- QUIPU always supplies the optional chaining argument "target" even if the base object in the DAP arguments is the same.

- The object class "without an assigned object identifier" is not recognised unless the "alias" object class is also present.

- Non Specific Subordinate References are never followed by a QUIPU DSA, but they are passed on correctly to the client if generated.

QUIPU is intended to provide an environment for experimentation with standardized directory services. It is used by the ISODE for identification of the location of OSI applications (including QUIPU) and for provision of white and yellow page services. The directory abstract service and DSA abstract service defined in [CCI88, ISO88] and their associated protocols are supported. For replication the Internet DSP protocol is used.

Major aspects of the QUIPU implementation are:

- Use of memory structures to provide fast access.

- Activity scheduling within the DSA to allow for multiple accesses.

- General and flexible searching capabilities.

- Extensions to provide access control.

- External schema management

- Use of the directory to control distributed operations.

The current implementation provides a DSA, and a procedural interface to the directory abstract service, which will enable other applications to use the directory. There is also a directory shell interface — DISH. This provides full access to the directory abstract service, using the procedural interface. standard distributed operations are used with both referrals and chaining (using the directory system protocol) provided.

A full discussion of the design issues relating to QUIPU can be found in [Kil89b].

2.2 Pronouncing QUIPU

The name of the INCA directory is QUIPU. The official pronunciation of QUIPU takes two syllables: *kwip-ooo*.

2.3 Why QUIPU

QUIPU was originally developed as a part of the INCA project. The Incas of Peru did not have writing. Instead, they stored information on strings, carefully knotted in a specific manner and with coloured thread, and attached to a larger rope. These devices were known as *Quipus*. The encoding was obscure, and could only be read by selected trained people: the *Quipucamayocs*. The Quipu was a key component of Inca society, as it contained information about property and locations throughout the extensive Inca empire.

2.4 Objectives

2.4.1 General Aims

QUIPU has a number of general aims:

- To produce an implementation which follows the emerging OSI directory standards.
- Flexibility to enable the system to be used for experimentation and research into problems relating to directory service.
- Investigation of distribution and replication.
- Pilot experimental usage.

2.4.2 Technical Goals

The major goals of the QUIPU directory service are:

- Full support of the directory access protocol, directory system protocol, and distributed operations, as defined in [CCI88].

- Support of the majority of the service elements specified in [CCI88].
- Ability for interworking with other directory implementations, including use of referrals and chaining.
- Very full searching and matching capabilities, beyond the minimum required by [CCI88].

The following are not goals:

- In practice, the memory based approach has led to a quite fast lookup and searching.
- The ability to handle very large volumes of data (e.g., greater than 100 MB or 1 Million entries per DSA) is not a requirement.
- Substantial data robustness is not required: there is no need to employ complex data backup techniques.
- Use (as opposed to provision) of authentication services.

2.5 Roadmap

This manual is split into 6 parts. You are reading Part I, which is a general introduction. Part II describes a set of user interfaces (DUAs) developed as part of QUIPU. Part III, an administrators guide, describes how to set up both the DUAs introduced in Part II, and how the install and manage a QUIPU directory system agent (DSA). Part IV is a programmers guide which discusses a procedural interface to the directory for those of you who want to write your own DUAs. Finally, Part V contains Appendices.

2.6 QUIPU Support Address

If you have any problem installing QUIPU, following the documentation or any other QUIPU-related problems, then there are two discussion lists.

Comments concerning the operation of QUIPU should be addressed to the QUIPU support address:

Internet Mailbox:	`quipu-support@cs.ucl.ac.uk`
Janet Mailbox:	`quipu-support@uk.ac.ucl.cs`
X.400 Mailbox:	`surname = quipu-support`
	`ou = cs`
	`Org = UCL`
	`PRMD = UK.AC`
	`ADMD = Gold 400`
	`C = GB`

Or, you could look up the mailbox attribute of

```
c=GB
o=University College London
ou=Computer Science
cn=QUIPU-Support
```

in the directory.

There is also a discussion list for a general discussion of topics related to QUIPU; the address is as above, but with "quipu-support" replaced by just "quipu" (e.g., quipu@cs.ucl.ac.uk). We suggest that everybody who is intending to run QUIPU should be on this list, as this will be used to keep you informed of what is happening. Details of updates will also be sent to this list.

If you would like to be added to the `quipu` discussion list, send a message to "quipu-request" (e.g., quipu-request@cs.ucl.ac.uk).

2.7 Acknowledgements

QUIPU was developed at the Department of Computer Science at University College London, under the ægis of the INCA (Integrated Network Communication Architecture) project, which is project 395 of ESPRIT (European Strategic Programme for Research into Information Technology). The partners of INCA (GEC plc, Olivetti, Nixdorf AG, and Modcomp GmbH) are acknowledged for releasing this software into the public domain.

Continued funding of QUIPU as openly available software is provided by the Joint Network Team (JNT).

QUIPU 6.0 was implemented primarily by Colin Robbins and Alan Turland, with considerable help from Marshall Rose of Performance Systems International. Mike Roe implemented the authentication code, and the T.61 string handling code.

After QUIPU 6.0, the core development has been from Colin Robbins. Tim Howes of University of Michigan has kindly donated the very valuable "TURBO" options to QUIPU. Alan Turland has added the asynchronous DUA interface.

Chris Moore of the Wollongong Group helped considerably in the early development of QUIPU, and integration with ISODE. Simon Walton of University College London, also provided much help in integrating the software with ISODE.

Steve Titcombe of University College London did much of the early work on DISH, and is now working on the management tools.

Paul Sharpe, of GEC Hirst Research Laboratories put considerable effort into the early development of SD. This work was continued at Brunel University by Andrew Findlay and Damanjit Mahl, who have also developed the POD interface.

Paul Barker of University College London has designed and developed the DE directory interface, funded by the COSINE project. Paul also developed the DSA log processing scripts and has provided very valuable feedback throughout the project.

Whilst at UCL on secondment from CSIRO, Andrew Worsley put considerable effort into enhancing Pepsy to provide all the extra features the QUIPU needed to be able utilise the tool.

George Michaelson of the University of Queensland, Julian P. Onions of X-Tel Services Ltd, Andrew Findlay of Brunel University, Gier Pederson from the University of Oslo, Petri Jokela of Telecom in Finland, Juha Heinänen of the Tampere University of Technology, Peter Yee from NASA and Piete Brooks of Cambridge University, have all run various test versions of the system, and provided much useful feedback.

Part II

User's Guide

Chapter 3

The OSI Directory

This chapter is designed as a quick introduction to the model of X.500 directories, to provide the reader with sufficient information to be able to understand the following chapters. For full coverage of X.500 readers should consult [Ros91].

3.1 The Model

The OSI directory is intended to support human user querying, allowing users to find, *inter alia*, telephone and address information of organisations and other users.

It is also intended to support electronic communication such as message handling systems and file transfer. The directory provides name to address mapping to support, for example, OSI presentation address look-ups. Message handling systems will be provided with support for user-friendly naming, security and distribution lists.

In essence the directory is a database with certain key characteristics.

- The directory is intended to be very large and highly distributed. It is anticipated that the directory will be distributed largely on an organisational basis.

- The directory is hierarchically structured, the entries being arranged in the form of a tree called the directory information tree (DIT). An example DIT is shown in Figure 3.1. Entries near the root of the

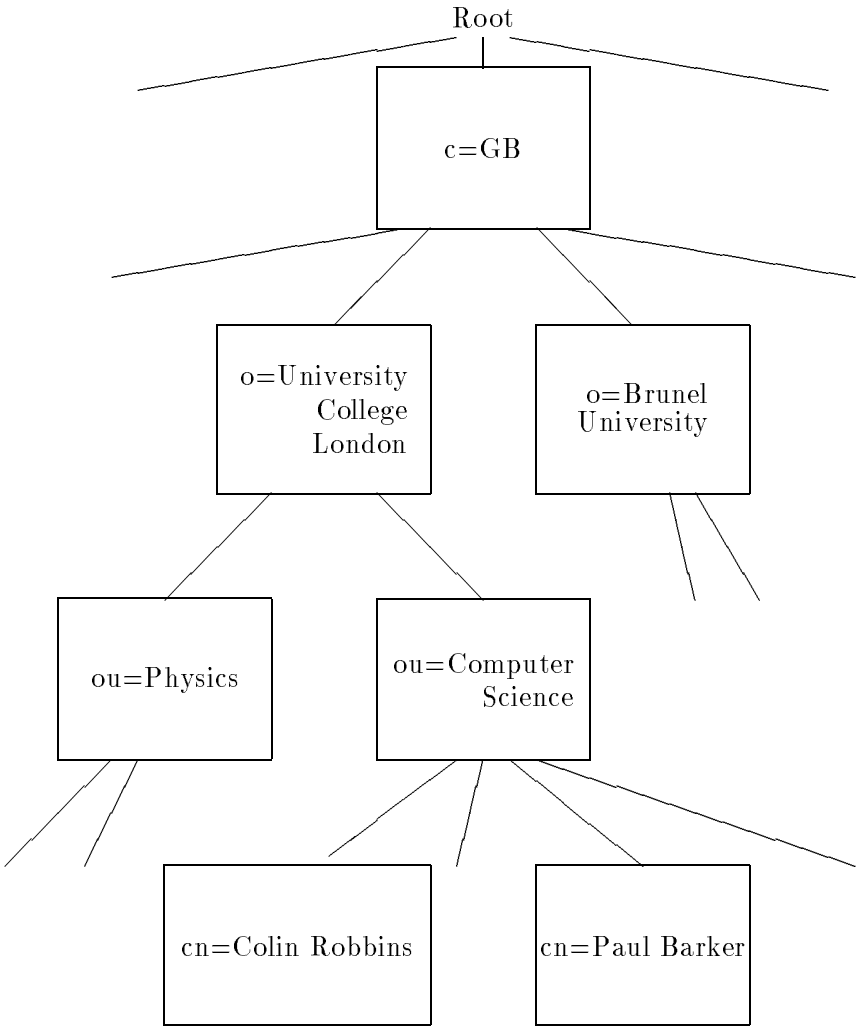

Figure 3.1: Example DIT