A Reference Schema for LDAP-based Identity Management Systems

Frank Tröger und Klaus Meyer-Wegener

Friedrich-Alexander-Universität Erlangen-Nürnberg

Regionales Rechenzentrum Erlangen

und

Technische Fakultät, Department Informatik,

Lehrstuhl für Informatik 6 (Datenmanagement)

In: German Unix User Group (ed.):

Proceedings of the 1st International Conference on LDAP

(LDAPcon, Cologne, Germany, Sept. 6th).

2007.

The proceedings of LDAPcon 2007 - together with those of ECAI6 2007 - are included in issue 3/2007 of GUUG's member magazine *UpTimes* that has ISBN 978-3-86541-228-7.

Abstract available via URL:

http://www.guug.de/veranstaltungen/ldapcon2007/abstracts.html#3_5_1

Abstract

This paper describes the experience about gathering and consolidating information about existing Lightweight Directory Access Protocol (LDAP) schemas. The current lack of a comprehensive summary and best practices are eliminated by a reference schema for LDAP-based identity management systems, comprising a superset of their attributes. The presented reference schema permits the comparison of standard and widely published schemas. Lossless where possible, it provides mappings for equivalent attributes with different syntaxes or different domains, e.g. emerged from additional constraints like enumeration types. New schemas for a specific case can be derived from the reference schema. Their mapping is given by the relationship to the reference schema.

The most important steps of the schema design process have been retained and written down. The paper provides a realistic example of this process and shows the simplification given by the use of the reference schema. A review of the schema design process in the project above, discusses some restrictions of this appliance. A prospective tool-based mapping for data exchange between different identity management systems completes the paper.

1. Introduction

To reconstruct the existent user management Regional Computing Center Erlangen (RRZE) at Friedrich-Alexander University Erlangen-Nuremberg started the project IDMone [15] in 2006. One of the early tasks was to find a useful schema for the new central directory service. In addition to the schema provided by the identity management (IDM) software, as many standard schemas as possible should be used.

The first step in a Lightweight Directory Access Protocol (LDAP) directory setup is the data design process. Detailed information is necessary concerning what kind of data elements should be stored in a directory. Afterwards the schema should be designed to fit the needs. Simply put, schema design involves the following four steps, which are derived from six ones given in [9]:

1. Locate application, standard and directory vendor-provided schemas

2. Choose other predefined schema elements

3. Develop schema extensions

4. Document the schema design

The first step is almost fixed by the applications planned to deploy and the schema already included in the directory system. Step two and three are much more arbitrary. Using public schemas in step two ensures as much compatibility as possible and avoids reinventing the wheel. But not every use case is covered by existing public schemas. Customization or development of a LDAP-enabled application often implicates the development of an own schema extension. The last step is as important as the previous ones. Schema documentation helps maintaining the directory and disburdens the development of future LDAP-enabled applications. This is additional reason for the preference of existing public schemas compared to proprietary ones. In most cases widely-published schemas are well documented and well-known beyond the boundaries of the individual enterprise.

Before choosing predefined schema elements as stated in step two, there is the retrieval for them. As part of the IDMone project, this drawback is meant to be eliminated for schema design in the sector of identity management. An IDM reference schema permits a comparison of published schemas, provides mappings for equivalent attributes and permits derivation of own schema extensions from it. The focus of the reference schema will be identity management systems for higher education because of the project's origin. The complexity in this area is higher than in normal organizations, e.g. a faculty member with appointments in multiple departments or a person who is a student in one department and an employee in another. In addition, there is more research work publicly available. Therefore the presented reference schema should be a superset for most IDM schemas.

The paper is organized as follows: Section 2 presents the most important public schemas and reviews existing tools, previous and related work. Section 3 gives a short report about problems in schema design process of the project. Section 4 describes the created reference schema. Section 5 draws conclusions and makes suggestions for further work.

2. Background

A good schema maintains the integrity and quality of the data stored, a bad one implies data redundancy, anomalies and retrieval of unwanted data. There are schemas for almost every purposes directories are designed to. But due to the history of origins of directories, schemas for the so called white pages are the most common ones. They enable the representation of individual persons within an organization. Nowadays they provide far more than simple address books, e.g. elements for organizational divisions, roles, groups and devices. The most important schemas for LDAP in this domain are given by some Requests for Comments (RFCs) from the Internet Engineering Task Force (IETF).

2.1. Standard Schemas from RFCs

The original LDAP schema for user management per se is defined in RFC 2256. It is the most common schema for LDAP/X.500 directories and represents a basis for many other white pages objects [22]. Beside other object classes, `person` and `organizationalPerson`, which is a subclass of the former one, are the important ones for user management. These two object classes provide a basic set of useful attributes. An interesting fact is, that not all in RFC 2256 defined attributes are used by the defined classes. In 2006 RFCs 4512, 4517, and 4519 replaced this RFC [24] [1] [17]. The main part of the attributes are covered in RFC 4519.

The next standard schema is defined in RFC 2798 [19]. It defines some additional attributes and the well known `inetOrgPerson` object class. This class is a subclass of `organizationalPerson` and therefore inherits its attributes. It includes several attributes defined in other RFCs (RFC 2256, RFC 1274, RFC 2079). This class is found in almost every user management directory and is effective, beside some small updates via RFCs 4519 and 4524.

The `inetOrgPerson` object class including its superclass chain (`organizationalPerson` → `person` → `top`) build the basis for almost every schema used in LDAP directories for user or identity management. LDAP directory software normally is set with these schemas included in its core schema. These schemas belong to those in step one of schema design process. Nevertheless it is important to know what elements they define.

2.2. Widely-Published Schemas

In 2000, Internet2, the foremost U.S. advanced networking consortium, convened the Middleware Architecture Committee for Education (MACE) to help creating a national interoperable middleware infrastructure for research and education. The MACE-Dir working group has the mission of defining an LDAP object class that includes common person attributes in higher education [11].

The actual specification (at present 200604a) of `eduPerson` can be found here [12]. It defines one auxiliary object class and ten attributes, all MAY attributes of the object class. Although there is no enforcement to use `eduPerson` in combination with `inetOrgPerson`, the task force recommends it. Three of the attributes establish the relationship of a person within an organization. This is necessary, if persons are stored flat in the Directory Information Tree (DIT) instead of hierarchically under their organizational division. Three other attributes map the affiliation(s) of the persons to the organization. `eduPerson` is becoming the de facto standard in higher education. But because of its focus on the U.S., there are still missing attributes for identity management systems in other regions.

A similar effort comes from the Trans-European Research and Education Networking Association (TERENA). Starting in 2004, the Task Force European Middleware Coordination and Collaboration (TF-EMC2) aims to provide a forum to discuss middleware issues and foster collaboration in the middleware arena [21]. Its project SChema Harmonisation Committee (SCHAC) provides a schema of the same name. The latest release of SCHAC (at present 1.3.0) can be downloaded from [20]. The schema refers to `inetOrgPerson` and `eduPerson`. In addition to assorted attributes for individual data, it uses ten categories, which were compiled from the NMI LocalDomainPerson survey and discussions with the International Schema Archives (Feb, 2004) [4]. Seven of the ten categories were instantiated by auxiliary object classes.

The schemas above have rather generic characteristics and try to provide a good basis for local extensions. Attributes are handpicked and should cover overall needed information instead of specific ones. Whereas the next schemas presented are designed for a single domain of purposes. The WA Libraries Authentication Project (WALAP) of the Western Australian Group of University Librarians (WAGUL) developed a schema for the distributed authentication infrastructure for the five western Australia universities [23]. The WALAP schema consists of object classes `auEduPerson` and `auEduUnit` and uses subtypes extensivly, i.e. almost every attribute is derived from a more generic standard attribute type.

The german project "Integrierende Benutzer- und Ressourcenverwaltung an den Thüringer Hochschulen (Codex – Meta Directory)" is quite similar to the Australian one. Codex defines the two object classes `thuEduPerson` and `thuEduRole` and it is the first schema in this paper with student-specific attributes like `thuEduStudentType` or `thuEduSemsterOfCourseStudy`. The broad usage of a minimum upper bound on the numbers of characters in values with string-based syntax is one thing to point out; most other schemas do not use it. A detailed documentation of the usage of schema elements, even those from other sources, is also remarkable.

Another german schema may be contributed by the Higher Education Information System (HIS). The federal government funded non-profit company HIS, which is conceived as an element of the German higher education system and for example supports universities in questions of administration. In the next release, some modules will provide an export of personal and organizational data via LDAP [8]. HIS software is deployed in the majority of German universities and the upcoming schema will also be present there.

There are further schemas publicly available and some may be added to the reference schema in the future. But due to the tight timetable of the project, a trade-off with respect to completeness has to be done. A more comprehensive list from 2005 can be found in [4].

2.3. Schema Tools Online

Miscellaneous websites in the internet already provide some kind of service around LDAP schemas. Three of them are picked up and described in a nutshell. The LDAP Schema Viewer by Alan Knowles [14] allows browsing of common LDAP object classes, attribute types, attribute syntaxes and matching rules. This is a good tool to get information about an already known schema element.

The Schema Registry [10] has been a result of the TERENA Directory Schema Registry (DSR) Project. Registered schemas are stored in a LDAP directory and the web interface is based on w2l, a web gateway written in Perl. It exists as a pilot service, which provides a browsable and searchable web interface. Schemas can be downloaded as LDIF files from the website directly. Unfortunately only standard schemas until 2003 are stored in the database. Since then no more schemas have been added to the registry.

Other websites offer access to the Object Identifiers (OID) tree, either through browsing or searching. The most popular website, offering an OID tree, was Harald Alvestrand's OID Registry [2]. In June 2003 the people of Elibel, France [5], merged the data maintained by Alvestrand into their repository, which now encompasses the one of Alvestrand. The OID Repository can be found here [6].

The presented online tools help getting an overview of existing and available schema elements. Each one has another focus and therefore has a right to exist. All in common, they don't specialize to a single area of application.

2.4. Other Previous Work

As part of the TERENA Definition of a European EduPerson (DEEP) project, a web-based questionnaire has been developed and announced to appropriate mailing lists . 18 institutions, 17 European and one Australian, took part and answered several questions about standardization of LDAP schema for the European academia. The most important results reasoned from the survey are [7]:

- Interoperability through common schema is seen as essential by 88% of the participants.

Another survey has been carried out by the MACE-Dir team in 2003. The "Local Domain Person Object Class Study" focused on gaining information about locally defined person object classes and attributes. Its goal was to find some patterns of deployment that can be recommended as best practices. 23 institutions reported about their reasons for creating local attributes. Further they gave information about their implementations of the `eduPerson` object class and the kind of attributes they have created. The report [3] contains a relevant list of attributes of the so called *localDomainPerson*, the local object class of an institute that contains additional attributes not included in the `eduPerson` specifications. The list is subdivided into eleven categories and provides a good profile of missing attributes in the de facto standard schemas.

In 2005, these two studies entered a document, that presents "A Comparative Analysis of Collaborative Public LDAP Person Object Classes in Higher-Education" [4]. A great number of contributing sources, nearly all from the European continent, lead to an even more extensive list of attributes than in [3]. The categories had been picked up and were refined to the following ones:

1. Personal Characteristics
2. Contact / Local Information
3. Student Information
4. Employee Information
5. Linkage Identifiers / Foreign Keys
6. Entry Metadata / Administration Information
7. Security Attributes and Keys
8. Confidentiality / Attribute Release (Visibility)
9. Authorization, Entitlements
10. Group-related Attributes
11. Other Miscellaneous Attributes

Although this research has been done concerning higher education, all attributes are applicable to normal organizations, except for those below "Student Information",

But even the last report does not provide enough information. In order to determine what schema elements can be reused and which have to be defined locally, an enterprise directory architect has to study each documentation of the listed schemas.

2.5. Related Work

There are already efforts to eliminate the lack of a comprehensive overview of existing identity schemas. The working group Identity Schemas, chartered by Identity Commons 2, recognized the entire problem, i.e. not restricted to LDAP directories. They try to solve the following drawbacks [18]:

1. No Catalogue.
2. No Common Description.
3. No Community.
4. No Reputation System.

They want to create a convenient catalogue of existing schemas for identity informations including their semantics. This is the basis for their vision of a world where identity agents can interact with a wide range of services. Unfortunately there is no work available yet, beside the "Poor man's registry", an incomplete list of existing schemas. In the sector of LDAP the most important schemas from RFCs and the eduPerson schema are mentioned.

Whereas their project has started to address the problem from a top-down perspective, the reference schema uses a bottom-up approach that focuses on the merging and mapping of one public schema to another. Useful results of their work are incalculable in time. Perhaps they can adopt conclusions from the reference schema. Their approach looks promising and availability of useful work may change in future.

3. Field Report

Gathering and consolidating information about existing LDAP schemas are two different steps in providing a basis for selecting predefined schema elements. Indeed the first one can imply some work, but there are enough sources and some tools available, that help to cope with this challenge as seen above.

Studying the available standard schemas from RFCs is a good starting point and using the presented tools makes this task relativly easy. After that, a good understanding of what is missing for the local LDAP directory should be available. The usage of internet search engines in order to find other publicly available schemas is recommended. Unfortunately the listed online tools above, provide no assistance locating actual published schemas. Section 2.2 documents the current state of affairs but without ongoing work it will soon be out of date. A running instance of a service like the Schema Registry, would be an immense benefit for enterprise directory architects.

But the more difficult challenge is the consolidation of gathered schemas and none of the available tools provide help for that step. First naive approaches attempted in project IDMone only worked for a limited number of schemas and winded up in the loss of overview. There is no problem in comparing single vs multiple values and different syntaxes and matching rules are subordinate. Of course, syntax and matching rules of a schema element have to fulfill the need of the local situation, but according to experience this is almost given. In the majority of cases this is decided earlier. The main problem is located in the semantics of the schema elements, which is given through the accompanying mostly informal documentation. In the same way additional constraints on attribute values are given only by further descriptions. Examples for this are the “controlled vocabulary” in eduPerson and the “Format” metadata item in SCHAC. The lecture and understanding of the accompanied documentation of public schemas was the most time-consuming part of work.

4. A Reference Schema

As there already is a conglomeration of published schemas, this paper is not meant to present yet another schema. There are two reasons for this decision. First, there are already enough standard or widely published schemas available, which will be extended by the corresponding communities. And second, there is no schema that fits all needs of a modern identity management system.

Based on practical experience of research group “MetaDirectory at Bavarian Universities” [16], each identity management system will need its own adapted schema. The schema does not only depend on the data that should be stored in a directory, as stated in section 1. But it also depends on the given identity management system and its application. According to this and the actual situation in an organisation, many different directory design strategies could make sense: From fat person objects, containing almost everything, to many small entries, distributing the individual information over the whole directory; from hierarchically organizational trees to flat people container. The only conclusion can be, that there is no unique schema for all circumstances.

The goal of the presented reference schema therefore is not the creation of one schema to rule them all. Instead it wants to help the enterprise directory architects to make their own decisions between the deployment of predefined schema elements and locally defined ones.

4.1. Method of Integration

The reference schema is attribute-based and object classes are not considered. The relevance of object classes in schema design should not be suppressed, but the integration of them poses no problem in most cases. A confirmation is given in [3, section 7], where the participants were asked about their localDomainPerson object class, the one locally defined person attributes are part of. The result was either an object class as structural with its superior being `inetOrgPerson`, an auxiliary object class or in one case both.

Beside object classes, matching rules are also not taken into account by the reference schema. Of course, matching rules are not only important in accessing entries in

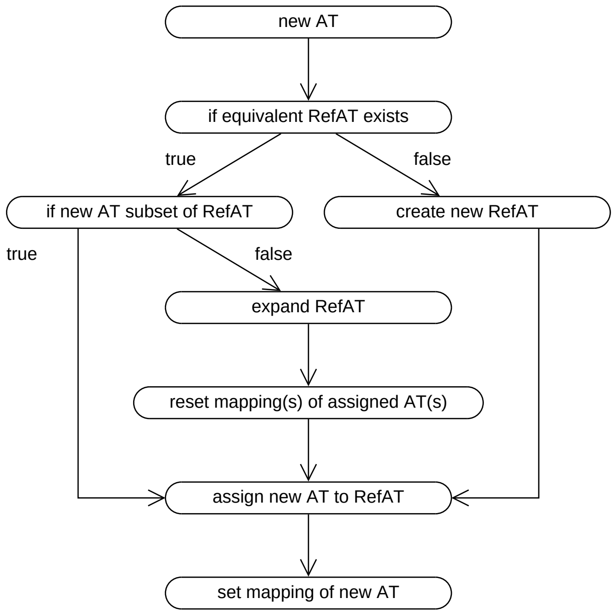

Figure 1: Method of Integration

the DIT, but they are subordinate in the decision process of different schema elements. If an adequate attribute is missing some needed matching rules, the deployment of an own attribute is necessary anyway.

Thus the reference schema emerges from integrating attribute types of existing schemas. Figure 1 represents the procedure, that is passed for all new attributes. At present every step is executed manually.

After recognizing the semantics of the new attribute, the reference schema had to be searched for an already existing equivalent reference attribute. If no one exists, a new reference attribute will be created and the new attribute will be assigned to it. If one exists, it has to be checked, whether the possible values of the new attribute are within the parameters of the reference attribute. If this is not the case, the domain of the reference attribute has to be expanded. After that, the mappings of already assigned attributes have to be checked and where necessary be reset. Finally, the new attribute will be assigned to the reference attribute and an appropriate mapping will be set.

The attribute types of the reference schema are classified according to the eleven categories from [4], listed above in 2.4. This division helps locating an attribute searched for, as in giving a first anticipation about the sense of an attribute, e.g. attributes in the category “Linkage Identifiers / Foreign Keys” are used for referencing other entries in the directory or in connected systems and therefore the attribute `Organization` will not contain a printable string of the organization of a person but a Distinguished Name (DN) of it.

The number of values a reference attribute can contain is given by the assigned attributes. If at least one attribute is multi-valued, the reference attribute is multi-valued too. Exceptions are attributes that are strictly single-valued by nature like `Date Of Birth`. Those attributes are single-valued, even if an attribute given by a schema exists, that is multi-valued. Mapping a multi-valued reference attribute to a single-valued attribute is done by returning an arbi-

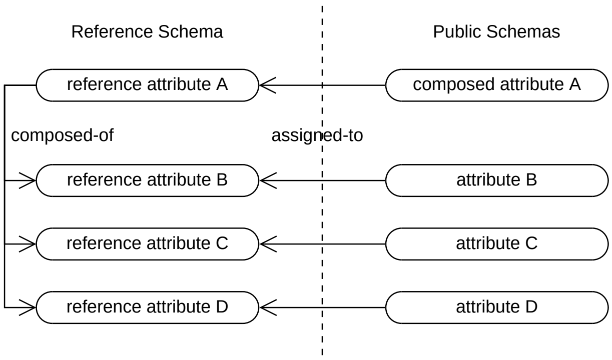

Figure 2: Example of Composed Attribute

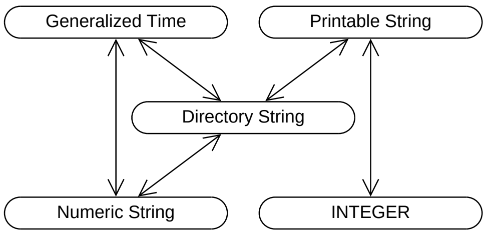

Figure 3: Mappings between Different Syntaxes

trary value, like some programming languages do (e.g. Perl's `get_value` in scalar context).

Most attributes represent domains whose elements are atomic, i.e. they are non-decomposable values from the application's point of view. But attributes may be decomposable in another or a global context. In order to cover all included schemas completely, the reference schema has to handle those composed attributes. Every attribute is assigned to exactly one reference attribute. In the case of a composed attribute, the according reference attribute has composed-of relations to other reference attributes, which represent part of its domain. Figure 2 shows the relationships of such a composed attribute. An example for a composed attribute is `postalAddress`.

4.2. Mapping

At present there are about 20 reference attributes, to which multiple attributes from different schemas are assigned. Mappings between those equivalent attributes enable a better comparison of them. As given through the method of integration, domains of reference attributes are always a superset of their public schema counterparts. Mappings towards the reference schema are hence always lossless. In the other direction, mappings with information loss cannot be avoided in all cases.

In order to map different attributes, their syntax has to be considered. The mostly used syntax of the included public schemas is `Directory String`. Followed by `DN`, `Printable String`, `Generalized Time`, and `Telephone Number` in this order. `DN` is only used to reference directory entries and is therefore never mapped to another syntax. Up to now only mappings between syntaxes shown in figure 3 have been occured. As a result of the demand, that "encoding rules for LDAP syntaxes should produce character strings that can be displayed with little or no translation by clients implementing LDAP" [17], values of attributes that use one of the displayed syntaxes in 3 can be treated as strings. The `Directory String` syntax is hence a superset of the other ones.

More limiting are the additional constraints given by the descriptions of the public schemas. Since there are no restrictions what kind of constraints are allowed, the reference schema cannot provide a complete list, but only the ones that occured. Basically two types can be differentiated:

- rule-based constraints (proprietary or RFCs)
- controlled vocabulary (enumeration types)

Examples for both types are given in the next section.

In consideration of the fact, that the reference schema is developed as part of the project IDMone, all mappings are described informal at present. But even a short textual description should be an assistance for directory architects.

4.3. Usage and Examples

Five of the more interesting reference attributes are picked up and are used to represent the different types of mappings. The examples show the benefits for an enterprise directory architects, as well as some restrictions of this appliance.

A link to the actual version of the reference schema can be found on the project's website [15]. The reference schema is "work in progress".

4.3.1 Date Of Birth

Three single-valued attributes are assigned to the reference attribute `Date Of Birth`. Although each one has another syntax, this has almost no effect on the mapping as shown shortly. The first one (`Directory String` syntax) restricts the values by the following "full-date" format, as described in RFC 3339 [13]: `YYYY-MM-DD`. The second has the same format but without the dashes, `YYYYMMDD`, and can therefore use the `Numeric String` syntax. The last one uses another approach and restricts its values to the shortest possible form of `Generalized Time`, namely: `YYYYMMDD00Z` (in which `00Z` is constant). The lossless mapping between these three attributes of different schemas should be quite obviously.

Table 1 represents a simplified extract of the reference schema. The directory architect can simply read off, that three public attributes for date of birth exist and that they are lossless mapped. Of course in this case, the ignored matching rules hide the different search possibilities of the attributes.

4.3.2 Gender

An example, where lossless mapping is not always possible, is given by `Gender`. Two schemas provide an attribute for it. The first one uses `Printable String` syntax and restricts its values to the following set: `M` and `m` (male), `F` and `f` (female). The second uses `INTEGER`

single-valued	composed	loss	Reference Schema	WALAP	Codex	HIS	eduPerson	RFC 4519	SCHAC
s			Date Of Birth		X	X			X
s		loss	Gender			(X)			X
		loss	Expiry Date	(X)	(X)				X
		loss	Affiliation	(X)	(X)		(X)		
	c		Official Postal Address					X	

Table 1: Simplified Extract of Reference Schema

as syntax and references ISO 5218 for its values: `0` (not known), `1` (male), `2` (female), and `3` (not specified). Attributes, which are able to keep only a subset of the domain are marked with parenthesis around the `X`. It is recognizable that not all attributes have the same amount of values.

4.3.3 Expiry Date

`Expiry Date` is another example of a mapping with loss. Two of the three attributes restrict the format to `YYYY-MM-DD` using an appropriate syntax. The last attribute uses `Generalized Time` syntax. While the first two have only a resolution of days, the last one is exact to the second.

4.3.4 Affiliation

The relationship(s) of a person to its institution in broad categories is represented by the reference attribute `Affiliation`. Three schemas define an equivalent attribute and all use a controlled vocabulary:

1. `student`, `staff`, `others`
2. `Mitarbeiter`, `Student`, `Bibliotheksbenutzer`, `Gast`, `Alumni`
3. `faculty`, `student`, `staff`, `alum`, `member`, `affiliate`, `employee`

Although some words can be mapped, no attribute is a superset of all others. Thus mapping from the reference attribute to any attribute is with loss.

4.3.5 Postal Address

The last example shows a composed attribute. `postalAddress` is assigned to `Official Postal Address`, which is composed of several other reference attributes. The table containing the composed-of relationships has been omitted on behalf of overview.

5. Summary and Conclusions

There is no identity management system like the other. They differ in as much detail as they share common identity-related attributes. The interoperability between organizations depends on either using the same syntax and semantics for common attributes or the availaility of mappings between them. Schema design is a key in enabling information interchange. The introduced reference schema supports the reuse of common schema elements and provides mappings for locally adapted ones. At present it provides assistance for directory architects in the design phase which was a demand of the IDMone project.

Further work on formal mapping of equivalent attributes would make sense. Most attributes have a strict semantic given by their documentation. Their values could be translated automatically on the fly by an appropriate tool. In this way a uniform view of data held in different identity management systems could be established.

References

[1] Ed. A. Sciberras. LDAP: Schema for User Applications. RFC 4519, IETF, Jun 2006.

[2] Harald Alvestrand. Object Identifiers Registry [online]. Available: http://www.alvestrand.no/objectid/, Feb 1997. [last visited: 17.08.07].

[3] Brendan Bellina. Local Domain Person Object Class Study. Technical report, NSF Middleware Initiative, Apr 2004.

[4] Brendan Bellina and Peter Gietz. Higher-Education Person: A Comparative Analysis of Collaborative Public LDAP Person Object Classes in Higher-Education. draft, NSF Middleware Initiative, Nov 2005.

[5] Elibel. ASN.1 Information Site [online]. Available: http://asn1.elibel.tm.fr, Aug 2007. [last visited: 17.08.07].

[6] Elibel. OID Repository - Home [online]. Available: http://www.oid-info.com, Aug 2007. [last visited: 17.08.07].

[7] Peter Gietz. Report on the results of the DEEP questionnaire. Technical report, TERENA, Dec 2002.

[8] Peter Gietz. Standardbasierte LDAP-Schemata für Personen- und Organisationsdaten und Spezifikation des HIS-LDAP-Schemas. Technical report, HIS, Jul 2005. Version 0.5, 4.7.2005.

[9] Timothy A. Howes, Mark C. Smith, and Gordon S. Good. *Understanding and Deploying LDAP Directory Services*. Addison Wesley Professional, second edition, 2003.

[10] DAASI International. schemareg.org. Available: http://www.schemareg.org, 2003. [last visited: 17.08.07].

[11] Internet2. MACE-Dir [online]. Available: http://middleware.internet2.edu/dir/. [last visited: 17.08.07].

[12] Internet2. Internet2 Middleware [online]. Available: http://www.nmi-edit.org/eduPerson/internet2-mace-dir-eduperson-200604.html, Apr 2006. [last visited: 17.08.07].

[13] G. Klyne. Date and Time on the Internet: Timestamps. RFC 3339, IETF, Jul 2002.

[14] Alan Knowles. AKBK home: [online]. Available: http://ldap.akbkhome.com. [last visited: 17.08.07].

[15] RRZE. RRZE - IDMone [online]. Available: http://www.rrze.uni-erlangen.de/forschung/laufende-projekte/idm_e.shtml, Jul 2007. [last visited: 17.08.07].

[16] RZUW. Rechenzentrum: Arbeitskreise [online]. Available: http://www.rz.uni-wuerzburg.de/infos/arbeitskreise/, Jul 2007. [last visited: 17.08.07].

[17] Ed. S. Legg. LDAP: Syntaxes and Matching Rules. RFC 4517, IETF, Jun 2006.

[18] Identity Schemas. FrontPage - Identity Schemas Wiki [online]. Available: http://identityschemas.org/, Feb 2007. [last visited: 17.08.07].

[19] M. Smith. Definition of the inetOrgPerson LDAP Object Class. RFC 2798, IETF, Apr 2000.

[20] TF-EMC2. TERENA Activities Tf-emc2 Download SCHAC Releases [online]. Available: http://www.terena.org/activities/tf-emc2/schacreleases.html, Jan 2007. [last visited: 17.08.07].

[21] TF-EMC2. TERENA Activities Tf-emc2 [online]. Available: http://www.terena.org/activities/tf-emc2/, May 2007. [last visited: 17.08.07].

[22] M. Wahl. A Summary of the X.500(96) User Schema for use with LDAPv3. RFC 2256, IETF, Dec 1997.

[23] WALAP. WALAP - WA Libraries Authentication Project [online]. Available: http://walap.curtin.edu.au/docs/, Aug 2003. [last visited: 17.08.07].

[24] K. Zeilenga. LDAP: Directory Information Models. RFC 4512, IETF, Jun 2006.