

Survey of previous work on directory schema registry related technologies and existing LDAP Schema

TERENA Project Directory Schema Registry, Deliverable B

Peter Gietz, DAASI International Ltd., peter.gietz@daasi.de

Version 0.9, 31.1.2003

1. Status of this document

This is deliverable B of the TERENA project Directory Schema Registry, which is co-funded by TERENA, JISC (Joint Information Systems Committee, UK), REDIRIS (Spanish National Research Network), CESNET (Czech National Research Network), POZMAN SUPERCOMPUTING (Poznan Supercomputing and Networking Center, Poland) and DAASI International and performed by DAASI International. Together with four other deliverables [RegSchema], [RegPolicy], [RegArchitecture], [RegBusiness] and the bibliography [RegBib] it forms the documentation of this project.

This version 0.9 reflects the status in January 2003 and is the first public version. A final version 1 will be published after all other project documents have been completed.

Some parts of this text, which are only indirectly related to the aims of the project, are only sketched to put the work of this project into a broader context that points to a possible future broadening of the scope of the registry after the end of this project. A version 2 of this document is planned for the time after the end of the project. For this future version comments and additions to the current document are welcomed. Please send them to the email address of the author.

Table of Contents

1. Status of this document	1
2. Introduction	1
3. Definitions	2
4. Existing LDAP schema standards	17
5. Prior approaches to schema registry problem space	21
6. LDAP schema standardisation and a new schema registry	36
7. Pointer to References	37
A. Glossary of used acronyms and technical terms.....	37
B. Detailed table of contents	38

2. Introduction

A vital part in the design and definition of directory services is the definition of data models or schemata, which in the case of the Lightweight Directory Access Protocol (LDAP) consist of the specification of object classes, attribute types, attribute syntaxes, matching rules, etc. These define information objects that map relevant aspects of material things (e.g. computers), living beings (e.g. persons) and immaterial things (e.g. policy) that together form "reality" as perceived (or constructed) by mankind.

Due to the fact that there is no reliable and easily searchable registry where one can find the big amount of already defined schemata, the danger arises that different schemata are used for the same kind of applications which leads to incompatibilities. This problem of effort doubling exists both in the commercial realm as well as in the academic community. As another result of the current situation duplicate schema names can be used for differently defined structures, which might confuse applications that use such names instead of the unique numeric Object Identifiers (OIDs) to identify schema elements. Vice versa the same schema element identified by an OID

can be named with different alphanumeric so called object identifier descriptors, e.g. "commonName" and "cn".

These problems of data schema is far from being restricted to LDAP. In a lot of established and emerging technologies the same need for common schema and its easy retrieval exists

As an introduction to the whole project this document gives a survey of schema registry related work that can be of help in the frame of this project.

2.1. Structure of this document

The first major part of this document (chapter 3) consist of definitions and short introductions to a variety of technologies relevant to the data schema problem space. Besides an introduction to LDAP terms including the single schema elements, and to technologies basic to LDAP (ASN.1 and OIDs), other technologies more or less relevant for the project are discussed more or less comprehensively: eXtended Markup Language (XML), Web Services technology, metadata formats, ontologies, Common Information Model (CIM) and some exemplary application technologies. The definition part ends with short descriptions of relevant standards organisations.

LDAP Schema definitions exist for a large number of different subjects, e.g. white and yellow pages services, authorisation and authentication, video conferencing, computing resources, printer, policy, etc. etc. The second major part of this document (Chapter 4) deals with a survey of the already existing LDAP schema that was produced by a number of communities, the Internet Engineering Task Force (IETF) being the most important but not the only such community. This survey allows to better specify the content scope of the planned registry.

There had been several attempts to provide an information service for LDAP schema publication and registration. The third major part of this document deals with such prior approaches as well as with to schema registry attempts in other technologies, although only very brief.

The last Chapter 6 provides a summary of the findings as well as a sketch of requirements for a useful and flexible schema registry.

Appendix A consists of a list of acronyms and technical terms used in the project documentation with references to sections of this document which deal with them.

Appendix B consist of a detailed table of contents.

2.2. Conventions used in this document

The references, marked with square brackets [xxx] refer to the references listed in [RegBib].

This documents contains a lot of quotations from other documents. Passages of these documents that were left out in quotations, are marked by "[...]".

Whenever the phrase "this project" is used in this document, the TERENA Directory Schema Registry project is meant.

A number of acronyms are being used throughout the text. Please view the Chapter 3. Definitions for short explanations of these acronyms. Appendix A gives an alphabetical list of the acronyms that provides the reader with pointers to the places of this document, where they are discussed.

3. Definitions

This chapter provides definitions for most acronyms and technical terms used in the project documents and, where appropriate, gives a short introduction into the subject. As such this chapter functions as a general introduction into the problem space of LDAP schema registry, to which other documents of the project will refer to.

3.1. Definitions of basic technologies for LDAP schema definition

3.1.1. OID (Object Identifiers)

OIDs are a means for world wide uniquely identifying information objects.

An information object is defined by the ASN.1 Standard [X.208] as:

"a well-defined piece of information, definition or specification which requires a name in order to identify its use in an instance of communication". [X.208]

An Object Identifier (OID) is thus defined:

"A value (distinguished from all other such values) which is associated with an information object". [X.208]

In principle OIDs are strings of numbers that are allocated in a hierarchical tree, where the owner of one node in the tree is able to assign new numbers below this node.

OIDs can be written in its numeric representation (e.g. 1.3.6.1.4.1.10126) or in a string representation, where each number corresponds to a registered string (e.g., iso.identified-organization.dod.internet.private.enterprise.daasi). both representations can be mixed. As soon as one string is involved it is no numeric representation anymore. These representations are not to be confused with the object identifier descriptor (e.g. "commonName" for attribute 1.3.6.1.4.1.1466.115.121.1.15).

The three starting points of this hierarchical tree are 0 for ITU (see below ITU), 1 for ISO (see below ISO) and 2 for joint ISO-ITU. These are described in Annex B of [X.208]. For information on OID registries see below.

OIDs are used to identify a great variety of items, e.g., recommendations or standards, firms, projects, ASN.1 modules, ASN.1 types (abstract syntaxes), LDAP elements, PKI policies, etc.

Following LDAP schema elements have an OID descriptor as identifier: object classes, attribute types, matching rules, name forms, DIT content rules and the LDAP syntaxes (which are derived from ASN.1). In addition, a number of LDAP operational elements have OIDs as identifiers, like controls, extensions, etc. For more information on LDAP and OIDs see [RFC 3383] which includes a list of currently assigned OIDs for LDAP elements.

Practices for IANA assignment of OIDS are described in [RFC 1155].

3.1.2. ASN.1 (Abstract Syntax Notation One)

ASN.1 is a formal notation for describing data transmitted by telecommunications protocols.

"Abstract Syntax Notation number One (ASN.1) is a notation that is used in describing messages to be exchanged between communicating application programs. It provides a high level description of messages that frees protocol designers from having to focus on the bits and bytes layout of messages. Initially used to describe e-mail messages within the Open Systems Interconnection protocols, ASN.1 has since been adopted for use by a wide range of other applications, such as in network management, secure e-mail, cellular telephony, air traffic control, and voice and video over the Internet." [ASN.1]

ASN.1 provides a number of pre-defined basic types, e.g., integers (INTEGER), booleans (BOOLEAN), character strings (IA5String, UniversalString...) as well as construction mechanisms for defining constructed types like structures (SEQUENCE), lists (SEQUENCE OF), choice between types (CHOICE), etc. For more Information see [ASN.1], as well as the standard itself [X.208] and [X.660].

3.1.3. ABNF (Augmented Backus-Naur Form)

ABNF [RFC 2234] is the most popular syntax specification language in IETF technical specifications and thus often replaces ASN.1 in this respect.

3.2. Definitions of LDAP schema elements

LDAP schema is specified by using a number of different elements which are described in the following. All these elements are derived from [X.501] and taken over more or less unchanged by the current LDAP standard documents listed in [RFC 3377] and of their revisions listed in [LDAPRoadmap] which are near to completion. The term "directory" is used in the following to refer to these three versions of the schema standard. For additional information about the schema elements, please see the above mentioned documents. The Quotations in the following are from [LDAPModels] if not otherwise stated.

The following sub chapters include a number of element descriptions in a more accessible form than the usual ABNF. Following rules apply here:

- ";" marks the beginning of a comment
- Non literary strings describing the content of a value rather than the value itself (which is variable) are put in angled brackets "<...>"
- "(M)" is used in the commentaries as abbreviation for mandatory
- if "(M)" is missing in the commentary the description element is optional
- <numericoid> means a numeric only representation of an oid
- <oid> means either numeric representation or object identifier descriptor.

3.2.1. DIB (Directory Information Base)

Following definitions of the Directory Information Base (DIB) are given in [LDAPModels]:

"The information held in the Directory is collectively known as the Directory Information Base (DIB)."

"The DIB contains two classes of information:

- 1) user information (e.g., information provided and administrated by users). [...]
- 2) administrative and operational information (e.g., information used to administer and/or operate the directory)."

3.2.2. Directory entry

A directory entry is the container for information on a particular information object, describing a "real world" object.

"A directory entry, a named collection of information, is the basic unit of information held in the Directory. There are multiple kinds of directory entries."

An entry consists of attribute types and attribute values, some of which have specialized functions.

3.2.3. Attribute type

An attribute is the container of single bits of directory information, comparable with the field of a relational database. The LDAP element that specifies such attributes is called attribute type.

"An attribute type governs whether the attribute can have multiple values, the syntax and matching rules used to construct and compare values of that attribute, and other functions. The attribute type indicates whether the attribute is a user attribute or an operational attribute. If operational, the attribute type indicates the operational usage and whether the attribute is modifiable by users or not. [...] An attribute type (a subtype) may derive from another attribute type (a direct supertype). The subtype inherits the matching rules and syntax of its supertype."

Each attribute type is uniquely identified by an OID.

An attribute type is specified by following attribute type description (for a formal correct ABNF see [LDAPModels]):

```
( <numericoid>          ; (M) object identifier (OID)
  NAME <short names (descriptors)>
  DESC <description>
  OBSOLETE              ; not active
  SUP <oid>             ; subtype of
  EQUALITY <oid>        ; equality matching rule (see below 3.2.6)
  ORDERING <oid>        ; ordering matching rule (see below 3.2.6)
  SUBSTR <oid>          ; substrings matching rule (see below 3.2.6)
  SYNTAX <numericoid>{<len>} ; attribute syntax and maximal length
                           (see below 3.2.5)
  SINGLE-VALUE          ; attribute type can only hold one value.
                           Default multi valued
  COLLECTIVE            ; attribtue value for more than one entry,
                           see [LDAPCollective]. Default non collective
  NO-USER-MODIFICATION ; not user modifiable. Default: user modifiable
  USAGE <usage>         ; four usage types allowed. Default type:
                           "userApplications"
  <extensions>         ; (M) not dealt with in this document.
)
```

One or more attribute type / attribute value pair form the relative distinguished name of an entry (see RDN). Name forms define which attributes can be used to form the RDN (see name forms 3.2.10).

A special attribute type named object class is used to specify one or more object classes of an entry (see 3.2.4).

3.2.4. Object class

An object class characterises an entry, i.e. it specifies what kind of entry it is, for instance whether the entry describes a person or an organisation.

An object class is:

"an identified family of objects (or conceivable objects) which share certain characteristics. Every object belongs to at least one class. An object class may be a subclass of other object classes, in which case the members of the former class, the subclass, are also considered to be members of the latter classes, the superclasses. There may be subclasses of subclasses, etc., to an arbitrary depth." [X.501].

"An object class (a subclass) may be derived from an object class (its direct superclass) which is itself derived from an even more generic object class. For structural object classes, this process stops at the most generic object class, 'top' [...]. An ordered set of superclasses up to the most superior object class of an object class is its superclass chain"

"Each object class identifies the set of attributes required to be present in entries belonging to the class and the set of attributes allowed to be present in entries belonging to the class. As an entry of a class must meet the requirements of each class it belongs to, it can be said that an object class inherits the sets of allowed and required attributes from its superclasses. A subclass can identify an attribute allowed by a subclass [sic!+ the texts means superclass here] as being required. If an attribute is a member of both sets, it is required to be present. Each object class is defined to be one of three kinds of object classes: Abstract, Structural, and Auxiliary."

Each object class is uniquely identified by an OID.

An object class is specified by following object class description (for a formal correct ABNF see [LDAPModels]):

```

( <numericoid>          ; (M) object identifier (OID) (see below OID)
  NAME <short names (descriptors)>
  DESC <description>
  OBSOLETE              ; not active
  SUP <oid>             ; the direct super object class
  <kind>                ; kind of class. Allowed kinds are:
                        ABSTRACT, STRUCTURAL and AUXILIARY
  MUST <list of oids>   ; mandatory attribute types
  MAY" <list of oids>   ; optional attribute types
  <extensions>         ; (M) not dealt with in this document
)

```

3.2.5. Attribute syntax

Attribute syntaxes define the syntax of the attribute values:

"Syntax definitions constrain the structure of attribute values stored in an LDAP directory, and determine the representation of attribute and assertion values transferred in the LDAP protocol" [LDAPSyntaxes].

Each attribute syntax is uniquely identified with an OID.

"A suggested minimum upper bound on the number of characters in an attribute value with a string-based syntax, or the number of octets in a value for all other syntaxes, MAY be indicated by appending the bound inside of curly braces following the syntax's OBJECT IDENTIFIER in an attribute type definition" [LDAPSyntaxes].

Syntaxes have to be known by the implementations. Therefore it is advised only to use standardised syntaxes, although the introduction of new syntaxes is theoretically possible. For a current list of known standardised syntaxes see [LDAPSyntaxes].

An attribute syntax is specified by following syntax description (for a formal correct ABNF see [LDAPModels]):

```

( <numericoid>          ; (M) object identifier (OID) (see below OID)
  DESC <description>
  <extensions>         ; (M) not dealt with in this document
)

```

3.2.6. Matching rules

Matching rules define the behaviour of the comparing and ordering attribute values.

In attribute type definitions several kinds of matching rules can be specified.

"Matching rules are used by servers to compare attribute values against assertion values when performing Search and Compare operations. They are also used to identify the value to be added or deleted when modifying entries, and are used when comparing a purported distinguished name with the name of an entry. A matching rule specifies the syntax of the assertion value. Each matching rule is identified by an object identifier (OID) and, optionally, one or more short names (descriptors)."

There are three different types of matching rules: EQUALITY (for exact matching operations), SUBSTRING (for substring matching operations) and ORDERING (for matching operations that are done to sort values). Every matching rule is of one of these three types, although this is not reflected in the matching rule description.

Matching rules have to be known by the implementations. Therefore it is advised only to use standardised matching rules, although the introduction of new matching rules is theoretically possible. For a current list of known standardised matching rules see [LDAPSyntaxes].

An matching rule is specified by following matching rule description (for a formal correct ABNF see [LDAPModels]):

```
( <numericoid>          ; (M) object identifier (OID)
  NAME <short names (descriptors)>
  DESC <description>
  OBSOLETE                ; not active
  SYNTAX <numericoid>    ; (M) value syntax and maximal length
                        (see below attribute syntax)
  <extensions>           ; (M) not dealt with in this document
)
```

In Addition to the matching rule element there is the matching rule use element.

"The <AttributeTypeDescription> does not list the matching rules which can be used with that attribute type in an extensibleMatch search filter. This is done using the 'matchingRuleUse' attribute"

"A matching rule use lists the attributes which are suitable for use with an extensible matching rule."

An matching rule use is specified by following matching rule use description (for a formal correct ABNF see [LDAPModels]):

```
( <numericoid>          ; (M) object identifier (OID) pointing to the
                        matching rule to which the use description
                        applies.
  NAME <short names (descriptors)>
  DESC <description>
  OBSOLETE                ; not active
  APPLIES <oids>          ; list of attribute types
  <extensions>           ; (M) not dealt with in this document
)
```

3.2.7. DIT (Directory Information Tree)

Following definitions of the Directory Information Trees (DIT) are given in [LDAPModels]:

"The set of entries representing the DIB are organized hierarchically in a tree structure known as the Directory Information Tree (DIT)."

"Specifically, a tree where vertices are the entries. The arcs between vertices define relations between entries. If an arc exists from X to Y, then the entry at X is the immediate superior of Y and Y is the immediate subordinate of X. An entry's superiors are the entry's immediate superior and its superiors. An entry's subordinates are all of its immediate subordinates and their subordinates."

"Similarly, the superior/subordinate relationship between object entries can be used to derive a relation between the objects they represent. DIT structure rules can be used to govern relationships between objects. Note: An entry's immediate superior is also known as the entry's parent and an entry's immediate subordinate is also known as the entry's child."

3.2.8. RDN (Relative Distinguished Name)

One or more attribute type / attribute value pairs are used to form the name of an entry.

"An entry's relative distinguished name must be unique among all immediate subordinates of the entry's immediate superior (i.e., all siblings)."

Thus: "Each entry is named relative to its immediate superior."

3.2.9. DN (Distinguished Name)

The DN uniquely identifies an entry among all entries in a DIB.

"An entry's fully qualified name, known as its Distinguished Name (DN) [X.501], is the concatenation of its RDN and its immediate superior's DN. A Distinguished Name unambiguously refers to an entry in the tree."

The representation of DNs and RDNs in LDAP is specified in [RFC 2253].

3.2.10. Name form

According to [X.501] a name form:

"specifies a permissible RDN for entries of a particular structural object class. A name form identifies a named object class and one or more attribute types to be used for naming (i.e. for the RDN). Name forms are primitive pieces of specification used in the definition of DIT structure rules" [X.501].

[LDAPModels] adds:

"Each name form indicates the structural object class to be named, a set of required attribute types, and a set of allowed attributes types. A particular attribute type cannot be listed in both sets. Entries governed by the form must be named using a value from each required attribute type and zero or more values from the allowed attribute types.

Each name form is identified by an object identifier (OID) and, optionally, one or more short names (descriptors)."

A Name Form is specified by following Name Form description (for a formal correct ABNF see [LDAPModels]):

```
( <numericoid>          ; (M) object identifier (OID) (see below OID)
  NAME <short names (descriptors)>
  DESC <description>
  OBSOLETE                ; not active
  OC <oid>                 ; (M) structural object class
  MUST <oids>             ; (M) mandatory attribute types to be used for
                           naming
  MAY <oids>              ; optional attribute types to be used for naming
  <extensions>           ; (M) not dealt with in this document
)
```

3.2.11. DIT structure rule

DIT Structure rules defines which kinds of entries may be stored below or above respectively other kinds of entries. Below and above refer to the location in the DIT. A DIT Structure Rule thus regulates where object entries can be placed in the DIT.

[X.501] defines a DIT Structure Rule as:

"rule governing the structure of the DIT by specifying a permitted superior to subordinate entry relationship. A structure rule relates a name form, and therefore a structural object class, to superior structure rules. This permits entries of the structural object class identified by the name form to exist in the DIT as subordinates to entries governed by the indicated superior structure rules" [X.501]

A DIT Structure Rule is specified by following DIT Structure Rule description (for a formal correct ABNF see [LDAPModels]):

```
( <ruleid>                ; (M) Rule number. This is a simple number and
                           no OID. Thus this number is not unique!
  NAME <short names (descriptors)>
  DESC <description>
  OBSOLETE                ; not active
  FORM <oid>              ; (M) oid of a name form to which this DIT
                           structure rule applies
  SUP <ruleids>          ; list of superior rules
)
```

```
    <extensions>          ; (M) not dealt with in this document
  )
```

3.2.12. DIT content rules

[X.501] defines a DIT content rule as a:

"rule governing the content of entries of a particular structural object class" [X.501].

[LDAPModels] adds:

"For DIT entries of a particular structural object class, a DIT content rule specifies which auxiliary object classes the entries are allowed to belong to and which additional attributes (by type) are required, allowed or not allowed to appear in the entries.

The list of precluded attributes cannot include any attribute listed as mandatory in rule, the structural object class, or any of the allowed auxiliary object classes.

Each content rule is identified by the object identifier, as well as any short names (descriptors), of the structural object class it applies to."

A DIT Content Rule is specified by following DIT Content Rule description (for a formal correct ABNF see [LDAPModels]):

```
( <numericoid>          ; (M) object identifier (OID) (see below OID)
                             of the of the structural object class associated
                             with this DIT content rule;
  NAME <short names (descriptors)>
  DESC <description>
  OBSOLETE                ; not active
  AUX <oids>               ; Auxiliary object classes
  MUST <oids>              ; mandatory attribute types
  MAY <oids>               ; optional attribute types to be used
  NOT <oids>               ; attribute types not to be used
  <extensions>           ; (M) not dealt with in this document
)
```

3.2.13. Subschema subentry

"Subschema (sub)entries are used for administering information about the directory schema. A single subschema (sub)entry contains all schema definitions [...] used by entries in a particular part of the directory tree."

For more information on subschema subentry see 5.1.1.

3.3. Definitions of XML, Metadata and Ontology technologies

As mentioned in the introduction there are other technologies that use definable schema and thus are in need of schema repositories or registries. Of these, XML is the most important one, which will be dealt with by introducing to the core specification as well as to 3 XML based technologies DSML, WSDL and UDDI.

Part of the schema registry problem space is the description of text sources (in our case schema specifications) and usage of keywords for classifying schema so that it can be found by such keywords. This leads to the problem space of metadata, data about data, in our case data about schema, which are metadata themselves. technologies that are relevant in this respect, namely Dublin Core and ISO/IEC 11179 are thus treated here as well.

For a future user friendly more intelligent registry that goes beyond the registry planned in this project, new technologies based on so called ontologies could get used. To introduce in such technologies, like RDF, OWL and CIM, they are described in this section as well. The schema problem can be found in these technologies as well.

3.3.1. XML (Extended Markup Language)

XML is a standard for defining markup languages. It facilitates for detailed structuring of documents by including a flexible and extensible semantic into the structure of the document. This semantic is well defined via a so called Document Type Definition (DTD) or via an XML schema. For the main specification see [XML]. There is a great variety of specifications that are based on XML. For more info on these see <http://www.w3c.org>.

3.3.2. DSML (Directory Service Markup Language)

DSML version 1 [DSMLv1] is a means for representing directory information as an XML document. It can be used as a directory enhancement for XML based applications or to convert XML data to directory data. A DSMLv1 document can describe: directory entries, directory schema or both.

In its version 2 [DSMLv2], DSML can also represent LDAP operations and their results, and will enable access to a directory through XML protocols like SOAP as it is used in Web Services (see below WSDL).

DSML is meanwhile being developed as an OASIS standard. For more information see <http://www.oasis-open.org/committees/dsml/>.

3.3.3. WSDL (Web Services Description Language)

WSDL [WSDL] is an XML-based language used to specify Web services and their access interface.

"WSDL is an XML format for describing network services as a set of endpoints operating on messages containing either document-oriented or procedure-oriented information. The operations and messages are described abstractly, and then bound to a concrete network protocol and message format to define an endpoint. Related concrete endpoints are combined into abstract endpoints (services)." [WSDL]

The Stencil Group defines Web Services as:

"Loosely coupled, reusable software components that semantically encapsulate discrete functionality and are distributed and programmatically accessible over standard Internet protocols." (see http://www.stencilgroup.com/ideas_scope_200106wsdefined.html)

Web Services described in WSDL can be registered in an UDDI directory.

3.3.4. UDDI (Universal Description, Discovery and Integration)

UDDI is a platform-independent framework for describing services, discovering businesses, and integrating business services by using the Internet. It is a directory for storing information about web services, and their interfaces described via WSDL

"Universal Description, Discovery and Integration (UDDI) is a specification for distributed Web-based information registries of Web services. UDDI is also a publicly accessible set of implementations of the specification that allow businesses to register information about the Web services they offer so that other businesses can find them." [UDDIWP]

"The core component of the UDDI project is the UDDI business registration, an XML file used to describe a business entity and its Web services. Conceptually, the information provided in a UDDI business registration consists of three components: "white pages" including address, contact, and known identifiers; "yellow pages" including industrial categorizations based on standard taxonomies; and "green pages", the technical information about services that are exposed by the business. Green pages include references to specifications for Web services, as well as support for pointers to various file and URL based discovery mechanisms if required." [UDDIWP]

The UDDI specifications are a number of documents describing, e.g. an API, Data structures, an XML Schema, replication mechanisms, etc. (see <http://www.oasis-open.org/committees/uddi-spec/>). The current version of the specs is 3. The main text is [UDDI].

SOAP (Simple Object Access Protocol) is used as transport protocol for UDDI registry access.

UDDI Data types can very well be represented in LDAP which then can be the directory technology for implementing UDDI registries. [UDDILDAP] from Novell defines schema elements to represent a businessEntity, a businessService, a bindingTemplate, a tModel (a reference system based on abstraction), and a publisherAssertion. The advantages for using LDAP as basis for UDDI registry are the well established features of LDAP: security, replication mechanisms, read access optimisation, etc.

Instead of integrating LDAP and UDDI, others are only mapping the security capabilities of their Web services platform to LDAP so that companies can specify security information in LDAP and use that to secure a UDDI registry.

3.3.5. DC (Dublin Core)

Metadata in its most simple definition is data about data, e.g. data about a textual resource. Technologies for describing metadata are, e.g. XML and RDF. Part of metadata are keywords that in an ideal case are linked to ontologies, at least to a controlled vocabulary.

The most important format for metadata about texts (in a narrow sense: books, articles, documents, web pages) is Dublin Core (DC) [RFC 2413] which defines a set of 15 descriptive elements. Since then one additional element has been added.

The 16 currently standardised elements are:

Title	A name given to the resource.
Creator	An entity primarily responsible for making the content of the resource.
Subject	The topic of the content of the resource.
Description	An account of the content of the resource.
Publisher	An entity responsible for making the resource available
Contributor	An entity responsible for making contributions to the content of the resource.
Date	A date associated with an event in the life cycle of the resource.
Type	The nature or genre of the content of the resource.
Format	The physical or digital manifestation of the resource.
Identifier	An unambiguous reference to the resource within a given context.
Source	A reference to a resource from which the present resource is derived.
Language	A language of the intellectual content of the resource.
Relation	A reference to a related resource.
Coverage	The extent or scope of the content of the resource.
Rights	Information about rights held in and over the resource.
Audience	A class of entity for whom the resource is intended or useful.

There is a possibility to enhance these elements by so called qualifiers that refine an element or provide encoding information:

"At the time of the ratification of this document, the DCMI recognizes two broad classes of qualifiers:

- **Element Refinement.** These qualifiers make the meaning of an element narrower or more specific. A refined element shares the meaning of the unqualified element, but with a more restricted scope. A client that does not understand a specific element refinement term should be able to ignore the qualifier and treat the metadata value as if it were an unqualified (broader) element. The definitions of element refinement terms for qualifiers must be publicly available.
- **Encoding Scheme.** These qualifiers identify schemes that aid in the interpretation of an element value. These schemes include controlled vocabularies and formal notations or parsing rules. A value expressed using an encoding scheme will thus be a token selected from a controlled vocabulary (e.g., a term from a classification system or set of subject headings) or a string formatted in accordance with a formal notation (e.g., "2000-01-01" as the standard expression of a date). If an encoding scheme is not understood by a client or agent, the value may still be useful to a human reader. The definitive description of an encoding scheme for qualifiers must be clearly identified and available for public use." [DCQual]

A current list of all approved DC elements and their qualifiers can be found in [DCcurrent], they are discussed in their relevance to the Schema Registry in [RegSchema].

DC can be represented with e.g. XML. There is an unfinished work on representing DC in LDAP [LDAPDC] which will be worked upon in the frame of this Project. Useful input for storing bibliographic data in LDAP can be found in [Klasen] and [RFC1807].

3.3.6. ISO/IEC 11179

ISO/IEC 11179 is an ISO standard that standardises data representation in metadata registries.

It is a description of data elements in a repository and can be used in actual implementations and for metadata exchange among repositories, such as registries for Name spaces, data elements, XML-tags, etc.

"To facilitate global electronic communications, the International Standards community has been working diligently to define an Open Systems Interconnection Environment (OSIE) within which diverse computer hardware and applications could share information. Standards have been proposed or defined for three (hardware, software, and communications) of the four (hardware, software, communications, and data) basic components required for open information processing systems. ISO/IEC 11179 for data specification, the fourth basic component for open information systems, provides a mechanism for enabling data to be shared in the OSIE." [ISO/IEC 11179-1]

"Sharing data involves the ability to locate desired data, retrieve the data, and to exchange the data with others. When data elements are well documented according to ISO/IEC 11179 and the documentation is managed in a Data Element Registry, finding and retrieving them from disparate databases as well as sending and receiving them via electronic communications are made easier" [ISO/IEC 11179-1]

A very interesting project that created a prototype implementation of ISO/IEC 11179 in combination with LDAP is the DDDS (Distributed Data Dictionary Service) (see below)

3.3.7. OWL

In philosophy ontology is the science of what exists, of the kinds and structures of objects, properties, events, processes and relations in every area of reality.

"An ontology defines the terms used to describe and represent an area of knowledge. Ontologies are used by people, databases, and applications that need to share domain information (a domain is just a specific subject area or area of knowledge, like medicine, tool manufacturing, real estate, automobile repair, financial management, etc.). Ontologies include computer-usable definitions of basic concepts in the domain and the relationships among them" [WebontReq]

Ontologies specify the following kinds of concepts:

- Classes (entities, things, terms) belonging to a domain of interest
- The relationships that can exist among Classes
- The properties (or attributes) Classes may have

In a broad sense we can define ontology as a database of concepts and relations between concepts. Within the scope of this definition also lie classification systems (concepts put into a hierarchical system of class and subclass, i.e. only one relation type) as well as thesauri (concepts and relations as "is equal", "is part of", etc, i.e. a limited number of relation types). Ontologies conceived as knowledge storage are polydimensional ontologies, i.e. they have an unlimited number of relation types.

Ontologies are important in any computer based knowledge system. In the sphere of the WWW, OWL (Ontology Web Language) [OWL], a standard for ontology description is being specified on the basis of XML (for more information see <http://www.w3.org/2001/sw/WebOnt/>).

Ontologies can be represented with LDAP technology, as respective work in CIM has shown [CIMLDAP].

3.3.8. RDF (Resource Description Format)

RDF is a framework for the description, processing and exchange of metadata (see below). The main specifications are the syntax definition [RDF] and a schema definition [RDFS].

The elementary data model consists of three elements: Resource, Property and Statement. In its simpleness and abstraction RDF/RDFS is suitable for any kind of metadata up to complex ontologies [Staab]. XML can be used for representing RDF.

3.3.9. CIM (Common Information Model)

CIM is a standard of the Distributed Management Task Force (DMTF) (see below)

"The DMTF Common Information Model (CIM) is an approach to the management of systems, software, users, networks and more, that applies the basic structuring and conceptualisation techniques of the object-oriented paradigm.

A management model is provided to establish a common conceptual framework for a description of the managed environment. A fundamental taxonomy of objects is defined — both with respect to classification and association, and with respect to a basic set of classes intended to establish a common framework. The management model is divided into the following conceptual layers:

Core Model—an information model that captures notions applicable to all domains of management
Common Models—information models that capture notions common to particular management domains but independent of a particular technology or implementation. The common domains include Systems, Applications, Devices, Users, Networks, Policies and Databases.

Extension Models—represent technology-specific extensions of the Common Models. These models are specific to environments, such as operating systems, or to vendors."
[CIMCore]

The current version of the CIM Schema is 2.6. See http://www.dmtf.org/standards/standard_cim.php, version 2.7. is currently being worked on.

Basically, CIM is an UML like ontology description language for computer related entities. It is possible to do mappings of CIM schema to XML [CIMXML] and to LDAP [CIMLDAP], an example for such a mapping is [CIMUserLDAP].

3.4. Definition of exemplary application technologies

This section describes two exemplary applications of LDAP, that require special purpose schema, namely Samba and Radius.

3.4.1. Samba

"Samba is a suite of Unix applications that speak the SMB (Server Message Block) protocol. Many operating systems, including Windows and OS/2, use SMB to perform client-server networking. By supporting this protocol, Samba allows Unix servers to get in on the action, communicating with the same networking protocol as Microsoft Windows products. Thus, a Samba-enabled Unix machine can masquerade as a server on your Microsoft network and offer the following services:

- Share one or more filesystems
- Share printers installed on both the server and its clients
- Assist clients with network neighbourhood browsing
- Authenticate clients logging onto a Windows domain
- Provide or assist with WINS name server resolution". [Eckstein]

Samba is an open source project (see www.samba.org).

3.4.2. RADIUS (Remote Authentication Dial In User Service)

RADIUS is an IETF protocol,

"for carrying authentication, authorization, and configuration information between a Network Access Server which desires to authenticate its links and a shared Authentication Server." [RFC 2865]

There are a number of open source implementations of RADIUS, e.g. FreeRadius (see www.freradius.org) and Cistron Radius (see <http://www.radius.cistron.nl/>).

3.5. Definitions of relevant organisations

3.5.1. IETF (Internet Engineering Task Force)

The IETF (www.ietf.org) is the most important standardisation organisation for Internet protocols.

"The Internet Engineering Task Force (IETF) is a large open international community of network designers, operators, vendors, and researchers concerned with the evolution of the Internet architecture and the smooth operation of the Internet"
(www.ietf.org/overview.html)

The actual work of the IETF is done in the frame of working groups that meet three times a year and have discussions on the groups mailing lists. IETF WGs normally have two chairs. The WGs are organized by topic into several areas like applications, security, network, etc. These areas have chairs, called Area Directors (AD). For the organisational parts of the IETF, namely IAB, IESG and IANA, see the three following subsections. A good introduction to IETF, IAB, IESG and IANA can be found in [RFC 3160].

The technical specifications of the IETF are named Requests For Comments (RFC), which are grouped into a number of categories (the first three of which comprise the standards track): Proposed Standard, Draft Standard, Internet Standard, Experimental, Informational, Historic, and Best Current Practice (BCP). RFCs must first be published as Internet Drafts that are open to discussions. For more information about the single RFC categories and the standardisation process see [RFC 2026]. There is a specification how to XML encode an RFCs and Internet Drafts in [RFC 2629].

The current finalisation of the LDAP standard specification is work of an IETF working group called ldapbis (see <http://www.ietf.org/html.charters/ldapbis-charter.html>).

The applications area has set up a so called LDAP Directorate:

"The LDAP Directorate provides expert review of LDAP I-Ds produced by individuals or non-LDAP working groups to Apps ADs, WG chairs, and IESG/IAB members. The Directorate may also provide (as time permits) guidance to authors of LDAP I-Ds. The Directorate consists of at least one chair from each of the LDAP WGs (LDAPBIS, LDAPEXT, LDUP) and a small set of LDAP (and possible other) experts selected by Application Area ADs (with input from WG chairs). [...]The directorate is not a formal review body; the IESG has that role. Therefore, comments made by the directorate have no more weight than those made by individual IETF participants. The directorate is best thought of as a convenient way to contact a group of LDAP experts within the IETF."
(<http://www.apps.ietf.org/ldap-directorate.html>)

3.5.2. IAB (Internet Architecture Board)

The IAB (<http://www.iab.org/>) "The IAB is responsible for defining the overall architecture of the Internet, providing guidance and broad direction to the IETF. The IAB also serves as the technology advisory group to the Internet Society, and oversees a number of critical activities in support of the Internet." (<http://www.ietf.org/glossary.html#IAB>).

3.5.3. IESG (Internet Engineering Steering Group)

The IESG (<http://www.ietf.org/iesg.html>) is a part of the IETF. Members of the IESG are the Area Directors of the IETF.

"The IESG is responsible for technical management of IETF activities and the Internet standards process. [...] The IETF is run by rough consensus, and it is the IESG that decides if a WG has come up with a result that has a real consensus." [RFC 3160].

The IESG is directly responsible for the actions associated with the Internet "standards track," including final approval of specifications as Internet Standards.

3.5.4. IANA (Internet Assigned Numbers Authority)

The IANA (www.iana.org) is an registration authority for the many unique parameters and protocol values necessary for operation of the Internet. In their own words: "Dedicated to preserving the central coordinating functions of the global Internet for the public good."

Types of registries range from unique port assignments to the registration of character sets, object identifier descriptors, error codes, protocol elements, MIME types, etc. [RFC 3383] mandates IANA registration of such LDAP elements.

The IANA Directory of General Assigned Numbers, which yet does not include the elements defined in [RFC 3383], is available at <http://www.iana.org/numbers.html>.

3.5.5. ITU (International Telephone Union)

The ITU (www.itu.int), formerly named International Telephone and Telegraph Consultative Committee (CCITT), is an international standardisation organisation that has the national telecommunication organisations and their commercial follow-ups as members.

"The three Sectors of the Union – Radiocommunication (ITU-R), Telecommunication Standardization (ITU-T), and Telecommunication Development (ITU-D) - work today to build and shape tomorrow's networks and services. Their activities cover all aspects of telecommunication, from setting standards that facilitate seamless interworking of equipment and systems on a global basis to adopting operational procedures for the vast and growing array of wireless services and designing programmes to improve telecommunication infrastructure in the developing world. [...]In ITU-T, experts prepare the technical specifications for tele-communication systems, networks and services, including their operation, performance and maintenance." (<http://www.itu.int/aboutitu/overview/role-work.html>)

The ITU defined the Open System Interconnection (OSI), a part of which are the X.500 recommendations.

3.5.6. ISO (International Standards Organisation)

The ISO (<http://www.iso.ch>) that among many other Standards co-authored the X.500 Recommendation (also published as ISO 9594 1-9) and with in the ISO/IEC JTC1/SC32 WG 2 (<http://metadata-stds.org/>) the Metadata standards ISO/IEC 11179.

3.5.7. ISO/IEC 11179 Metadata Registry Implementation Coalition

"ISO/IEC 11179 Metadata Registry Implementation Coalition has been organized to provide a forum for information exchange on the implementation of metadata registries based on the ISO/IEC-11179. The Consortium consists of members interested in addressing ISO/IEC-11179 reference implementations of metadata registries, influencing commercial vendors to support ISO/IEC-11179 in their tools, developing methods to support metadata exchange between metadata registries, sharing information and lessons learned on implementation approaches, being an advocate and clearinghouse for metadata registry issues, and developing partnerships to support data management across organizations." [U'Ren]

3.5.8. DMTF (Distributed Management Task Force)

The DMTF is an industrial organisation that develops management standards and initiatives for desktop, enterprise and Internet environments. It aims at enabling a more integrated, cost-effective approach to management through interoperable management solutions. An important group of such standards is the Common Information Model (see above).

3.5.9. OASIS (Organization for the Advancement of Structured Information Standards)

OASIS (www.oasis-open.org/)

"is a not-for-profit, global consortium that drives the development, convergence and adoption of e-business standards. [...] OASIS produces worldwide standards for security, Web services, XML conformance, business transactions, electronic publishing, topic maps and interoperability within and between marketplaces." (<http://www.oasis-open.org/who/>)

OASIS is the board that defines, e.g. DSML.

3.5.10. GGF (Global Grid Forum)

GGF (www.gridforum.org) is an international body that aims at specifying standards and recommendations in the realm of Grid Computing. The standardisation procedures are modelled on the IETF procedures. Besides standards, it defines a general Grid architecture to promote Grid computing.

"The Grid is a consistent and standardized environment for collaborative, distributed problem solving that requires high performance computing on massive amounts of data that are stored, and/or generated at high data rates using widely distributed, heterogeneous resources

The Grid is an inherently layered architecture that provides for common services and a diversity of middleware that supports building distributed, large-scale, and high performance applications and problem solving systems" W.E. Johnston quoted by [Foster]

3.5.11. Internet2

Internet2 (<http://www.internet2.org/>) is

"is a consortium being led by 200 universities working in partnership with industry and government to develop and deploy advanced network applications and technologies, accelerating the creation of tomorrow's Internet. Internet2 is recreating the partnership

among academia, industry and government that fostered today's Internet in its infancy. The primary goals of Internet2 are to:

- Create a leading edge network capability for the national research community
- Enable revolutionary Internet applications
- Ensure the rapid transfer of new network services and applications to the broader Internet community.". (<http://www.internet2.edu/about/aboutinternet2.html>)

Part of Internet2 is the Middleware Architecture Council for Education (MACE), see <http://middleware.internet2.edu/MACE/>.

3.5.12. EDUCAUSE

EDUCAUSE (<http://www.educause.edu>) is

"a nonprofit association whose mission is to advance higher education by promoting the intelligent use of information technology" (<http://www.educause.edu/defined.html>)

3.5.13. The Open Group

The Open Group (www.opengroup.org) is an industrial consortium, in their own words "an international vendor and technology-neutral consortium that is committed to delivering greater business efficiency by bringing together buyers and suppliers of information technology to lower the time, cost and risk associated with integrating new technology across the enterprise." (<http://www.opengroup.org/overview/index.htm>)

One of the many technology fields in which The Open Group is active is directories. These activities are bundled in The Directory Interoperability Forum (<http://www.opengroup.org/directory/>), which aims at "working to Enable and Promote Interoperable Directory Solutions".

4. Existing LDAP schema standards

This chapter summarises possible sources for LDAP schema that could be registered in the schema registry. It is structured according to the different standardisation and other relevant bodies that have published LDAP schema.

4.1. X.500 schema standards

LDAP is directly derived from X.500 [X.500], the common directory standard jointly defined by ITU and ISO. One part of the X.500 recommendations, namely [X.520] and [X.521], contains schema definitions that are considered as belonging to the standard. This schema mainly covers schema for an international white pages service.

4.2. IETF LDAP schema standards

Most of the schema of [X.520] and [X.521] has been transferred to LDAP in [RFC 2252] and [RFC 2256] which is being updated in [LDAPSchema]. Syntaxes and Matching Rules have been specified in [RFC 2256] which is being updated by [LDAPSyntaxes]. Besides this core schema additional schema for specific purposes has been specified in a number of IETF documents.

In a first survey, following RFCs have been found to contain relevant schema:

[RFC 1274], [RFC 1276], [RFC 1279], [RFC 1608], [RFC 1609], [RFC 1801], [RFC 1804], [RFC 2079], [RFC 2164], [RFC 2247], [RFC 2293], [RFC 2294], [RFC 2307], [RFC 2377], [RFC 2587], [RFC 2589], [RFC 2649], [RFC 2657], [RFC 2713], [RFC 2714], [RFC 2739], [RFC 2798], [RFC 2926], [RFC 3045], [RFC 3060], [RFC 3112], [RFC 3296].

Whether it makes sense to include the rather old and X.500 related RFCs mentioned here [RFC 1274-1804], is yet to be considered.

In addition you can find LDAP schema in Internet-Drafts that are work in progress. If the registry will include such scheme, it has to be marked as work in progress and a mechanism has to be implemented for storing different versions. The details of the policy that defines if and which Drafts shall be registered in the frame of this project and in the future operation of the registry is specified in [RegPolicy]. Versioning will be an issue for the schema registry design.

[UDDILDAP] and [PolicyLDAP] will be definitely be registered during the project phase because they make good testing candidates with the complex schema they define.

In a first survey following additional Internet-Drafts have been found to contain relevant:

[ComponentMatch], [CrispASN], [CrispCore], [CrispDNS], [CrispIPv4], [CrispIPv6], [DUACConf], [KDCDAP], [KDCkeysLDAP], [Laser], [LCUP], [LDAPNDS], [LDAPPKI], [LDAPPMI], [LDAPPrinter], [LDAPX509Cert], [RFC2307bis], [VpimDir].

4.3. LDAP schema standards of other standards organisations

4.3.1. DMTF

Besides a guideline for CIM to LDAP mappings [CIMLDAP], the DMTF has published a number of documents with LDAP mappings of CIM schema, e.g. [CIMUserLDAP]. In the frame of this Project only this example will be registered.

4.3.2. Open Group

The Open Group (www.opengroup.org) is also involved in directory work. In some of their activities, LDAP scheme is being defined, e.g. [LDAPDCE].

4.3.3. Internet 2/EDUCAUSE

Internet2 in cooperation with EDUCAUSE have made considerable effort to define LDAP schema for use in the educational community in the US, and has thus set de facto standards that have been considered and used all over the world. Schema that have to be included into the registry are:

- eduPerson (1.0, 1.5 and 1.6), and eduOrg for representing educational persons and institutions in an LDAP directory (see <http://www.educause.edu/eduperson/>).
- comObject for a schema for representing video and voice over conferencing endpoints in LDAP directory (see <http://middleware.internet2.edu/video>).
- Other schema proposals might come out of other Internet2 middleware activities e.g. on health care and on groups.

4.4. LDAP schema de facto standards of proprietary directory software

For each of the most important currently available LDAP supporting directory server there has been published extra schema that enables special features of the product. In addition some client programs also make use of LDAP by specialised schema. This chapter gives a short survey about these two types of schema

4.4.1. Special schema for Novell

Novell Directory Service NDS (called eDirectory since version 8.0) adds a number of schema elements to allow for special features, e.g. the following:

- A container class, called domain that accepts most leaf objects as subordinate objects
- Attributes for effective class status for the Person, Residential Person and Organizational Person object classes
- An ndsLoginProperties class which allows Person, Organizational Person, Organization, and Organizational Unit classes to inherit all the attributes required for logging in to NDS

NDS schema is specified in [LDAPNDS] and is made available by Novell at <http://www.novell.com/products/edirectory/schema/>. For an introduction to NDS see [Walton] and to NDS schema see [McLain]. On mapping between LDAP schema and NDS schema see [Burnett].

The NDS schema is extensible which means developers can add to the types of objects and attributes maintained by NDS. Novell maintains a registry for such schema (see below).

4.4.2 Netscape Directory Service

Netscape Directory Services was one of the most important LDAP server implementations. After having been included into the iPlanet product (marketed in an alliance between Netscape and Sun) and after separation of the iPlanet alliance, it is unclear whether there will be a new Netscape DS. Anyway SUN is now marketing an own LDAP implementation as part of the SUN ONE suite (see below).

Netscape specified a great variety of LDAP schema for additional functionality, like storing configuration parameters of different Netscape products in an LDAP directory. Following additional schemata can be found for the old NDS 4.1. version: Administration Server Object Classes, Calendar Server Object Classes, Collabra Server Object Classes, Compass Server Object Classes, Delegated Administrator Object Classes, Directory Server Object Classes, Enterprise Server Object Classes, Mission Control Desktop & Client Object Classes, Messaging Server Object Classes. Each of these schemata define up to 20 or so new object classes. These are specified at <http://developer.netscape.com/docs/manuals/directory/schema2/41/contents.htm>.

Schema supported by Netscape Directory Server 6.0 is documented in <http://enterprise.netscape.com/docs/directory/60/schema/server.htm>.

The Netscape Browser allows Roaming which allows to have bookmarks and preferences automatically synchronised and ready to use, independent of the computer on which the browser is installed. The browser is able to retrieve these data from a so called roaming profile stored in an LDAP Server. This feature and the needed LDAP schema is documented in [Kooij].

4.4.3. SUN Directory Server

The Directory Server marketed by sun is the former iPlanet server. Its latest version 5.1. contains, like earlier versions, a documentation of all schema that goes with it. That schema specified at <http://docs.sun.com/source/816-5613-10/index.html>.

SUN also uses LDAP for the storage of distributed Java objects, specified in [RFC 2713]. In its Unix operating system Solaris since version 8 a directory service is used for managing the users and resources. Sun uses an extended version of the NIS schema [RFC 2307] which now is being specified in [RFC 2307bis]. An example of Solaris schema enhancements can be found at <http://www.tzone.org/~okapi/up2/solaris.schema>. A good overview on Solaris and LDAP is given by [Bialaski].

4.4.4. Microsoft Active Directory.

The Microsoft version of X.500/LDAP Server is called Active Directory (AD) and is used for user and resource management in the operating systems Windows 2000 and Windows XP as well as in the .NET environment. Again, additional schema has been defined for AD.

When looking for an online specification of this schema, we only found a reference to a non existing resource: "For more information, and to view the online reference pages for the Active Directory schema, see <http://msdn.microsoft.com/certification/schema/>".

4.5. LDAP schema of research projects

In a number of research projects in which LDAP was used, interesting LDAP schema has been published.

4.5.1. Grid Computing projects

Grid computing, being a very relevant research topic, is one of the most productive fields with respect to LDAP schema definition.

In Grid computing the LDAP based Globus toolkit (www.globus.org) is widely being used. Parts of Globus are based on special LDAP schema. A good introduction to LDAP in Globus can be found in [Fitzgerald] and in [Czajkowski], e.g.

- schema for the replication service (with object classes like GlobusReplicaCatalog, GlobusReplicaLogicalFileObject, available at <http://www-fp.globus.org/qt2.2/replica.html>)
- schema of the Metacomputing Directory Service (MDS) for representing computers and software (with object classes like MdsSoftware, MdsComputer, see <http://www.globus.org/mds/Schema.html>).

Other relevant grid related schema was developed e.g. in the frame of the Datatag project (<http://datatag.web.cern.ch/datatag/>) in an activity called GLUE Schema (Grid Laboratory Uniform Environment, see <http://www.cnaf.infn.it/~sergio/datatag/glue/index.htm>). The GLUE schema is available at <http://cvs.infn.it/cgi-bin/cvsweb.cgi/datatag-glue/glue-schemas/>.

The schema in grid projects is not very well documented, compared with the schema specifications in IETF RFCs, you often can find only the OpenLDAP schema configuration file, sometimes including some comments.

4.5.2. The EU Project OASIS

"The OASIS software is intended to perform intelligent information search and delivery service based on artificial neural network techniques and methods. To overcome the lack of experience and knowledge in the field of distributed fuzzy search strategies and algorithms, an experimental implementation of an information search and delivery server was developed." (<http://www.oasis-europe.org>)

The Project defined LDAP schema used to describe a Collection description entry in the OASIS Directory. (see <http://www.oasis-europe.org/docs/en/d0305/node84.html>)

4.6. LDAP schema of Open Source Projects

In a number of open source development projects that found a wide importance, LDAP technology is used as basis, or there exist at least LDAP enabled features.

In a first survey, a number of such projects have been found that contributed relevant and well documented schema. Just to name a few of them:

- Sendmail (www.sendmail.org) is the de facto standard implementation of an Mail Transfer Agent (MTA) on Unix and Linux platforms. Since Version 8.10 it includes the ability to route email traffic based on information stored in an LDAP server (see <http://www.iconimaging.net/~jradford/sendmail/sendmail-ldap.html>), by using the long expired Internet-Draft [Laser]. Other MTAs, like qmail support the same (see <http://www.lifewithqmail.org/ldap>).
- The Samba project (see above) includes a module for using LDAP as central user management tool, which is documented (including the OpenLDAP schema) in [Coupeau]. Additional schema has been defined for Netscape Directory Server 5.x and for IBM Secureway schema.
- There is a patch for the open source implementation of RADIUS (Cistron RADIUS) that allows the RADIUS server to authenticate users and retrieve RADIUS attributes from an LDAP server. The respective LDAP schema can be found at: <http://works.agni.com/cistron-ldap.html>. Similar LDAP patches exist for other RADIUS implementations like FreeRadius.

- BibTex is bibliography manager for the LaTeX typesetting language (see <http://www.latex-project.org/>). There are two projects that define an LDAP schema for storing BibTex data into an LDAP server: one documented at <http://mbdyn.aero.polimi.it/~masarati/ldap2bibtex.html>, the other documented in [Klasen].
- An implementation of the software packet manager Pacman (<http://physics.bu.edu/~youssef/pacman/index.html>) called PIPPY (Pacman Information Provider in PYthon, see <http://heppc12.uta.edu/~mcquigan/pippy/>) uses LDAP for managing the software packages. It is based on the Globus toolkit and is also used in Grid projects. The schema is available at <http://nut001.bu.edu/atlasgrid/installation/pippy-0.2/pip.schema>.

All these schemata have the problem that it is unsure how stable they are. The schema registry will need a good versioning mechanism, if it will include such schema.

5. Prior approaches to schema registry problem space

In the X.500/LDAP sphere as well as in other already mentioned technologies, there have been approaches to set up a schema registry. In addition there exist OID registries. This Section gives a survey on such approaches. Thus it also shows features from other schema registries that show what could be possible in an LDAP schema registry.

5.1. X.500 style schema administration

5.1.1. The subschema mechanism defined in X.500

Already in [X.501] (Par.14) a mechanism was defined for directory schema administration. Following object class for subschema subentry was defined:

```

subschema OBJECT-CLASS ::= {
    KIND auxiliary
    MAY CONTAIN {
        dITStructureRules |
        nameForms |
        dITContentRules |
        objectClasses |
        attributeTypes |
        matchingRules |
        matchingRuleUse }
    ID id-sc-subschema }

```

All attributes of this object class are defined as operational ("USAGE directoryOperation") to distinguish them from normal user attributes. operational attributes will not be displayed in normal user queries.

Subschema subentries can thus store and publish all schema elements in one entry. As Subentry the schema definitions can be defined for several parts of the DIT of a server. These parts are called administrative areas.

In addition subschema administrative capabilities are defined for the purpose of managing a DIT domain. They include:

- creation, deletion and modification of subschema subentries;
- support of the publication mechanism for the purpose of permitting X.500 servers to exchange schema information and X.500 clients to retrieve subschema information;
- subschema regulation for the purpose of ensuring that any modify operations will be performed in accordance with the applicable subschema specification.

A very similar mechanism has been included in the LDAP [LDAPSsubentry] and is implemented e.g. in OpenLDAP. An attempts were made to also specify procedures for merging, updating and removing LDAP schemata [LDAPSchUpdate] and for identifying different schemas in effect across a directory name space [LDAPSchPart].

It has to be noted that this mechanism, relying on operational attributes was only intended for the communication of X.500 servers and clients, it was not intended as a publishing mechanism that could be used by users to find schema.

5.1.2. The alternative mechanism of RFC 1804

Since these mechanisms of the 1993 Version of the X.500 Standard have not been implemented for a long time after their publication, and the community used implementations of the 1988 version X.500, schema information was passed between X.500 servers as textfiles via FTP. An attempt to specify this procedure [X500Schema] expired.

[RFC 1804] specifies a solution to this schema distribution problem using the existing mechanisms of the directory. It presents a naming scheme for naming schema objects and a meta-schema for storing schema objects and describes procedures for fetching unknown schema from the directory.

The naming scheme consist of a fixed value of the commonName attribute: "cn=subschema". The schema defined in such named entries is ruling the subtree that starts directly below the parent entry of the subschema entry. "All schema information relevant to that naming context is stored below the subschema entry. Children of the subschema entry store information about objects, attribute types, attribute syntaxes or matching rules." For this, the document specifies a meta schema with the following attribute types and object classes for the single schema elements (beware for clarity and shortness this is a non parsable abridged form of the ASN.1 definitions without oids!)

```
subschema OBJECT CLASS
  Subclass of TOP
  MUST CONTAIN {commonName}

objectClass OBJECT CLASS
  Subclass of TOP
  MUST CONTAIN {objectIdentifier}
  MAY CONTAIN {commonName,
    mandatoryNamingAttributes,
    mandatoryAttributes,
    optionalNamingAttributes,
    optionalAttributes,
    obsolete,
    description,
    subClassOf}

attributeType OBJECT CLASS
  Subclass of Top
  MUST CONTAIN {objectIdentifier}
  MAY CONTAIN {commonName,
    constraint,
    attributeSyntax,
    multivalued,
    obsolete,
    matchRules,
    description}

matchingRule OBJECT CLASS
  Subclass of Top
  MUST CONTAIN {objectIdentifier}
  MAY CONTAIN {commonName,
    matchtype,
    description,
    obsolete}

objectIdentifier ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX objectIdentifierSyntax

mandatoryNamingAttributes ATTRIBUTE
```

```

WITH ATTRIBUTE-SYNTAX SET OF OBJECT IDENTIFIER

mandatoryAttributes ATTRIBUTE
WITH ATTRIBUTE-SYNTAX SET OF OBJECT IDENTIFIER

optionalNamingAttributes ATTRIBUTE
WITH ATTRIBUTE-SYNTAX SET OF OBJECT IDENTIFIER

optionalAttributes ATTRIBUTE
WITH ATTRIBUTE-SYNTAX SET OF OBJECT IDENTIFIER

obsolete ATTRIBUTE
WITH ATTRIBUTE-SYNTAX BOOLEAN DEFAULT FALSE

subClassOf ATTRIBUTE
WITH ATTRIBUTE-SYNTAX SET OF OBJECT IDENTIFIER

constraint ATTRIBUTE
WITH ATTRIBUTE-SYNTAX Constraint

Constraint ::=Choice {StringConstraint,IntegerConstraint}

StringConstraint ::= SEQUENCE { shortest INTEGER,
                                longest INTEGER}

IntegerConstraint ::= SEQUENCE {lowerbound INTEGER,
                                upperbound INTEGER OPTIONAL}

attributeSyntax ATTRIBUTE
WITH ATTRIBUTE-SYNTAX ASN1DataType

multivalued ATTRIBUTE
WITH ATTRIBUTE-SYNTAX BOOLEAN DEFAULT FALSE

matchRules ATTRIBUTE
WITH ATTRIBUTE-SYNTAX SET OF OBJECT IDENTIFIER

matchtype ATTRIBUTE
WITH ATTRIBUTE-SYNTAX INTEGER {
    PRESENT                (0),
    EQUALITY                (1),
    ORDERING                (2),
    CASESENSITIVEMATCH     (3),
    CASEINSENSITIVEMATCH  (4) }

```

As Subschema subentry an entry with the object class subschema is used. Below that entry for every object class an entry with the object class object Class is used, for every attribute type an entry with the object class attribute type is used etc.

The advantages of this method to publish schema are:

- It uses directory technology for publishing directory related information. Thus directory implementations can access them with their standard access protocol.
- It uses normal user attributes, thus also users can access the schema information.
- the schema is published in a whole subtree of entries. This granularity eases the search for specific things (e.g. only object classes)

This mechanism can thus also be used as technology for a directory schema registry. The approach taken in this project will be similar to this approach.

5.2. The new IANA procedures for registering LDAP elements

[RFC 3383] defines best current practices of procedures for registering extensible LDAP elements and provides policy guidelines to the IANA describing conditions under which new values can be assigned.

Following LDAP elements can be registered at the IANA according to this document:

- LDAP message types
- LDAP extended operations and controls
- LDAP result codes
- LDAP authentication methods
- LDAP attribute description options
- Object Identifier descriptors.

Only the latter is directly related to schema elements, since most LDAP schema elements have an OID and one or more short descriptive names (or descriptors) that can be used instead of a numeric Object Identifier to identify it. It is important to register these descriptors to prevent a situation that the same descriptor is assigned to several OIDs. Following rules are specified for Object Identifier descriptors:

- Multiple names may be assigned to a given OID
- Descriptors longer than 48 characters may be viewed as too long to register.
- values ending with a hyphen ("-") reserve all descriptors which start with the value
- Descriptors beginning with "x-" are for Private Use and cannot be registered.
- Descriptors beginning with "e-" are reserved for experiments and will be registered on a First Come First Served basis.
- All other descriptors require Expert Review to be registered.

Following Template for registering descriptors is given in the document:

```
A.3. LDAP Descriptor Registration Template
Subject: Request for LDAP Descriptor Registration Descriptor (short name):
Object Identifier:
Person & email address to contact for further information:
Usage: (One of attribute type, URL extension, object class, or other)
Specification: (RFC, I-D, URI)
Author/Change Controller:
Comments: (Any comments that the requester deems relevant to the request)
```

[RFC 3383] also specifies the registration procedure, as following:

- fill out the appropriate form
- If the policy is Standards Action
 - provide the completed form to the IESG with the request for Standards Action
 - Upon approval the IESG forwards the request to IANA
 - IESG is viewed as the "owner"
- If the policy is Expert Review

- post the completed form to the public mailing list <directory@apps.ietf.org> for a public review period of 2 weeks, which starts again after each revised form posting.
- An Expert appointed by the Applications Area Director(s) approves or denies the request based on the public review
- On approval the Expert forwards the request to the IESG
- the requester is the "owner"
- If the policy is First Come First Served
 - the requester submits the completed form directly to the IANA iana@iana.org.
 - the requester is the "owner"

According to [RFC 3383] IANA makes the list of registered values available on the web site, which has not taken place yet. The "owner" can update the registered value to the same constraints and review as with new registrations. Comments can be attached to the registration upon Expert review if there are significant objections that the "owner" does not solve through a update.

A great merit of [RFC 3383] is that it lists all currently assigned values (and thus makes the publication of the lists on the IANA web site not too urgent) in an Appendix. B, which lists elements of the following specifications:

[RFC1274], [RFC1488], [RFC2079], [RFC2164], [RFC2247], [RFC2252], [RFC2253], [RFC2256], [RFC2293], [RFC2587], [RFC2589], [RFC2739], [RFC2798], [RFC3296], [X.501].

[RFC 3383] shows that other elements of LDAP need also to be registered, besides the schema elements dealt within this project. For LDAP extensions see also [LDAPExt]. Another LDAP element that would be worthwhile to register is server feature [LDAPFeature].

5.3. The proposal of the IETF Schema Working Group

5.3.1. Documents produced by the IETF Schema WG

The IETF WG called schema was established to provide specifications for a schema listing service for the directory technologies LDAP, Whois, Whois++ and Rwhois. The idea was to provide a single point of discovery, to promote reuse, reduce duplication of effort and to promote interoperability. This work is based on a document [RFC 2425] that defines a MIME Content-Type for holding directory information.

A number of drafts were produced by the group in 1998, those relevant for LDAP were:

- A requirement document [SchemaReq], "for listing directory services schema in a centrally operated, administered, and maintained repository. This repository will be available as a resource to directory protocol and service implementors to facilitate schema discovery." The service envisioned was a schema listing service:
 - with public read access and restricted, moderated write access
 - centrally operated, administered, and maintained
 - largely automated, with minimal human involvement (like reviewing schema listing requests on a mailing list prior to publishing in the listing repository)
 - it "SHALL maintain information about schema units, beyond their definition. This information is referred to as metadata and will consist of information used for cataloging listings in the repositories."
 - The listing service SHALL maintain information about schema units, beyond their definition (metadata)

- "All versions of all listings MUST be retained. A simple method for getting the most recent version of a particular listing MUST be provided."
- accessible via FTP, HTTP, and SMTP. Updatable via SMTP
- language tags as specified in [RFC1766] MUST be used in all listings.
- Metadata element values MUST be encoded using the UTF-8 (UCS Transformation Format - 8 bit) form [RFC2044].
- a document on metadata [SchemaMeta] that defines a MIME directory profile for content transfer and encoding of metadata elements used for cataloging schema listings
 - Defining following profile types: listingName, listingTitle, listingUse, specFile, relatedTo, contactLanguage, contactName, contactEmail, contactPhone, contactAddress, authLanguage, authName, authEmail, authPhone, authAddress, specURL, security, created, moreInfo, caveat, listingComments, schemaPak, pakMember.
 - a discussion of these types can be found in [RegSchema].
- a document on Filenames [SchemaFile] that specifies a file name syntax for use by the primary listing repository operator of the directory schema listing service.
 - The proposed scheme is an OID scheme in the form "sequence.listversion.type", where:
 - *sequence* "consists of a serial number generated by the primary listing repository operator and is unique within the context of the schema listing service".
 - *listversion* "represents the version number of a listing within the context of the schema listing service", where "0" or "current" is equivalent to the most current version.
 - *type* consists of a token or number representing a file type. Following types are relevant in the context of LDAP:
 - 1: ldap, being "a related or grouped set of object attributes that form a discrete unit within the context of a schema for LDAP.
 - 2: pak-ldap, being "a related or grouped set of schema units that collectively specify a schema associated with LDAP"
 - 0: meta-unit for schema unit metadata, being "characteristics that differentiate one schema unit or schema pak from another"
 - With such inherent semantics the filename can be used as search criteria.
 - Since the planned schema registry will be rather webbased with the information stored in an LDAP database, these file name specifications that make more sense in a file-based service that was conceived by the schema WG, will not be supported in the currently planned schema registry.
- a document on procedures for running the schema repository [SchemaProc], that specifies "schema listing procedures which use the Internet Directory Consortium as the primary listing repository". The Internet Directory Consortium (IDC) is not defined in the IETF schema documents. There was the attempt to set up such a IDC (see 5.2.4.)
- a document [RFC 2927] that defines a multipurpose internet mail extensions (MIME) directory profile for holding an LDAP schema and how to register such schema in the schema listing service (see below)

Although only [RFC 2927] was standardised as an informational RFC the other documents also provide useful input for the design of the Directory Schema Registry.

5.3.2. Specifications of a MIME Directory Profile for LDAP Schema

As stated [RFC 2927] was the only document of the schema WG that was finalised and published as RFC. It specifies the following:

1. a syntax for schema definition (1.3.6.1.4.1.1466.115.121.1.56 DESC 'LDAP Schema Definition'):

```
LdapSchema = "(" whsp
  numericoid whsp
  [ "NAME" qdescrs ]
  [ "OBSOLETE" whsp ]
  [ "IMPORTS" oids ]
  [ "CLASSES" oids ]
  [ "ATTRIBUTES" oids ]
  [ "MATCHING-RULES" oids ]
  [ "SYNTAXES" oids ]
  whsp ")"
```

THE "IMPORTS" field lists the OIDs of other schemata which are to be incorporated by reference into the schema.

2. LdapSchemas attribute type definition

```
( 1.3.6.1.4.1.1466.101.120.17 NAME 'ldapSchemas'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.56 USAGE directoryOperation )
```

3. as well as registration forms for :

- The "schema-ldap-0" MIME Directory Profile Registration Profile types (all but LdapSchema multivalued)
 - SOURCE
The "SOURCE" type is optional, and if values are present they SHOULD be URIs of the "ldap" form (which point to a subschema entry)
This type is defined in [RFC 2425] as: to "identify the source of directory information contained in the content type"
 - LdapSchemas
 - attributeTypes
 - matchingRules
 - objectClasses
 - matchingRuleUse
 - LdapSyntaxes

With these specifications it is possible to send a schema definition via email, http, or other MIME aware protocols to a schema registry.

5.3.3. The MIME Content-Type for Directory Information

The profile specified in [RFC 2927] is based on [RFC 2425] which specifies the MIME Content-Type for Directory Information.

Following is specified by [RFC 2425]:

1. Content-Type: text/directory; with descriptions of parameters and encoding
2. Some a number of basic Predefined types (SOURCE, NAME, PROFILE, BEGIN and END)

3. The IANA registration for new profiles. Following Profile attributes are defined:
 - Profile name:
 - Profile purpose:
 - Profile types:
 - Profile special notes (optional):
 - Intended usage: (one of COMMON, LIMITED USE or OBSOLETE)
4. Following process is defined:
 - post the profile definition
 - at least two weeks discussion on a mailing list in which consensus must be reached
 - registration application at the Profile Reviewer appointed by the Application Area Directors and can either accept or reject the profile registration. Following reasons for rejection are given:
 - "1) Insufficient comment period;
 - 2) Consensus not reached;
 - 3) Technical deficiencies raised on the list or elsewhere have not been addressed"
5. A new type has to be described with following attributes:
 - Type name:
 - Type purpose:
 - Type encoding:
 - Type valuetype:

"The format a value of the type must have in the body of a text/directory MIME Content-Type. This description must be precise and must not violate the general encoding rules"
 - Type special notes (optional):
 - Intended usage: (one of COMMON, LIMITED USE or OBSOLETE)
6. A process for type changes
7. Similar registration processes for new parameters of a type, for parameter changes and for new value types

An important usage of the specification of Content-type text/directory is to pass around vcard information (the electronic business card) as specified in [RFC 2426].

5.3.4. The Open Group attempt to run a schema WG defined service

The Open Group intended to run a registry based on this technology (see <http://www.openldap.org/lists/ietf-ldapext/199810/msg00086.html>). They set up the Internet Directory Consortium and searched for members and for funding to run the schema listing service, but obviously were not able to find the respective funding. The Open Group now links the former IDC webpage to The Directory Interoperability Forum (<http://www.opengroup.org/directory/index.htm>), but no reference to any schema activity can be found there. Thus one can deem the activity to set up this schema listing service as dead.

5.4. Other IANA registries

Any kinds unique parameters and protocol values relevant in the Internet are registered at the IANA, from simple numbers like unique port numbers or error codes to more complex data like MIME types.

The IANA was thought of as registry for profiles of the MIME Content type text/directory [RFC 2425], as well as for LDAP schema [RFC 2927] as conceived by the schema WG (For both see above).

Other IANA registries are interesting for this project mainly with respect to the registration procedures, that may influence [RegPolicy].

In the following two such registries are discussed, the well established MIME media type registry and the currently defined registry for message headers.

5.4.1. Registration procedures for MIME media types

The IANA registration procedures for MIME media types, external body access types, and content-transfer-encodings are specified in [RFC 2048], whereby a number of registration trees are distinguished: IETF Tree, Vendor Tree, Personal or Vanity Tree, Special 'x.' Tree (for experimental media types and Additional Registration Trees. A web form for this registration process is available at <http://www.iana.org/cgi-bin/mediatypes.pl>. Following requirements and procedures described are noteworthy in the context of schema registration:

- universal support and implementation of a media type is NOT a requirement for registration.
- Proposals for media types registered in the IETF tree must be published as RFCs
- Other than in the IETF tree, the registration of a data type does not imply endorsement, approval, or recommendation by IANA or IETF or even certification that the specification is adequate.
- Following procedure has to be followed for the IETF tree of media types:
 - Send a proposed media type registration to the ietf-types@iana.org mailing list for a two week review period to solicit comments and feedback.
 - Media types registered in the IETF tree must be submitted to the IESG for approval.
 - "Provided that the media type meets the requirements for media types and has obtained approval that is necessary, the author may submit the registration request to the IANA, which will register the media type and make the media type registration available to the community."
 - Comments on registered media types may be submitted by members of the Internet community to IANA, which if possible will pass them on to the "owner" of the media type. Submitters of comments may request that their comment be attached to the media type registration itself. Whether this is done is decided by IANA.
- Similar procedures are defined for change control of media types, for external body access types, and content-transfer-encodings.

5.4.2. Registration procedures for message header fields

Currently work is being done to specify a new registry at IANA for registering message header fields [MsgHdrRegistry]. A registration process similar to [RFC 2524] is being proposed:

"The registration template is submitted for incorporation in one of the IANA message header field repositories by one of the following methods:

- An IANA considerations section in a defining RFC, calling for registration of the message header and referencing information as required by the registration template within the same document. Registration of the header is then processed as part of the RFC publication process.
- Send a copy of the template to the designated email discussion list. Allow a reasonable period - at least 2 weeks - for discussion and comments, then send the template to IANA at the designated email address. IANA will publish the template information if the requested name and the specification document meet the criteria [...], unless the IESG or their designated expert have requested that it not be published [...] IESG's designated expert should confirm to IANA that the registration criteria have been satisfied.

When a new entry is recorded in the permanent message header field registry, IANA will remove any corresponding entries (with the same field name and protocol) from the provisional registry."

It is noteworthy that a distinction between a provisional and a permanent registry is made.

Thus there are a number of specifications for registry procedures for the IANA, which all are similar in nature. For standards elements a mailing list discussion and a formal IESG approval are required. Basic sets of metadata are defined, content is input in general metadata fields like purpose. No keywords, with or without controlled vocabulary, classification schemes etc. are used, that are prerequisites for a useful search interface.

5.5. LDAP Schema Viewer

After the attempts for establishing an RFC 2927 based schema registry failed there was a so called grass root approach from an individual in Hong Kong to publish existing LDAP schema. After a period where it seemed that this attempt also died, it was revived and is now available under the name LDAP Schema Viewer at <http://ldap.akbkhhome.com/>.

This service provides a number of schema definitions via a PHPbased web interface. Following services are provided:

- Object
- class Viewer (alphabetical list of 47 object classes and name forms)
- Objectclass Tree (the same list arranged according to the inheritance, e.g.:


```
[+] person
    [+] organizationalPerson
        [+] inetOrgPerson
        [+] uidOrganizationalPersonNameForm
    [+] residentialPerson
```
- Attribute Viewer (alphabetical list of 122 attributetypes)
- Syntax Viewer (alphabetical list of 43 syntaxes)
- Matching rule viewer (alphabetical list of 24 matching rules)
- web formulars to add comments

Each list item is a link to a web page which includes

- a link to a local copy of the RFC where the schema element is specified
- a short explanation quoted from the RFC without pointing to the paragraph
- the BNC definition
- comments made from users and
- the web form for new comments.

In addition:

- attribute type pages contain all information about the syntax
- object class pages contain an example entry and all information of all attribute types included in the class

Thus a very handy service is provided to browse LDAP schema elements. There are a few limitations though that distinguishes this service from a schema registry as proposed in this project:

- Only a limited number of RFCs have been included, other schema is not included.
- It is not made explicit which RFCs have been included. Following documents are available on the site, which thus might be the list of texts included: [RFC 1617], [RFC 2247], [RFC 2252], [RFC 2256], [RFC 2307], [RFC 2377], [RFC 2798], [RFC 3112].
- There is no policy defined what else to include. The site just says: "Send me suggestions on RFC's to add". There needs to be a specification of the registry procedures and criteria for inclusion and classification of schemata and their stati.
- The valuable comment mechanism lacks a structure as well; it needs a policy for comment inclusion and a possible distinction between different comment types.
- There is no search interface to find schema elements via their name or their content.
- The schema can only be retrieved via HTTP. It is a service only for human users
- A careful data modelling for schema metadata could provide enhanced usability as well as the possibility to include other than LDAP schema.

5.6. Novell schema registry

Novell maintains an own OID registry where NDS (eDirectory) developers can register schema extensions before they release a product that extends the NDS schema. All schema object and attribute class definitions must be unique to the NDS tree to prevent name collisions.

To register schema one has to fill in a web form at <http://developer.novell.com/support/sform.htm> and pay a fee. Besides contact name, affiliation and address, the web form asks for following data:

- Requested Name Prefix: (8 alpha/numeric characters)
- Name Prefix Type: (either "product specific" or "general")
- Product Name (if product specific)
- whether to register an OID under the Novell Sub-registry (Novell will provide the OID) or an developer supplied OID and in the latter case the associated OID.

After payment one receives a unique prefix that can be used for naming new attribute and class definitions. One also receives two OIDs, one for object classes and one for attribute definitions. Once schema extensions are registered, they become available for other people to use, which also includes e.g. to extend it with new attributes.

Thus this is only a naming and OID registry. The schema definitions themselves are neither stored nor evaluated, no syntax checks is being done by the registry. For more information on the Novell registry see <http://developer.novell.com/support/schreg2c.htm>. A repository of schema registered with this mechanism could not be found.

5.7. The need for schema registries in grid computing

In the GGF an effort was made to specify the Grid Object Specification (GOS) language [GOS], for specifying object classes, which are used to specify the contents of entities that pertain to the Grid and grid-based applications. It was defined in a generic way that allowed for the construction of automated translators that can generate implementation specific forms like LDAP, XML and SQL

syntax. The guidelines on defining schema in GOS are more strict than in LDAP, e.g. the description field is mandatory to be filled in. In the frame of the discussions on GOS the need for a schema repository was felt.

One concept that got important in grid computing is the concept of a virtual organization (VO), that has "geographical dispersion of organizational units" and an "electronic linking of production process" as already defined by [Travica]. In Grid computing such a VO can consist of different Universities that work on the same problem space together using shared resources. Some schema will be only relevant in one VO, so a schema registry for every VO was proposed.

[Allan] describes how an UK e-Science UDDI registry might be used to register information about e-Science Virtual Organisations and to enable inter-working between them by exposing their contacts and service points.

5.8. OID registries

Two OID registries can be found on the web, which will be discussed in this section.

5.8.1. Object Identifier Registry of Harald Alvestrand

Harald Alvestrand since a long time maintains an OID registry at www.alvestrand.no/objectid. The entry page gives a short introduction to OIDs and then links to the hierarchically organised OID tree as well as to a web form for entering new OIDs (<http://www.alvestrand.no/objectid/form.html>) and to a list of submissions not yet included into the site (<http://www.alvestrand.no/objectid/submissions/>). Following data can be input into the web form:

- OID value ("Use dot notation such as 1.2.567.34. Do NOT use names")
- Title ("A descriptive title for the OID such as 'Microsoft Excel' or 'RSA Encryption'")
- Description (A free text field)
- URL that points to further information
- Email address (of the submitter)

The OID tree pages consist of often very elaborate descriptions of the tree node, as well as of references to all superior nodes and to the direct subsidiary nodes. A search interface to search below the current node is attached to each page.

Being a registry for all OIDs, not only for LDAP schema elements, it has a broader scope, the registry is not complete and it is dependent on submissions from interested users. According to personal communication, the registry is still maintained, but in a "low-key activity". Because of the interesting comments, it makes sense to link to this registry if LDAP elements included in this project have respective entries there.

5.8.2. The Object identifier tree of ASN.1.Information site

The maintainer of an ASN.1 information site at (<http://asn1.elibel.tm.fr/en/index.htm>), also maintains a web based OID browser. Again you can find an interesting introduction into OIDs and a hierarchical structured site with a web page for each OID node. These pages again have links to all superior nodes and to all immediate subordinate nodes. Every node is described with the following fields:

- oid with the alphanumeric representation of the OID, e.g.: "{iso(1) identified-organization(3) dod(6) internet(1) private(4) enterprise(1)}" for 1.3.6.1.4.1
- Node type, either "node" or "leaf"
- Description, a short description of the node
- Information, additional information about the node, including links to other resources

- Creation date, of the web page in this tree (only for leafs?)

Where such information is known additional data about the registrant are displayed:

- current registrant name and email address
- postal and web address
- telephone
- fax

The site seems to offer the possibility of automatically creating a subnote or a node at the same level, but when trying out this feature only a server error saying "The url indicated does not exist on this server" appears.

In addition to the described text oriented browser with one web page per oid node, the service also provides a graphical hierarchical view called tree view which gives access to the whole tree in the fashion, modern operating systems give access to the file directory hierarchy. Because of the additional information, not to be found at the site of Alvestrand, it also makes sense to link to the nodes of this site.

5.9. XML, RDF and related registries

Not only in the realm of directories the need for schema registries is seen. Especially for the technologies XML Schema and RDF Schema, new registries have been created to fulfil these needs. This section gives short descriptions of a few of these approaches. An interesting introduction to XML schema registries can be found in [Blake].

5.9.1. XML.org registry

The XML.org was set up by OASIS in June 1999 to minimize overlap and duplication in XML languages and XML standard initiatives by providing public access to XML information and XML Schemas. A major function is the maintenance of an XML schema registry (<http://www.xml.org/xml/registry.jsp>). The site gives following description of this registry:

"The XML.org Registry is a community resource for accessing the fast-growing body of XML specifications being developed for vertical industries and horizontal applications. The XML.org Registry offers a central clearinghouse for developers and standards bodies to publicly submit, publish and exchange XML schemata, vocabularies and related documents. Operated by OASIS - the non-profit XML interoperability consortium -- the XML.org Registry is a self-supporting resource created by and for the community at large." (http://www.xml.org/xml/registry_about.jsp)

The Site gives access to XML DTDs ("old style" Document Type Definitions) as well as to XSD (XML Schema Definitions) files. You have to be registered to submit new schema.

The search interface provides following fields to search for:

- Keyword: (free text)
- Industry (a fixed list, see below)
- Last Update: list, e.g. "Since inception", "last month", "last 3 months", etc.
- Document Type: list: "All Schema Types, XSD, XDR, DTD, XSL, Code Sample, Specification, other"

Currently over 200 schemata have been registered in the following industry areas:

Accounting (7)	ERP (1)	Real Estate (2)
Aerospace (1)	Education (9)	Robotics/AI (2)
Arts/Entertainment (6)	Energy/Utilities (24)	Science (8)

Astronomy (4) Automotive (4) Banking (1) Biology (4) Business Reporting (1) Business Services (1) Catalogs (1) Chemistry (3) Construction (2) Customer Relation (2) Databases (1) E-Commerce (9) EDI (2)	Financial Service (21) Food Services (2) Geography (2) Healthcare (6) Human Resources (9) Industrial Control (1) Internet/Web (7) Manufacturing (4) Multimedia (6) News (3) Other Industry (5) Publishing/Print (3)	Security (1) Social Sciences (1) Software (10) Supply Chain (5) Telecommunications (5) Transportation (2) Travel (1) Weather (1) XML Technologies (19)
---	---	--

This service shows how content search criteria can be used in a registry.

For posting new schema, it says you have to be registered for free. The attempt to register lead to a server error.

According to [Blake], "Sun, IBM, Oracle, SAP and other large corporations supported this effort through significant investment of \$100,000 each to sponsor XML.org. However, with one full time and one part time staff person, neither of whom were technical, there was still a question as to whether OASIS had the infrastructure or bandwidth to create a fully functional registry site".

This scepticism seems to have been baseless, since by now XML.org has developed to such a fully functional registry site and a well accepted service.

[Blake] also describes Microsoft's competing XML schema registry called BizTalk.org which was created in September, 1999, to encourage the exchange and definition of XML-based documents. It seems that XML.org won this competition, since Microsoft shut down the BizTalk.org web site in summer 2002. If you now type in the biztalk.org URL, you will be led to the Microsoft product BizTalk Server, an enterprise software for application integration.

5.9.2. DESIRE Metadata Registry

In the EU-funded project DESIRE (www.desire.org) one deliverable was to set up a metadata registry:

"Metadata registries enable authoritative information about metadata schemes to be declared and thus support the extensibility and evolution of element sets and provide some basis for interoperability. The DESIRE metadata registry demonstrates how a metadata registry might work. Elements from several different metadata element sets, including Dublin Core, have been added." [Heery]

The pilot registry created in the DESIRE II project does not seem to be run anymore. The URL of the registry (<http://desire.ukoln.ac.uk/registry/>) is unreachable.

[Heery] describes the registry as follows:

- The DESIRE registry offers both search and browse interfaces for navigating the registry.
- The index provides access to the browse and search interfaces for each of the entities that can be registered, and to the page for generating cross-walks.
- All of the registered entities of a particular type, for example namespaces, can be viewed via the browse interface.
- The search interface for a particular entity type can be accessed from the Index page. Equality and substring (contains) searches are supported for each entity type.
- In addition to basic metadata registry functionality, the DESIRE metadata registry aims to provide mappings between different metadata vocabularies. Instead of mapping between

every pair of vocabularies, every vocabulary is mapped onto an underlying semantic layer. This achieved by using Basic Semantics Register BSR), an ISO standard that identifies and defines semantic components for use in data exchange.

- The prototype implementation does not provide support for end-user registration of new elements

5.9.3. German Metadata Registry

"The purpose of the German Metadata Registry is to provide an overview of the metadata efforts and implementations within Germany and German-language areas. This site does not constitute an official or obligatory registration site for metadata, but is rather a collection of materials to help those interested in implementing metadata, as well as giving others an overview of the metadata activities in the German language area. It is also intended as an instrument to further discussion and development of standards regarding metadata in terms of accepted definitions, substructures, usage, and limitations." (<http://www.mpib-berlin.mpg.de/en/institut/dok/metadata/qmr/Gmr1e.htm>)

Basically this registry is an information base about the usage of DC in different subject areas (Education, Medicine, Physics and Mathematics).

5.9.4. DDDS (Distributed Data Dictionary Service)

"In an interconnected world, applications need to interconnect in much the same way that computers are connected today: i.e. through the use of well established standards-based technologies. Data dictionaries are the keys to these applications outlining the meaning and structure of the information contained within them. What is currently missing is an architecture and mechanisms that allows data dictionaries to be systematically linked to each other thus enabling connectivity between applications." [U'Ren]

The DDDS uses a combination of DC, LDAP and the ISO/IEC 11179 Data Element Set for setting up a registry service for vocabularies, data elements and data models (see <http://step.jpl.nasa.gov/ldap>).

5.9.5. RDF Schema registries

Also for RDF schema there is the need of registries and there are a number of approaches, that will only be listed here for future reference:

- RDF Schema Registry at FORTH Institute of Computer Science (<http://schemas.amberarcher.org/registry/core/xml/req-config-doc.html>), providing links to a variety of important RDF schemata
- An registry is planned in [XMLRegistry], which describes an IANA maintained registry for IETF standards which use XML related items such as Namespaces, DTD, and RDF Schemata.
- An interesting project worth mentioning here is an RDF based multilingual registry for metadata schemata called "ULIS Open Metadata Registry" available at (<http://avalon.ulis.ac.jp/registry/>) and described in [Nagamori]

5.9.6. The EOR Toolkit

Another project worth mentioning is the EOR Toolkit project (<http://eor.dublincore.org/>), that provides open source software for storing RDF data.

"The goal of the EOR (Extensible Open Rdf) Toolkit is to seamlessly integrate all the components needed to build a generic web search interface of RDF Models. This toolkit as such provides the basic building blocks for supporting search services, topic-maps, site-maps, annotation environments and semantic metadata registries based on RDF.

The base level functionality provided by this toolkit include:

- Creation, deletion, and management of RDF databases.
- Ability to infuse RDF instance data into RDF databases.
- Ability to search RDF databases.
- Generic interface design capabilities to support RDF applications."
(<http://eor.dublincore.org/>)

6. LDAP schema standardisation and a new schema registry

This document provided an introduction into technologies and organisations relevant in the schema registry problem space, a survey of existing LDAP schema and approaches for setting up schema registries for LDAP and other technologies.

It was shown that the problem of schema that has to be well published for reuse is the same for LDAP schema registries and other schema registries (e.g., for XML schema or RDF schema).

All registries found have to be seen as preliminary. It seems none of them provide retrieval possibilities for both humans and applications. The problem of content classification is also mostly ignored. The XML schema registry at XML.org being the only one with a quite simple classification with respect to industrial area.

For the LDAP realm considerable work has been done on schema publication via the LDAP protocol itself, that allows applications to retrieve supported schema. The only more elaborate user oriented approach aside from the ASCII file model of IANA is the LDAP schema viewer, which thus provides a very helpful tool. It cannot be taken as proper registry, since as listed above, essential features are missing.

What still is missing is an up to date registry of LDAP schema elements which fulfils the following requirements:

- it has to contain comprehensive amount of relevant LDAP schema
- it has to contain metadata about the defined schema in addition to the schema definition itself to make it easily searchable
- It has to provide a convenient search and browse interface for humans
- It has to provide an access functionality via the LDAP protocol itself for LDAP applications. Current work in this field has to be taken into account.
- It has to provide a handy MIME based interface for schema delivery via SMTP and HTTP, as specified in [RFC 2927]
- it has to provide a web form based interface for schema delivery including additional data not specified in RFC 2927
- it should provide links to the existing OID registries, where appropriate
- It has to be run according to a well documented policy for
 - criteria for inclusion or exclusion
 - schema inclusion process
 - schema update process
 - comments facility

This proposed project wants to fill this gap and deliver an LDAP schema registry that fulfils all these requirements..

7. Pointer to References

[RegBib] Gietz, Peter, "Bibliography for the Directory Schema Registry Project",
Version 1, Deliverable B of the TERENA project Directory Schema Registry,
January 2003

[RegBib] contains all references of the project documents.

A. Glossary of used acronyms and technical terms

Acronym or technical term	subsection	page
ABNF (Augmented Backus-Naur Form)	3.1.3.	3
ASN.1 (Abstract Syntax Notation One)	3.1.2.	3
Attribute syntax	3.2.5.	6
Attribute type	3.2.3.	4
CIM (Common Information Model)	3.3.9.	13
DC (Dublin Core)	3.3.5.	11
DIB (Directory Information Base)	3.2.1.	4
Directory entry	3.2.2.	4
DIT (Directory Information Tree)	3.2.7.	7
DIT content rules	3.2.12.	8
DIT structure rule	3.2.11.	8
DMTF (Distributed Management Task Force)	3.5.8.	16
DN (Distinguished Name)	3.2.9.	7
DSML (Directory Service Markup Language)	3.3.2.	10
EDUCAUSE	3.5.12.	17
GGF (Global Grid Forum)	3.5.10.	16
IAB (Internet Architecture Board)	3.5.2.	15
IANA (Internet Assigned Numbers Authority)	3.5.4.	15
IESG (Internet Engineering Steering Group)	3.5.3.	15
IETF (Internet Engineering Task Force)	3.5.1.	14
Internet2	3.5.11.	16
ISO (International Standards Organisation)	3.5.6.	16
ISO/IEC 11179	3.3.6.	12
ISO/IEC 11179 Metadata Registry Implementation Coalition	3.5.7.	16
ITU (International Telephone Union)	3.5.5.	15
Matching rules	3.2.6.	6

Name form	3.2.10.	7
OASIS (Organization for the Advancement of Structured Information Standards)	3.5.9.	16
Object class	3.2.4.	5
OID (Object Identifiers)	3.1.1.	3
OWL	3.3.7.	12
RADIUS (Remote Authentication Dial In User Service)	3.4.2.	14
RDF (Resource Description Format)	3.3.8.	13
RDN (Relative Distinguished Name)	3.2.8.	7
Samba	3.4.1.	14
Subschema subentry	3.2.13.	9
The Open Group	3.5.13.	17
UDDI (Universal Description, Discovery and Integration)	3.3.4.	10
WSDL (Web Services Description Language)	3.3.3.	10
XML (Extended Markup Language)	3.3.1.	9

B. Detailed table of contents

1. Status of this document	1
2. Introduction	1
2.1. Structure of this document	2
2.2. Conventions used in this document	2
3. Definitions	2
3.1. Definitions of basic technologies for LDAP schema definition	3
3.1.1. OID (Object Identifiers).....	3
3.1.2. ASN.1 (Abstract Syntax Notation One)	3
3.1.3. ABNF (Augmented Backus-Naur Form).....	3
3.2. Definitions of LDAP schema elements	4
3.2.1. DIB (Directory Information Base).....	4
3.2.2. Directory entry	4
3.2.3. Attribute type	4
3.2.4. Object class	5
3.2.5. Attribute syntax.....	6
3.2.6. Matching rules	6
3.2.7. DIT (Directory Information Tree).....	7
3.2.8. RDN (Relative Distinguished Name)	7
3.2.9. DN (Distinguished Name).....	7
3.2.10. Name form.....	8
3.2.11. DIT structure rule.....	8
3.2.12. DIT content rules	9
3.2.13. Subschema subentry.....	9
3.3. Definitions of XML, Metadata and Ontology technologies	9
3.3.1. XML (Extended Markup Language).....	10
3.3.2. DSML (Directory Service Markup Language).....	10

3.3.3. WSDL (Web Services Description Language).....	10
3.3.4. UDDI (Universal Description, Discovery and Integration).....	10
3.3.5. DC (Dublin Core).....	11
3.3.6. ISO/IEC 11179	12
3.3.7. OWL	12
3.3.8. RDF (Resource Description Format).....	13
3.3.9. CIM (Common Information Model)	13
3.4. Definition of exemplary application technologies	14
3.4.1. Samba	14
3.4.2. RADIUS (Remote Authentication Dial In User Service)	14
3.5. Definitions of relevant organisations	14
3.5.1. IETF (Internet Engineering Task Force).....	14
3.5.2. IAB (Internet Architecture Board)	15
3.5.3. IESG (Internet Engineering Steering Group).....	15
3.5.4. IANA (Internet Assigned Numbers Authority).....	15
3.5.5. ITU (International Telephone Union)	15
3.5.6. ISO (International Standards Organisation)	16
3.5.7. ISO/IEC 11179 Metadata Registry Implementation Coalition.....	16
3.5.8. DMTF (Distributed Management Task Force).....	16
3.5.9. OASIS (Organization for the Advancement of Structured Information Standards).....	16
3.5.10. GGF (Global Grid Forum).....	16
3.5.11. Internet2	16
3.5.12. EDUCAUSE	17
3.5.13. The Open Group.....	17
4. Existing LDAP schema standards	17
4.1. X.500 schema standards	17
4.2. IETF LDAP schema standards.....	17
4.3. LDAP schema standards of other standards organisations	18
4.3.1. DMTF	18
4.3.2. Open Group.....	18
4.3.3. Internet 2/EDUCAUSE	18
4.4. LDAP schema de facto standards of proprietary directory software	18
4.4.1. Special schema for Novell	18
4.4.2. Netscape Directory Service	19
4.4.3. SUN Directory Server	19
4.4.4. Microsoft Active Directory.....	19
4.5. LDAP schema of research projects.....	19
4.5.1. Grid Computing projects.....	20
4.5.2. The EU Project OASIS	20
4.6. LDAP schema of Open Source Projects	20
5. Prior approaches to schema registry problem space	21
5.1. X.500 style schema administration	21
5.1.1. The subschema mechanism defined in X.500.....	21
5.1.2. The alternative mechanism of RFC 1804	22
5.2. The new IANA procedures for registering LDAP elements	24
5.3. The proposal of the IETF Schema Working Group	25
5.3.1. Documents produced by the IETF Schema WG	25
5.3.2. Specifications of a MIME Directory Profile for LDAP Schema.....	27
5.3.3. The MIME Content-Type for Directory Information.....	27
5.3.4. The Open Group attempt to run a schema WG defined service	28
5.4. Other IANA registries	29
5.4.1. Registration procedures for MIME media types	29
5.4.2. Registration procedures for message header fields	29
5.5. LDAP Schema Viewer.....	30
5.6. Novell schema registry.....	31
5.7. The need for schema registries in grid computing	31

5.8. OID registries	32
5.8.1. Object Identifier Registry of Harald Alvestrand.....	32
5.8.2. The Object identifier tree of ASN.1.Information site	32
5.9. XML, RDF and related registries.....	33
5.9.1. XML.org registry	33
5.9.2. DESIRE Metadata Registry	34
5.9.3. German Metadata Registry	35
5.9.4. DDDS (Distributed Data Dictionary Service)	35
5.9.5. RDF Schema registries	35
5.9.6. The EOR Toolkit.....	35
6. LDAP schema standardisation and a new schema registry	36
7. Pointer to References	37
A. Glossary of used acronyms and technical terms.....	37
B. Detailed table of contents.....	38