to the global availability of services on the network accessed through a uniform application interface.

At a point in the near future, users may very well be identified by their X.500-like addresses, and this information could be used directly in routing, making the X.500 to IP-address conversion an unnecessary step. In this sense, we can say that using the directory address at an early stage is a good practice for the future.

## BIBLIOGRAPHY

- [1] B. Smetaniuc, "Distributed Operation of the X.500 Directory", *Computer Networks and ISDN Systems*, pp. 17-40, 21 (1991).
- [2] J. M. Bennet, M. A. Bauer, "An Analysis of Replication Strategies for X.500-like Distributed Directories", *Replicated Data*, Houston, Texas, November 8-9, pp. 137-142, 1991 IEEE Computer Society Press, The Institute of Electrical and Electronics Engineers, Inc.
- [3] J. Craigie, "UK Academic Community Directory Service Pilot Project", *Computer Networks and ISDN Systems*, pp. 305-310, 17 (1989).
- [4] C. Huitema, "Proposal for Co-ordination of a RARE OSI Directory Pilot", *Computer Networks and ISDN Systems*, pp. 350-351, 17 (1989).
- [5] C. Huitema, "The X.500 Directory Services", *Computer Networks and ISDN Systems*, pp. 161-166, 16 (1988/89).
- [6] E. Mendoza, "Directory Services and COSINE", *Computer Networks and ISDN Systems*, pp. 44-47, 16 (1988/89).
- [7] -, "CCITT Recommendation X.219, Remote Operation Service", *CCITT Blue Book*, 1988.
- [8] -, "CCITT Recommendation X.500, The Directory - Overview of Concepts, Models and Services", *CCITT Blue Book*, 1988.
- [9] -, "CCITT Recommendation X.501, The Directory - Models", *CCITT Blue Book*, 1988.
- [10] -, "CCITT Recommendation X.509, The Directory - Authentication Framework", *CCITT Blue Book*, 1988.
- [11] -, "CCITT Recommendation X.511, The Directory - Abstract Service Definition", *CCITT Blue Book*, 1988.
- [12] -, "CCITT Recommendation X.518, The Directory - Procedures for Distributed Operation", *CCITT Blue Book*, 1988.
- [13] -, "CCITT Recommendation X.519, The Directory - Protocol Specification", *CCITT Blue Book*, 1988.
- [14] -, "CCITT Recommendation X.520, The Directory - Selected Attribute Types", *CCITT Blue Book*, 1988.

- [15] –, “CCITT Recommendation X.521, The Directory - Selected Object Classes”, *CCITT Blue Book*, 1988.
- [16] –, “ISO IS 9594, Information Processing Systems - Open Systems Interconnection - The Directory”, Parts 1-8, 1988.
- [17] C. Weider, J. Reynolds, S. Heker, “Technical Overview of Directory Services Using the X.500 Protocol”, RFC 1309, FYI 14, March 1992.
- [18] C. Weider, J. Reynolds, “Executive Introduction to Directory Services Using the X.500 Protocol,” RFC 1308, FYI 13, March 1992.
- [19] M. Lottor, “Internet Growth (1981-1991),” RFC 1296, January 1992.
- [20] The North American Directory Forum, “User Bill of Rights for entires and listings in the Public Directory”, RFC 1295, January 1992.
- [21] R. Lang, R. Wright, eds. “A Catalog of Available X.500 Implementations”, RFC 1292, FYI 11, January 1992.
- [22] S. E. Hardcastle-Kille, “X.500 and Domains”, RFC 1279, November 1991.
- [23] S. E. Hardcastle-Kille, “A String Encoding of Presentation Address”, RFC 1278, November 1991.
- [24] S. E. Hardcastle-Kille, “Encoding Network Addresses to Support Operation over Non-OSI Lower Layers”, RFC 1277, November 1991.
- [25] S. E. Hardcastle-Kille, “Replication and Distributed Operations Extensions to Provide an Internet Directory Using X.500,” RFC 1276, November 1991.
- [26] S. E. Hardcastle-Kille, “Replication Requierements to Provide an Internet Directory Using X.500”, RFC 1275, November 1991.
- [27] P. Barker, S. E. Hardcastle-Kille, “The COSINE and Internet X.500 Schema,” RFC 1274, November 1991.
- [28] C. J. Robbins, S. E. Kille, “The ISO Development Environment: User’s Manual, Volume 5: QUIPU”, Version 7.0, July 18, 1991.
- [29] M. T. Rose, W. Yeong, “PSI White Pages Pilot Project: Administrator’s Guide”, October 14, 1991
- [30] D. Taylor, “Elm Users Guide”, Version 2.3, Elm Distribution Package, May 1, 1990.
- [31] D. Taylor, “The Elm Reference Guide”, Version 2.3, Elm Distribution Package, May 1, 1990.



- [32] D. Taylor, "The Elm Alias System Users Guide", Version 2.3, Elm Distribution Package, May 1, 1990.
- [33] D. Taylor, "A Guide to Forms Mode in Elm", Version 2.3, Elm Distribution Package, May 1, 1990.
- [34] D. Taylor, "The Elm Filter System Guide", Version 2.3, Elm Distribution Package, May 1, 1990.

## VITA

Mihai G. Sîrbu was born in Timisoara, Romania on October 5, 1962. He received his high school education in Timisoara. In 1981 he was admitted to the Timisoara Polytechnic Institute. He started college in 1982, after doing service in the military, and graduated in July of 1987 at the top of his class with a degree in Electrical Engineering. While in college, he won two national first prizes for research in robotics software and a series of national prizes in professional contests (software development and electronics). For five consecutive semesters he was awarded the National Merit Fellowship, a scholarship given for outstanding academic achievements to less than 1% of Romanian students. He has been developing complex software since his first year in college.

In October of 1987, he joined the Enterprise for Electric Measuring Devices in Timisoara, Romania, where he worked as an electronics engineer and later as a software developer. In February of 1990, Mihai Sîrbu accepted a position of Assistant Professor in the Applied Electronics Department at the Timisoara Technical University (formerly Timisoara Polytechnic Institute). He taught labs and projects in Computer Aided Design, Microprocessor Systems Design and Basic Electronics.

Since August 1991, he has been a graduate student at the University of Missouri at Rolla, pursuing an M.S. degree in Computer Science. He holds a position as a graduate teaching assistant in the Computer Science Department. Mihai Sîrbu is a member of A.C.M. and Upsilon Pi Epsilon.