#13

The Need for X.500 Index DSAs

Paul Barker

This paper was produced by Paul Barker, in his role as consultant to DANTE on NameFLOW-Paradise development issues - June 1995.

DANTE IN PRINT is a track record of papers and articles published by, or on behalf of DANTE. HTML and Postscript versions are available from: http://www.dante.net/pubs/dip

For more information about DANTE or *DANTE IN PRINT* please contact:

DANTE
Francis House
112 Hills Road
Cambridge CB2 1PQ
United Kingdom

Tel: +44 1223 302 992
Fax: +44 1223 303 005
E-mail: dante@dante.org.uk

The Need for X.500 Index DSAs

Paul Barker

Abstract

This is a discussion paper on Index DSAs, which are advanced as part of a more service-oriented approach to using X.500. The paper argues that Index DSAs are needed to provide key services which cannot be provided by simply using the existing X.500 infrastructure. Some of the problems with the use of strict X.500 techniques are then discussed. An alternative approach is proposed using X.500 servers partially connected to the main DIT. These servers hold information in the most natural place in the DIT, but deliberately provide a particular view of the DIT. This might be a restricted view of information in the main DIT, or even a view which is enhanced by a service provider adding more information themselves.

Introduction

A few years ago, many writers, including the author of this paper, wrote confidently about X.500[1] being the directory service, or even the Directory. Despite a wariness in the Internet community about CCITT / OSI protocols, X.500 was even adopted as the basis of an Internet directory service[2]. However, while deployment of X.500 has grown steadily, it has not matched the rate of growth of the Internet, and X.500 is no longer viewed as the automatic choice as the basis for an Internet directory service. In a recent re-appraisal of Internet directory service policy [3], the emphasis has shifted to one of having several underlying technologies somehow coordinated to provide an overall service. This view is endorsed in the Whip Internet Draft [4], which is an attempt at a new specification for an Internet white pages service. My personal view is that the community will be better served by one technology, that X.500 is close enough to what we want to be a good starting point, but that we need to use X.500 creatively to provide services, rather than expecting services to magically appear. Why has take-up of X.500 been relatively slow? Several reasons have been put forward:

- many in the Internet community felt that X.500 was being forced upon them, whereas many in the community would prefer to develop their own protocols and service definitions;
- the technology is seen by some as unnecessarily complex;
- the publicly available implementation is big, hard to configure, install and manage;
- the problem of providing a white pages directory service is much harder than is generally understood[5];
- privacy concerns about directory services in general have limited deployment, especially in some parts of Europe.

An equally important problem in my view is that the balance of X.500 activities has tended to be 'protocol dominated', with a lot of effort going into conformance, using the right underlying networking technology etc, and much less effort into how services were to be provided. Some specific examples of this include:

- The recent excellent OIFP work resulted in many bugs being fixed in two or three implementations. However these bugs were not, in the main, causing service problems for ordinary users.
- The 'tick in the box' mentality associated with standards-based software has led to implementors putting effort into XDS APIs, rather than writing simple-to-use APIs, such as that provided in the University of Michigan LDAP code. The availability of this simple API has done much to encourage the writing of user interfaces which access X.500 based information.

Although it is undoubtedly more difficult writing the standard first and then the code (the OSI way), rather than the other way round (the Internet way), the problem is that OSI standards come to

Paul Barker has worked as a consultant for DANTE on NameFLOW-Paradise. He is employed by University College London (UCL). His e-mail address is P.Barker@cs.ucl.ac.uk

be regarded as strait-jackets in areas where this need not be so.

My conversion to the belief that we had to use X.500 creatively to provide services came when I tried to explain to a UK provider of information services one of the limitations of the X.500 service: that in general one has to provide the organisation name if one wanted to find a person's entry. His response was that one didn't have to do this with the old whois[6] service, that X.500 didn't appear to represent progress, and that he couldn't sell such a service to his users.

The need to be able to find people without having to specify an organisation name also emerged from a meeting held in November 1993 to re-evaluate the policy of using X.500 for the Internet directory service; the discussions of that meeting are summarised in RFC 1588[3].

One of the goals of RFC 1588 was to update an earlier Internet directory services plan [7] and to re-assert the functions that a white pages directory service should perform. At the head of the functional requirements section, it states:

> the service should work fast when searching for people by name, even if the information regarding location or organization is vague.

Thus, the ability to search large portions of, or even the whole name-space, was seen as a fundamental requirement, although it is interesting that the Whip Internet Draft does not support this view.

Unfortunately as I noted earlier, this type of searching is either not possible, or at best impractical, with the current infrastructure of distributed X.500 servers. This lack of functionality has helped to cement the view that not only is X.500 unnecessarily heavyweight for straightforward directory service requirements, but that it also lacks crucial features for more complex tasks.

While there is some truth in the view that the public domain implementation of X.500[8] is a cumbersome package, there is no truth in the assertion that X.500 cannot be used for efficient searching of the white pages name-space. The rest of this paper is about how we can use X.500 index servers, or Index DSAs as I shall refer to them, to provide high performance look-ups of information in X.500 DSAs. A moment's reflection indicates that this type of enhancement has been applied to several favourite Internet information retrieval tools.

- FTP functionality has been enhanced by archie;
- Veronica facilitates the searching of gopher-space;
- CUSI and pages such as the Internet Resources Meta-Index help users find pages in the World-Wide Web.

Index DSAs are intended to enhance X.500's basic functionality in the same way. Before proceeding further a definition of Index DSA is required. I use the term broadly to mean a DSA that holds a collection of X.500 entries (or more typically just the names of those entries) such that they can be searched more efficiently than in the distributed DIT. An Index DSA can be characterised further:

- it will certainly require the physical gathering of entry names into a single DSA, or perhaps, a small cluster of collaborating DSAs;
- it may also require the logical gathering of entries from disparate parts of the DIT;
- it will probably require, depending on the number of entries involved, the use of database indexing technology;
- it may require small changes to the DIT held within the Index DSA, particularly of directory knowledge, in order to control the scope of searches;
- a query sent to an Index DSA may deliberately give different answers to those given to a query sent to a DSA fully integrated into the main DIT;
- Index DSAs may be able to see out into the main DIT, but will not generally be visible from the main DIT.

The perspective we shall adopt is primarily that of a service provider, considering how to provide the best possible services. We will see that Index DSAs can be used to solve several difficult service provision problems, including a searchable directory of all organisations in the world, directories for specialities, and even a unified directory of separate X.500 DITs. We will consider approaches which may not be 'pure' X.500, but that nevertheless make effective use of the information in X.500 DSAs.

We could even consider broadening our use of X.500 to, for example, act as a search engine for organisations World-Wide Web pages. Alternatively, a primarily X.500 based white pages directory could be enhanced by building an Index

DSA which contained URLs to other types of organisational directory servers.

The paper primarily takes a technical view of how we can provide useful services with X.500. However we should note that many directory service problems are not technical, but are legal or even moral issues. The legislative framework concerning making information about people available differs widely from country to country, and its stringency in some European countries has hampered development of a white pages service. Moral issues concerning an individual's right to privacy are probably even more important. While compliance with the law may be relatively straightforward judgements about the types of information that should be made available are harder to make. These issues become much more sensitive when we are trying to create services based on aggregating information from a number of directory servers, with the explicit purpose of making the information more readily searchable. It is not clear which services we will be able to provide even when we have technical solutions, be they X.500 or some alternatives. Discussions of some of the legal and privacy issues can be found in [9] and [10].

I have assumed throughout that access to information should be by X.500. However it should also be possible to access information by LDAP since LDAP closely follows the X.500 information model.

2. The Need for Index Servers

In this section, we will review why we need index servers and look at some of the services which index servers can help provide.

2.1 Global White Pages Directory

Let us first re-state one of the requirements set out in RFC 1588: "the service should work fast when searching for people by name, even if the information regarding location or organization is vague".

The vagueness of location or organisation is not a problem for centralised systems such as whois, where all the data is held within a single database and better search performance can be achieved by indexing the data within the database, or enhancing existing indexing. However, there are inherent limitations of scale with this centralised approach. While it is feasible to build very big databases, probably adequate for the entire Internet user population, data management is much harder for a centralised database, and databases remote from those who manage the information tend to become out of date and contain lots of 'dead wood'. The solution to the data management problem, and also ultimately to the scaling problem, is to have a distributed directory such as X.500.

However, whereas we can find people entries quickly in a distributed X.500 directory if we know the organisation and / or localities names, it is is very difficult to find entries for people if we cannot narrow the search space to an organisation or locality. The problem is that widening the scope of the search to include all organisations in a country, or even in the world, implies distributing the search to a large number of servers. While this type of brute force searching is just about possible in most countries in the current X.500 directory, such a solution is not realistic for a full-scale global directory.

For a variety of reasons, including X.500's supposed deficiencies as a provider of indexed services, Whois++ [11] has been put forward as the solution to the Internet's directory problems. One of the attractions offered by whois++ is that index servers are a fundamental part of the original design. Queries with vague or no organisation or locality information are sent to an index server which holds centroids from participating database servers. The index server uses these centroids to calculate which servers might be able to answer the query, thus eliminating from the search scope all those servers which do not hold information relevant to the search.

There is no reason why index servers cannot be built using X.500 technology. However, it is understandable why no-one has yet built index servers based on X.500. While the 1988 standard acknowledged the use of shadowed entries, the standard lacked a replication protocol. An interim protocol[12] was specified for use on the Internet, but this lacks important facilities such as subtree replication and incremental updates. Furthermore it was not widely adopted, only being implemented in the Quipu system. The 1993 standard includes powerful replication facilities that solve the technical problems of building Index DSAs. Issues concerned with replication are discussed further in section 5.

Before moving on to consider other services that would benefit from indexing support, we can

briefly consider what evidence there is that users need this sort of service. I have made some analysis of calls to the PARADISE pilot's central DUAservice[13]. While the default mode of querying insists on users entering an organisation name in their queries, the UFN style of querying imposes few restrictions on how users formulate their queries. Studying input of UFN queries allows us some insight into the information that users are both able to and / or think it helpful to provide when querying the directory. In 2.5% of queries, users made some attempt to wild-card the organisation name, either by entering an asterisk character or entering nothing at all (between a pair of field-separating commas) for organisation name, although users were given no indication that this was a valid query format. Another measure of the likely proportion of searches where no organisation name is specified can be attained by looking at how often users select the interface's 'power search' mode. This mode allows users to omit organisation name details and to search a country-wide name-space. From data on searches made from UCL, 8.4% of all searches were power searches. My guess is that the true percentage of searches where the organisation is unknown lies between these two values, and is probably nearer 2.5% as I suspect many of these power searches were made by users experimenting with the system.

2.2 World-Wide Directory of Organisations

While there has been no attempt so far to build directory-wide indexes of the one to two million people entries, no-one has yet even built a satisfactory directory of the four thousand organisation entries in the X.500 directory.

The X.500 directory allows considerable flexibility in where organisation entries can be placed in the DIT, as illustrated in Figure 1. The vast majority of organisation entries have, so far, been placed under their respective country entries. A few international organisations have been placed immediately beneath the root node. A substantial number of entries, usually for smaller organisations, have been placed beneath state or locality entries; countries using this structure include the US and Australia. There are also a few organisation entries held beneath parent organisation entries. While this diversity of structuring may help with management of the directory name-space, it has unfortunate consequences for the searchability of the directory. Whereas the main problem with building Index DSAs for people entries was the lack of a suitable replication protocol, this is less of a problem here as even the interim Internet replication protocol is adequate. Furthermore, some DSA managers replicate the top levels of the DIT by FTP. The result is that many DSA managers already choose

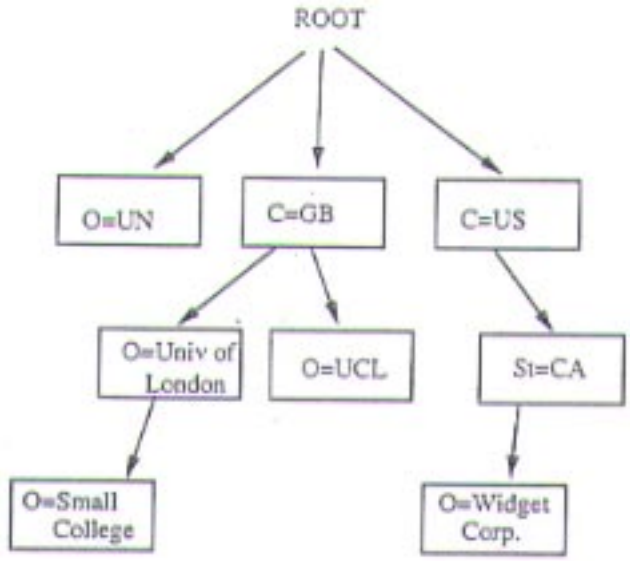

Figure 1: Organisations at different points in DIT hierarchy

to shadow the vast majority of organisation entries in their DSAs. The problem in this case is that the protocol does not allow us to specify a simple search that includes in its scope all the organisation entries, but at the same time excludes the millions of leaf entries.

A further aspect to the problem is that current DIT contains only a small percentage of organisations. A service provider would have to add entries for many organisations not represented in the directory, in order to provide customers with a directory service for which they might be prepared to pay.

Several solutions are considered in this paper, including the conventional subtree of aliases approach.

2.3 Directories of Specialists

Often users will want a global or country-wide scope for their queries but will be want to restrict their searches to a particular discipline, such as librarians or computer scientists. As before it is not possible with a single query, or even a small number of queries, to search the appropriate parts of the directory.

It is not a straightforward matter to automate the building of such views of the directory as there are so many names used to identify, for example, computing departments: e.g., Computer Science, Computing Science, Computer Systems, Computing Centre, Information Science, Information Systems, and many more.

The diversity of structure and naming is illustrated in Figure 2. One can envisage a service provider adding value to the directory by organising the information into searchable structures.

2.4 A Unified Directory of Separate X.500 DITs

We currently have the luxury of being able to talk about the DIT when considering the freely accessible directory on the Internet. This used to be referred to as the PARADISE directory[14], and is now managed by DANTE as the NameFLOW-Paradise directory service. However, there are already many private DITs being used both experimentally and in service. In some cases, these other DITs are deliberately private, possibly holding commercially sensitive information; these directories are unlikely ever to be fully connected to a global public directory. In other cases, separate DITs have been forced upon service providers,

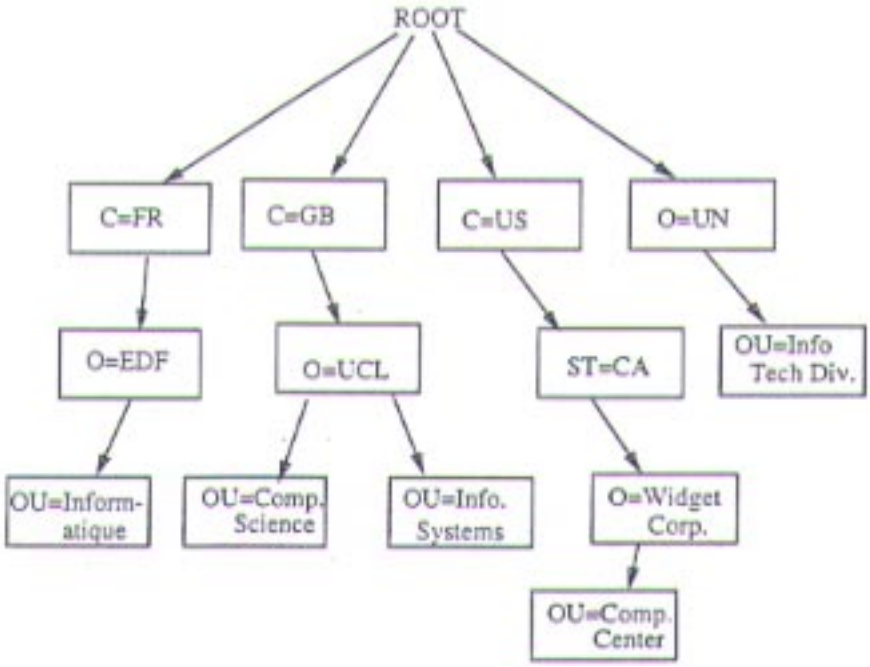

Figure 2: Organisational units widely distributed throughout DIT

because they have no way of representing the information they need in their DIT alongside the information in the NameFLOW-Paradise directory. This problem arose twice during the PARADISE project, affecting the British Ministry of Agriculture, Fisheries and Food (MAFF) and the Y-NET project, who both maintained private X.500 email directories.

The problem is illustrated in Figure 3. There are two parts to the problem. First there is a clash in the GB name-space, where the registration authorities for two DITs have included different but overlapping sets of organisation entries. There is also a problem lower down the DIT where the two DITs may hold different, overlapping sets of entries for each organisation, and those entries may hold different attributes.

Although the advantages to users of seeing a unified DIT are relatively small while there are just a few small, private DITs, the situation will be different when large service providers such as the Public Network Operators (PNOs) offer X.500 services with data which complements that already offered by the NameFLOW-Paradise directory service.

Some attempt has been made to tackle this problem by the North American Directory Forum (NADF) [15]. They have produced a solution based on building a common DIT, which places entries in the DIT in their natural place. The service providers build their own DITs, naming their entries beneath administrative domains. Service providers hold partial entries for people and organisations: for example, post offices hold postal address information while telephone companies hold telephone and fax numbers. The common DIT exists as a naming tree with naming links, effectively distinguished name pointers, to the entries in the separate DITs. Although there is relatively little experience with the system yet, some efficiency problems with the scheme have been noted, as several DSAs may have to be contacted to read one entry. The NADF naming links do not show which attribute are held. A proposal which solves this problem has been circulated by Chadwick; in this scheme, attribute types are now associated with naming links. There are also doubts about how well the solution will scale, with the vast amount of link information that has to be maintained. For the moment, despite the doubts about the technical merits of the NADF solution, ISO have accepted the approach as the way of dealing with multiple DITs. It is useful to bear in mind the nature of the NADF when considering their solution. A primary consideration for the NADF was to produce a model that allowed multiple service providers to collaborate yet compete. These considerations have inevitably led to a model where querying

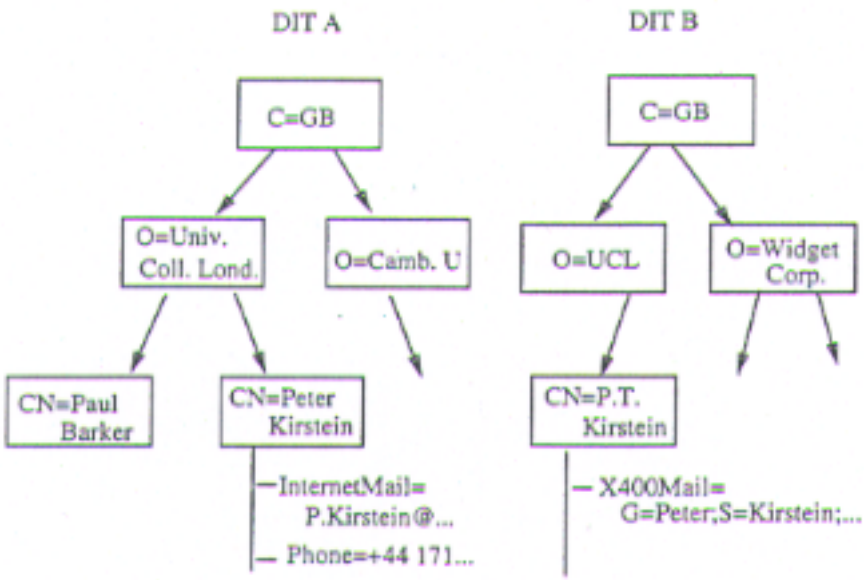

Figure 3: Parallel and over-lapping DITs

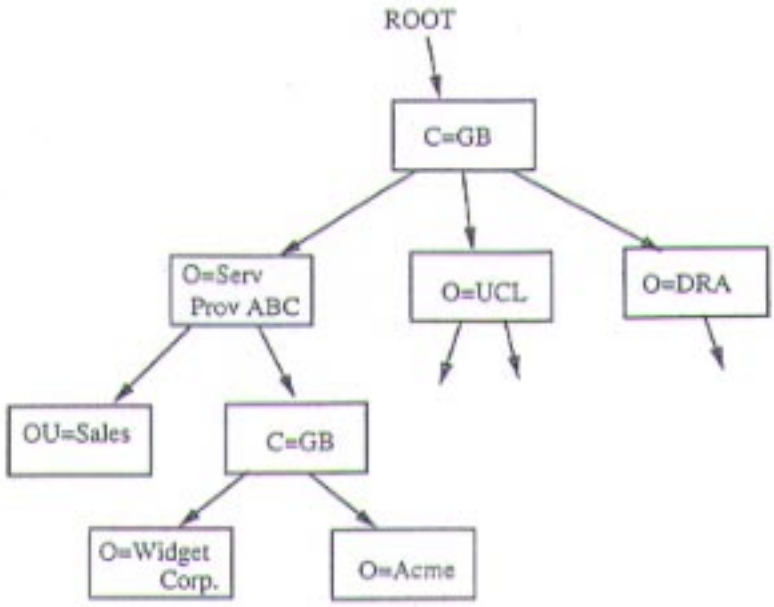

Figure 4: Additional Organisation Entries Beneath Service Provider Entry

efficiency is secondary to allowing service providers to retain control over their information.

2.5 Srvice Provider Adding Entries into Main DIT

A service provider must try and offer some coherent directory if it is to attract customers to its service. This is not possible with the current DIT, which is patchy and reflects the fact that the DIT has largely grown organically; most of the data in the directory is for sites which run their own X.500 DSAs (although the majority of organisation entries have been bulk-loaded). On the other hand, customers want services which provide them with details of most or all the organisations in which they are interested. There is a mismatch between the information demanded and that supplied, a problem which Colin Robbins refers to in terms of cost-benefit analysis in [16].

One way in which service providers could respond to this would be to register additional entries in the directory. The problem is where to place these entries in the main DIT since registration authorities are unlikely to allow arbitrary service providers to place entries which are in some sense provisional in the natural place in the DIT.

One approach would be for service providers to register these additional entries under special subtrees beneath their own organisation nodes, as illustrated in Figure 4. While this solves the registration problem, it creates a searching problem, since DUAs now have to search the main DIT as well as the service provider's special subtree.

2.6 Summary of Requirements

Before examining some possible techniques for providing indexed services with X.500, I will set out some principles which will influence the way that these services are provided.

X.500 as a tool: The approach is service provider oriented, where the service provider is trying to provide customers with a useful directory. The service provider will be using the X.500 infrastructure as a building block for a directory service, and will be trying to use the directory data as effectively as possible rather than following some prescribed model.

High performance: The aim of any services described here is to provide high performance access to data which is inherently distributed throughout the DIT and mastered on a number of DSAs. An indexed service should allow a user to find their required data with access to at most three or four DSAs.

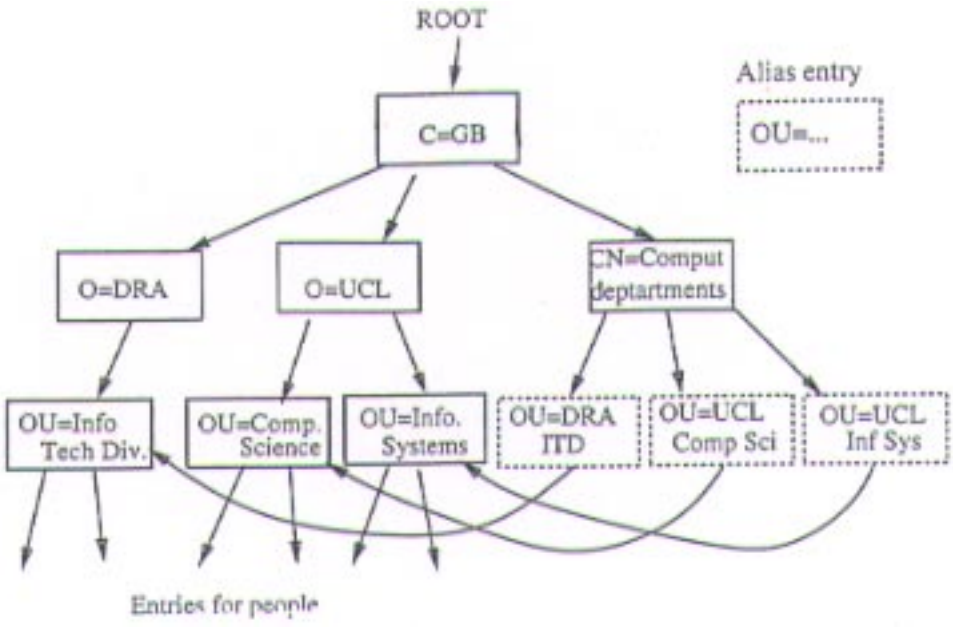

Figure 5: Aliases for a directory of computer personnel

Names only or partial entries: The main goal of an index server is to provide rapidly searchable indexes of information, which can then be used to provide pointers to the full entries in the DIT. For a white pages service this means indexing the naming attributes. In some cases, it may be prudent for performance reasons to hold a limited set of other attributes in the index server. For a white pages service, the index server might also hold email addresses and telephone numbers.

Different views of DIT: The assumption that the X.500 DIT should look the same from all DSAs, except for the proviso that shadow copies may possibly be out-of-date, is dropped. Service providers have an active interest in their DIT looking different to (and better than) other service providers' DITs.

Extra entries: Service providers will want to add extra entries to their DIT, in order to provide a useful set of information which can be used as the basis for a service.

Annotation of existing entries: Service providers may wish to annotate entries that are mastered by other organisations. This is not possible with conventional X.500 as updates are directed to the master copy of entries. An example service is one where an organisation might wish to annotate another organisation's entry with attributes which in some way describe the relationship between the two organisations. Possible attributes in this category include those for train or bus timetable information, or more plausibly for URLs to repositories of that information. Another use would be storing directory response time information as described in the QOS paper [17].

3. Aliasing and Replication

The conventional approach to making the distributed directory more searchable is to use a mixture of aliasing and replication. They are complementary techniques.

Aliasing is used to gather together disparate parts of the DIT to present an alternative view of the information. Aliasing gives extra names to entries so that those entries may be found more readily in the directory. Figure 5 shows how a set of aliases (the entries with dotted boxes) to computing departments could be used to provide a directory of computer specialists. The user can now search a number of organisations' computer personnel by making a single query beneath the "cn=Computer Personnel" node. It is important to note though that aliasing is merely the logical gathering of information and that the entries for people in those computing departments would still typically be mastered on a number of separate DSAs. Searching all computer personnel would thus require the searching of many DSAs.

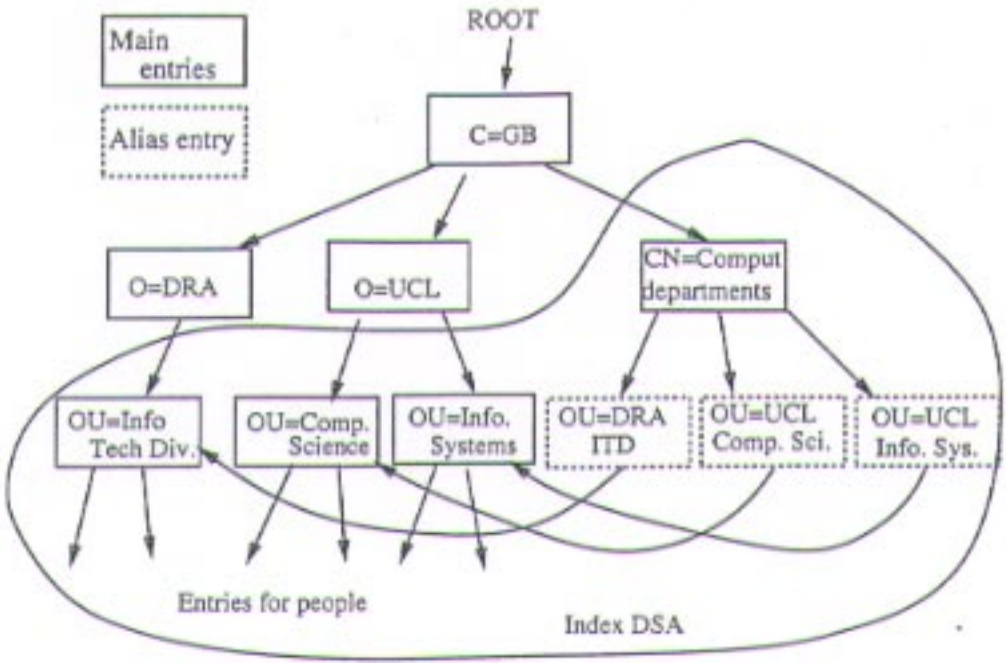

Figure 6: Computer personnel data replicated into Index DSA

Replication of entries into a single DSA (or small group of DSAs) is also required to improve search performance and to reduce the load on the distributed system. If all the nodes beneath the various computing nodes are replicated into a single DSA, along with all the alias nodes, the computing personnel name-space can now be searched within a single DSA. Figure 6 shows the data held by this DSA. Crucially, from a performance point of view, the data can also be indexed as it now resides within a single server.

3.1 How Useful is the Technique of Aliasing and Replication?

How generally applicable are these aliasing and replication techniques for the service provision problems outlined earlier?

The case for replication is the simpler to answer. Services such as archie [18] have shown that there is no substitute for gathering the information that has to be searched into a single database and using indexing techniques to allow fast searching of even very large databases. With archie, the filenames are replicated in large databases. The user searches these databases and, if the search is successful, retrieves the files from their original filestore. The model should be exactly the same for X.500 indexed services. For a WP directory of people, we can envisage replicating common names, surnames and possibly userids as these are the attributes that users will use to construct their searches. The searches will be directed to special DSAs, which will identify the distinguished name and location of the full entry or entries.

The case for aliasing is less easy to answer. In some cases, such as the directory of computer specialists, it seems to offer precisely the right functionality. Even in these cases, there are some characteristics of the technique which we should not forget.

- Aliases have to be maintained. There is no automatic system based on back pointers which ensures that the deletion of an entry leads to the corresponding deletion of any corresponding alias entries.
- The aliases are pointers to an arbitrary collection of data. Even if the aliases are systematically maintained, there is no guarantee that the set of aliases is in any sense complete.
- The aliases are grouped under nodes at some arbitrary place in the DIT. Users and / or their user interfaces have to know about these special places if the aliasing is to be effective.

However, in other cases such as the directory of all organisations, even though we could use aliasing to make the directory more searchable, it seems an inappropriate technique. For the world

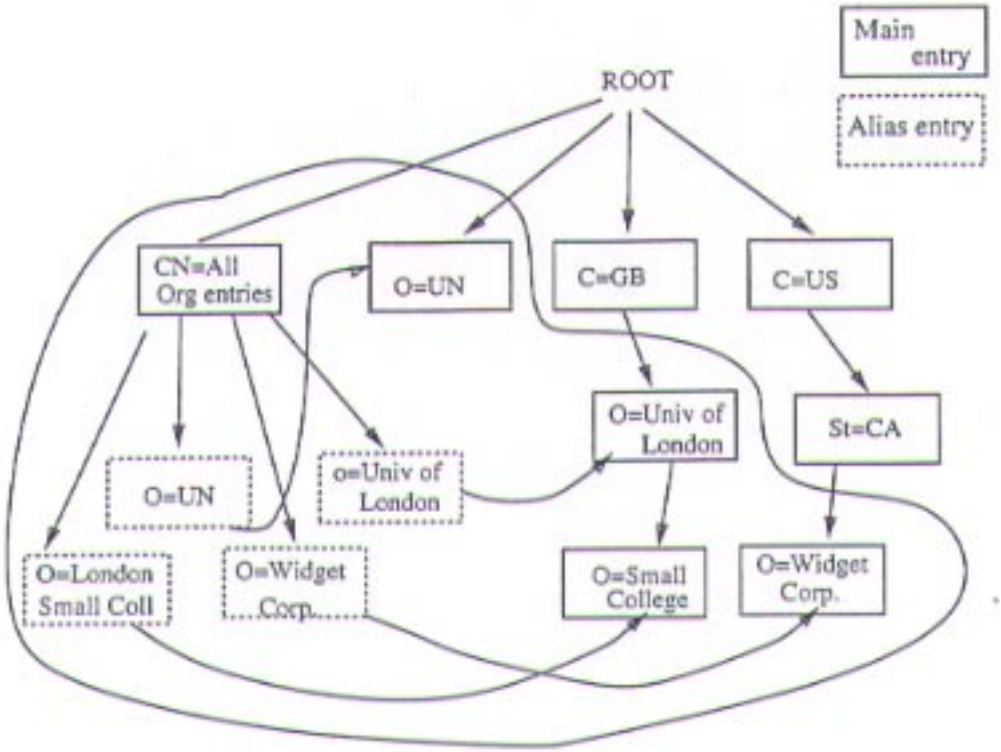

Figure 7: Aliases to all organisations

directory of organisations, the scope of the search is the vast majority of the nodes near the root of the DIT, rather than scattered nodes throughout the DIT. The problem is how to restrict the scope of the search from the root of the DIT as far down as the organisation entries but no further. A subtree search for organisation entries from the root of the DIT puts the whole DIT in scope, and is thus prohibitive. The alternative of single level searches is unworkable because, as was described earlier, there are simply so many places to search.

A solution using aliases would be to create a set of aliases to all organisation entries under a special node. The search could now be expressed as a single level search, which avoids the scope problems associated with subtree searching. An example structure is shown in Figure 7. However, as with the previous aliasing example, the approach requires users to know special places in the DIT where to root their searches. This convolution of queries, with contrived search bases, seems an unnatural approach.

3.2 Searching Entries Added by Service Providers

Earlier we noted that service providers might want to add entries to the directory in order to provide a better service. The problem was that registration issues would be likely to force the service provide to place these extra entries beneath their own organisation entry. This complicates searching by requiring DUAs to now search in the main DIT, and also in the service provider's special subtree. One solution for the service provider would be to create aliases for all the required entries in the main DIT and place these alongside the added entries.

This offers a solution to the search problem, at a cost of having to base searches at an unnatural place in the directory. In Figure 8, for example, searches would have to have a base object of "c=GB; o=serv Prov ABC; c=GB" rather than simply c=GB.

3.3 A Question on Performance

So far I have raised questions about the appropriateness of using aliases to some of these service provision problems. These questions have mostly concerned architectural niceties and the need to maintain a spaghetti of aliases. However, the recent PARADISE OIFP report [19] also questioned the impact of aliases on performance.

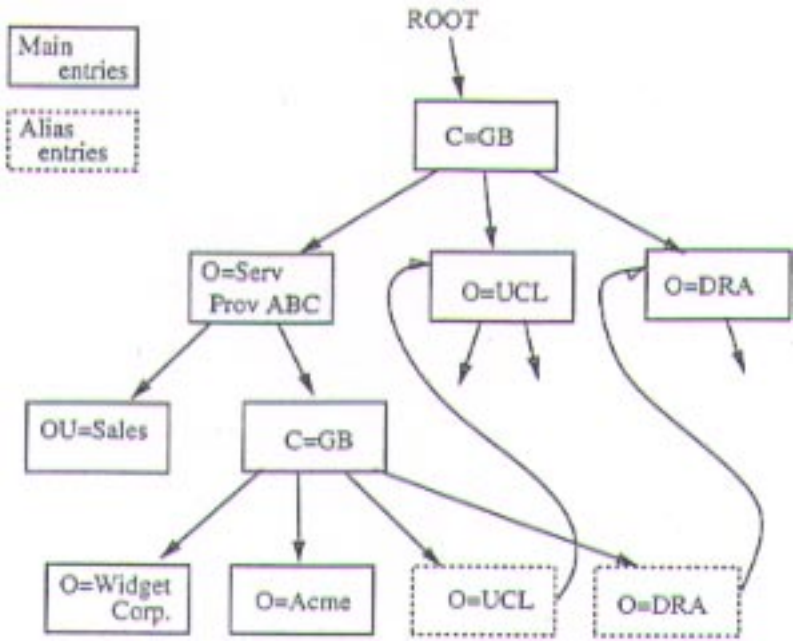

Figure 8: Mixture of Ordinary and Alias Entries

The report authors noted that one supplier estimated that the possibility of having to handle aliases halved the performance of X.500 systems even if aliases were not involved in a given query, and that performance degradation was much worse if aliases were used.

I have spoken to some other X.500 implementors and designers who dispute this comment and say that they believe that aliases offer only a slight overhead.

To test this, I have made some simple preliminary tests using a Quipu 8.0 system. Searching 4000 entries dereferenced through aliases was at least three times slower than searching the same entries directly; both searches were made within a single DSA.

4. An Alternative Approach

The focus of this paper is to consider an alternative approach to aliasing. The proposed solution also solves the scope problems that exist with some of the services. The solution also seems to fit in well with a model where various service providers can compete by offering different views of the directory. The main points of the proposal are:

- We will regard the data in the X.500 directory as an infrastructure on which to build services.
- Special services will be built by extracting (replicating) the required data into special-purpose, high-performance, Index DSAs.
- The data in the Index DSAs will generally be just the searchable attributes from the entries, as well as the distinguished names of the entries.
- Just as a set of alias entries offers a selected view of some / all of the data in selected parts of the DIT, Index DSAs will offer a similar selected view. The basic rule is that if an entry is not in the Index DSA, then it does not exist in the directory as far as that service is concerned.
- Whereas aliasing techniques rely on creating special points in the DIT under which to gather their entries, with Index DSAs the main DIT is searched, with the scope of the search being pruned to include only those entries in the Index DSA. Thus searches are based on more 'natural' names.
- Since we are now regarding the data from the main DIT as merely infrastructure, we might now choose to restructure this data in forms other than the original DIT if such structures gave better search characteristics. One possible example is that it might be useful to flatten the naming hierarchy.

- The most important capability of an Index DSA is that it can pass back a distinguished name and DSA reference back to the DUA, so that the DUA can retrieve the full entry from the main DIT.

- Index DSAs should be regarded as being no more than loosely connected to the main DIT, since they break the rule that the DIT should look the same irrespective of which DSA answers the query. It may be that Index DSAs can see out into the main DIT, but they should not be seen in the main DIT as shadowing any part of the DIT. Entries for Index DSAs should be visible in the DIT with an indication of what services they offer.

- It is not clear to me at the moment how feasible it will be to build and run Index DSAs solely using standard features provided by the 1993 X.500 specification, along with any implementation specific optimisations and use of appropriate service controls. The key area is how to prune the scope of subtree searches, so that they can search all the necessary entries, yet not be obliged to return continuation references for those parts of the DIT that have not been searched, but which do not contain entries of interest. Possible approaches include:
  - Implementations could be optimised to hold information of which object classes are held in any subtree. This information could be propagated up the DIT in a manner similar to that proposed in the Internet Draft on counting entries in the DIT[20].
  - Implementations may be able to replicate the data normally using X.500 replication protocols, but then allow us to modify the knowledge references such that a DSA believes there are no other parts of the DIT that need to be searched.
  - It may be possible use the service control localScope for a DUA to indicate that it does not want searches propagated beyond the access point DSA, and that it is not interested in continuation references for parts of the DIT that have not been searched. Possibly a noContinuationReferences service control is required?
  - It may be possible for a DSA to hold the required entries and to be able to determine from replication agreements that it holds shadow copies of all parts of the DIT that it needs to search.

5. Creating an Index DSA with the Required Data

We noted briefly earlier that one reason why no-one has built Index servers using implementations of the 1988 standard is that the original standard did not include a replication protocol. Most implementors of 1988 systems defined proprietary replication protocols and one was defined as an interim Internet standard in RFC 1276. Unfortunately this protocol is somewhat simple-minded and lacks two very important features. First, there is no facility for subtree replication. Second, it is not an incremental protocol. The change of one byte in a set of entries means that all siblings have to be copied. This is a very expensive operation when some of the entries contain images and audio attributes.

The 1993 standard, in contrast, has powerful facilities, which include:

- single entry through to subtree replication;
- shadowing of: full entries; selected attributes only, based on inclusion or exclusion lists; or name only;
- shadowing of entries selected by object class;
- incremental updates.

Unfortunately, we will not have X.500 systems with 1993 replication mechanisms widely available for (and I'm guessing) two years. In the meantime, in order to gain some experience with the issues of appropriate DIT structures, we must build our Index DSAs by ad hoc means. In practice we can do this by using a mixture of RFC 1276 replication plus some use of search. Using search has some advantages as it allows us to replicate just those attributes we want. We can even do incremental replication by using the lastModifiedTime attribute as a filter. There are problems as well: administrative limits may make it hard to retrieve all the entries we require; search based replication techniques do not automatically detect the deletion of entries, although in practice few entries are deleted at the top levels of the DIT.

Andrew Powell (of the University of Bath) and I have some experience of using these ad hoc techniques, which we have used in building an Index DSA of world-wide organisations. This DSA is accessible by:

```
dish -u -c Internet=138.38.32.45+17005
```

The DSA supports subtree searches from the root

and country level; it returns matching organisation names and entries only. The intention is that follow-up queries (for entries subordinate to the organisation) can then be made in the main DIT, using the organisation entry as the base object of further operations. Although the DSA is freely available, it is a guest on a host that runs other advertised services, and thus could not be used for a serious trial in its current form.

6. Index DSAs and scale

It was not clear to me whether Index DSAs solved the searchability problem of a large-scale WP directory at the cost of returning an unmanageable number of results or references to a user. I have investigated this problem with an experiment using real user queries and the UK DIT. The results are presented in [21].

The main results were:

- The cost of a typical DE power search of the UK is currently about 55000 bytes of transport data, of which over half is due to connection establishment and disestablishment.
- Surname only queries return very large numbers of results (average of 100+) even for a portion of the DIT of a quarter of a million entries. Initial plus surname queries returned on average 7 entries.
- An DSA of 10000 entries has a 50% chance of matching an arbitrary surname only query; this means that the search space of DSAs could not be pruned effectively for surname only queries given large DSAs. The equivalent figure for initial plus surname queries was a 20% chance of matching.
- If Index DSAs were organised hierarchically to cooperatively index a large name-space, the increase in the number of servers that have to be contacted is modest.

7. Security Issues

The issue of access to information is complicated by the use of index servers. Without them, access may now be granted or denied to the main entries holding the attribute information. Index servers create new information by their aggregation of entries into a more searchable form. These indexes may now need protecting, as service providers will presumably deny access to their services for users other than their customers.

In a preliminary study I came to the following conclusions about security issues:

- Access would be controlled by an authorisation policy governing use of the Index DSA service, rather than access control on individual or groups of entries.
- A user would generally bind directly to an Index DSA for a given special service; the Index DSA would thus do its own authentication, rather than trust another DSA to have authenticated the user correctly.
- A service provider might, for reasons of simplifying authentication, prefer to issue its users with special identities, rather than allowing access to users identified with their normal DIT personas.

Modify access will not be allowed on information held in Index DSAs. There two main reasons. First, an Index DSA only holds a shadow copy of information held elsewhere; updates must be performed on master copies. Second, Index DSAs are fundamentally to help users querying a large and diverse name-space; update operations will presumably be done by people who know the distinguished name(s) of the entr(ies) they wish to modify.

8. Possible Future Work

The following is a list of ideas or further work.

1. Build an Index DSA of all organisations in the directory, and offer it as a trial service on the Internet. This could be done by building on the work described in Section 5.

2. Modify the DE user interface (or any other) to make use of this Index DSA.

3. Extend an organisational Index DSA to be a search engine for World-Wide Web sites.

4. Extend the organisational Index DSA to hold pointers to alternative, non-X.500 directory services.

5. Study the extent to which Index DSAs can be built within the 1993 standard, and determine what, if any, extra protocol would be useful.

References

1 The Directory - Overview of Concepts, Models, and Service, - ISO 9594 and CCITT X.500

2 S Kille, E Huizer, V Cerf, R Hobby, S Kent. (1993) A Strategic Plan for Deploying an Internet X.500 Directory Service, Request for Comments RFC1430 URL= <ftp://ds.internic.net/rfc/rfc1430.txt >

3 J Postel, C Anderson. (1994) White Pages Meeting Report, Request for Comments RFC1588 URL= <ftp://ds.internic.net/rfc/rfc1588.txt >
4 T. Genovese. (1994) A Specification for the Simple Internet White Pages Service, Internet Draft URL= <ftp://ds.internic.net/internet-drafts/draft-ietf-whip-iwps-requirements-01.txt >
5 A J Waugh (1994) X.500 and the 1993 Standard, Technical Report TR-SA-94-03, CSIRO Division of Technology
6 E. Feinler, K. Harrenstien, M. Stahl (1985) NICNAME/WHOIS, Request for Comments RFC954 URL= <ftp://ds.internic.net/rfc/rfc954.txt >
7 K Sollins (1989) Plan for Internet directory services, Request for Comments RFC1107 URL= <ftp://ds.internic.net/rfc/rfc1107.txt >
8 Colin Robbins, Steve Hardcastle-Kille (1991) The ISO Development Environment: User'sManual (version 7.0), July 1991
9 E Jeunink, E Huizer (1994) Directory Services and Privacy Issues, presented at JENC 1994
10 J M Hill, (1992) The X.500 Directory Services: a discussion of the concerns raised by the existence of a global directory, Vol.2, No. 1, Electronic Networking (Spring 1992)
11 C Weider, P Fältström. (1994) WHOIS++, ConneXions, Vol 8, No 12, December 1994
12 Steve Hardcastle-Kille (1991) Replication and Distributed Operations Extensions to Provide an Internet Directory using X.500, Request for Comments RFC1276 URL= <ftp://ds.internic.net/rfc/rfc1276.txt >
13 Paul Barker (1995) An Analysis of User Input to an X.500 White Pages Directory Service, in Transactions on Networking, vol 3, no. 2, pp 112-125, April 1995
14 D. Goodman (1991) PARADISE: the COSINE X.500 pilot service, pp. 111-114 in Computer Networks and ISDN Systems 23, Published by North Holland.
15 NADF (1993) NADF Standing Documents, SD1-SD12, EMA 1993
16 Colin Robbins. (1995) Monitoring Quality of Service in a Global Information Service, Draft paper for Dante.
17 P. Barker (1994) Providing the X.500 Directory User with QOS Information, pp. 28-37 in Computer Communication Review (ACM SIGCOMM), July 1994, Vol 24, No. 3
18 Alan Emtage, Peter Deutsch (1992) archie - An Electronic Directory Service for the Internet. Proceedings of Usenix, Winter 1992.
19 P-A Pays, B Koechlin (1994) Operational Interworking Forum and Platform Final Technical Report, a report published by the VALUE PARADISE project URL= <ftp://ftp.nameflow.dante.net/paradise/oifpfinal.txt >
20 Steve Kille (1992) Counting the Directory Information Tree, OSI-DS 30, April 1992, URL= <ftp://cs.ucl.ac.uk/osi-ds/osi-ds-30-00.txt >
21 P Barker (1995) X.500 Index DSAs and Scaling Issues for an Indexed White Pages Directory Service, published in proceedings of JENC 6, Tel Aviv, May 15-18, 1995