
CodeX: LDAP-Schema för LADOK

Leif Johansson, Stockholms universitet <leifj@it.su.se>

Detta dokument beskriver ett LDAP-schema och en URN-struktur för LADOK-information. Syftet är att konstruera ett schema som kan användas för att modellera en delmängd av LADOK informationsmodell tillräcklig för representation av information om antagna och registrerade studenter, lärare, examinatorer och andra liknande relationer i en högskolas enterprisekatalog.

Table of Contents

Vem bör läsa detta dokument	1
Relation med LpW	1
Design	1
En URN-specifikation för LADOK	2
Objektmodell	3
LADOK-Schema v1.0	4
Exempel	7
Användning av schemat	8
Referenser	8

Vem bör läsa detta dokument

Detta dokument riktar sig till mjukvaruutvecklare och katalogadministratörer vid svenska högskolor och universitet. Dokumentet är tekniskt och förutsätter kunskap om LDAP och schemadesign. Förtrogenhet med objektmodellering underlättar förståelsen av schemats design. Läsaren förutsätts också vara insatt i hur LADOK är strukturerad.

Relation med LpW

LpW - LADOK på Webb utgör ett alternativt sätt att komma åt information i LADOK. Många tillämpningar ställer emellertid krav på prestanda som LpW inte kan uppfylla. Tillgång till LADOK-data i en LDAP-katalog är ofta en naturlig lösning på prestandaproblemet för sådana applikationer.

Design

LADOK har en komplex informationsstruktur som inte är uttryckt i OO-termer. Detta faktum och andra egenskaper hos LADOK medför att en fullständig representation av LADOK som ett LDAP-schema varken är möjlig eller speciellt intressant. Detta dokument är resultatet av arbete inom CodeX arbetsgrupp för LDAP inom Svenska Universitet och Högskolor. Denna arbetsgrupp har fokuserat på användningen av LADOK-data inom IT-drift (kontoadministration, behörighetskontroll mm).

Designmålen för schemat har varit:

1. Representera grupperingar av personer (tex alla antagna studenter till en viss kurs) som auxiliära objektklasser associerade med objekt av typen `groupOfUniqueNames` eller `groupOfNames`. Syftet med detta mål är att LADOK-relaterade grupper ska kunna användas av andra applikationer (tex mailinglistor) som inte behöver känna till gruppernas association med LADOK.
2. Schemat ska så långt som möjligt inte förutsätta någon speciell trädstruktur i katalogen. Detta

medför att vissa referenser mellan objekt i schemat måste göras explicit mha attribut istället för implicit baserat på katalogstrukturen.

3. Schemat ska medge att vanliga sökningar och uppdateringar kan ske effektivt.
4. Schemat skall uppfylla kraven som ställs på enterpriskatalogen av VHS NyA.
5. Schemat skall inte kräva syntaxer, namn- eller strukturregler eller matchningsregler som inte är implementerade i marknadsledande LDAP-serverar.
6. Schemat skall uppfylla kraven i [codex-directory-bcp].

En URN-specifikation för LADOK

Referenser mellan objekt i LDAP kan antingen ske med DN (distinguished name) eller med annan data som via en sökning i katalogen unikt bestämmer ett entry. Referenser som använder DN förutsätter att katalogstrukturen är relativt stabil och inte (tex) påverkas av personers organisationstillhörighet. En sådan stabilitet kan vara svår att uppnå och de flesta katalogserverar implementerar inte referensintegritet för DN ens inom samma server. Därför är det ofta bra att skapa korsreferenser på annat sätt. Ett sätt är att skapa ett enhetligt namnrum som är oberoende av katalogens namnstruktur.

Vid vissa högskolor har man sedan en tid använt ett namespace för att referera till LADOK-kurser där kurskoden ingår som komponent i namnet. Genom att konstruera ABNF [RFC2234] för en sådan namnstruktur gör man det enkelt att göra sökningar i katalogen efter kurser samtidigt som man kringgår begränsningar till följd av att attributvärden i LDAP är strängar (se design-mål ovan angående strukturerad data). Ett URN [RFC2141] är ett sätt att formalisera denna ide samtidigt som man på ett unikt sätt kan referera till samtliga i schemat representerade informationsobjekt (kurser, prov, kurstillfällen etc etc). En URN-specifikation blir dessutom utbyggbar på ett naturligt sätt.

Detta är ABNF för URN-specifikationen för LADOK-schemat. Vi har valt att inte använda svenska ord som tokens för att undvika framtida problem med utf8 i protokollparametrar.

```
ladok-urn          = "urn:ladok:" ladok-urn-nss
ladok-urn-nss      = "info:" school-id ":" info-spec
info-spec          = ( course / course-instance / test )
school-id          = 1( ALPHA / DIGIT )
course             = "c" "-" code
course-instance    = "ci" "-" semester "-" code 0*1( ";" uniquepart )
uniquepart         = 1( ALPHA / DIGIT )
test               = "t" "-" code
code               = 1( ALPHA / DIGIT )
semester           = 4*4( DIGIT ) ("1" / "2")
```

För att göra det möjligt att representera andra entiteter än rena informationsobjekt införs prefixet info: för det URN-träd som innehåller referenser till kurser, prov och kurstillfällen.

Den av LADOK-konsortiet tilldelade databasidentiteten (ex US för Stockholms universitet) skall användas som school-id.

Här avses den unika koden som identifierar objektet i motsvarande LADOK-tabell.

Terminer anges i LADOK-format dvs året (4 siffror) följt av en terminssiffra som är antingen 1 eller 2 (VT respt HT).

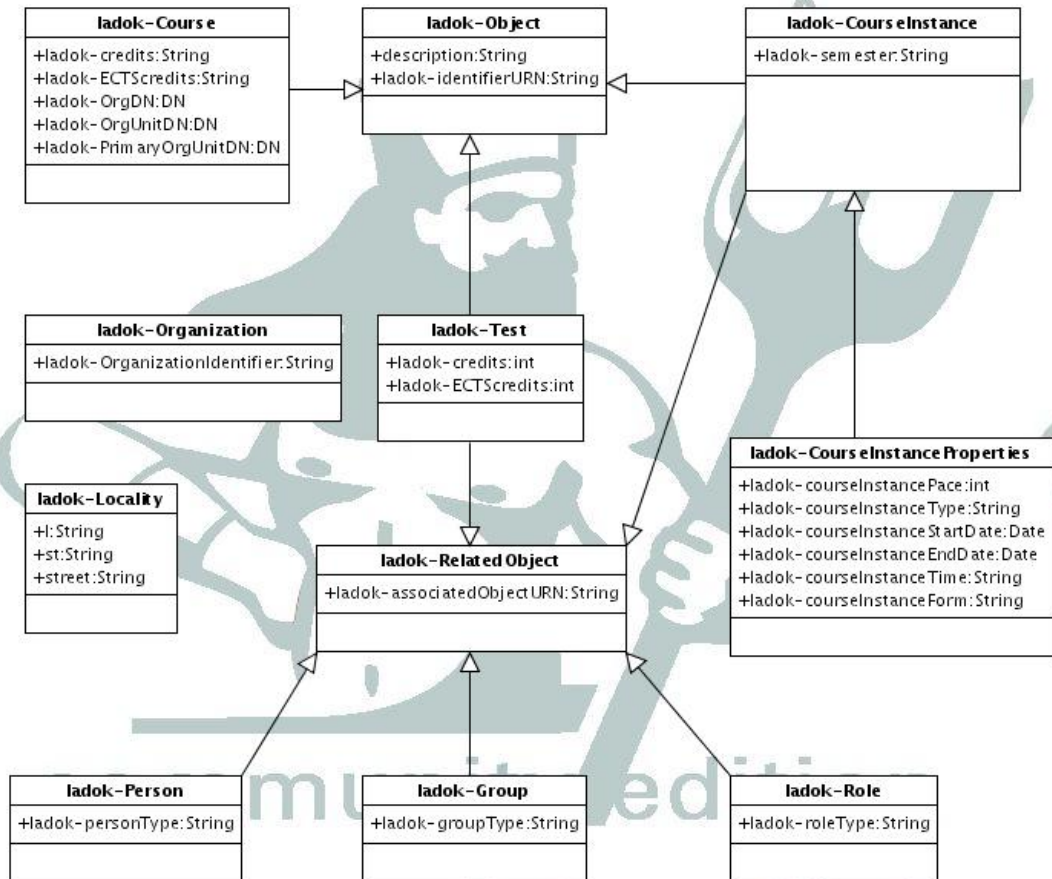
Multipla kursinstanser kan förekomma på samma termin. I LADOK använder man diverse tabeller (tex **ETYPBEN**) för att skilja dem åt. produktionen **uniquepart** kan konstrueras mha denna tabell.

Exempel

En URN som refererar till kursen AI1020: 'Introduktionskurs i Astronomi' vid Stockholms universitet: **urn:ladok:info:US:c-AI1020**.

Objektmodell

Även om ett schema inte är en OO-modell är det ändå värdefullt att se en UML-modell av ett schema speciellt om schemat är någorlunda komplext. Detta är den UML-modell som ligger till grund för LADOK-schemat:



Det finns i huvudsak två typer av objektclasser i schemat: dels klasser som (i OO-termer) ärver från `ladok-Object` och dels klasser som ärver från `ladok-RelatedObject`. Objektclasserna `ladok-Course`, `ladok-Test` och `ladok-CourseInstance` är strukturella klasser. De två sistnämnda hör även till den andra typen av objektclasser eftersom det är associerade med kurser. Klasserna `ladok-CourseInstanceProperties` och `ladok-Locality` är bärare av metadata om kurstillfällen som inte måste implementeras av alla. Syftet med att separera dessa attribut (tex kurstid, kursform etc) är att göra uppdelningen tydligare samtidigt som applikationer enkelt kan avgöra om metadata finns representerat i katalogen.

Den andra typen av objektclasser är auxiliära objekt som samtliga är associerade med någon av de strukturella objekten. Typen av association (grupp, person, roll, etc) är bestämt av objektclassen. Detta faktum medför att sökningar av typen "alla studenter registrerade på en kurs" eller kommer att använda filter av typen **objektClass=ladok-Course** i kombination med andra filter. Det är förhållandevis enkelt att optimera en katalogserver så att sådana sökningar blir effektiva.

Objekt som är associerade med `ladok-objekt` (auxiliära objektclasser) saknar alltså egen unik identifierare men har alla ett attribut `ladok-associatedObjectURN` som innehåller URN för de LADOK-objekt (kurser, prov etc) som associationen avser. Detta attribut är flervärdigt men samtliga associerade objekt måste vara av samma typ.

Namnregler och strukturregler kan inte specificeras i LDAP-schema och detta schema innehåller inga formella sådana regler. Emellertid skall följande krav uppfyllas av implementationer av detta schema:

1. Strukturella objekt skall namnges med cn-attributet. Namnet kan innehålla samma värde som värdet av attributet ladok-identifierURN men andra namngivningssystem för RDN kan förekomma. Applikationer ska bara använda attributet ladok-identifierURN.
2. Entryn av objektclass ladok-Group skall vara associerade med strukturella entryn av typ groupOfNames eller groupOfUniqueNames. Klienter skall välja värdet av attributet member respektive uniqueMember som listan på personer som har den av ladok-groupType givna associationen med de strukturella objekten som är associerade med gruppen. I enlighet med hantering av grupper i [codex-directory-bcp] är värdet av uniqueMember alltid ett DN (dvs ej på formen # uid).

LADOK-Schema v1.0

Schemat följer syntaxen i [RFC2252].

```
## Objektidentifierare

# se-id          OBJECT IDENTIFIER ::= { iso(1) member-body(2) se(752) }
# su-id          OBJECT IDENTIFIER ::= { se-id (su.se)43 }
# ladok-id       OBJECT IDENTIFIER ::= { su-id 8 }
# ladok-oc-id    OBJECT IDENTIFIER ::= { ladok-id oc(1) }
# ladok-a-id     OBJECT IDENTIFIER ::= { ladok-id a(2) }

## Attributtyper

attributetype ( 1.2.752.43.8.2.1
  NAME 'ladok-identifierURN'
  DESC 'A ladok-URN'
  EQUALITY caseIgnoreIA5Match
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 )

attributetype ( 1.2.752.43.8.2.2
  NAME 'ladok-credits'
  DESC 'Course or test-credits multiplied by 10'
  EQUALITY integerMatch
  SINGLE-VALUE
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 )

attributetype ( 1.2.752.43.8.2.3
  NAME 'ladok-ECTScredits'
  DESC 'Course or test-credits according to ECTS multiplied by 10.'
  EQUALITY integerMatch
  SINGLE-VALUE
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 )

attributetype ( 1.2.752.43.8.2.4
  NAME 'ladok-associatedObjectURN'
  DESC 'The URN of the associated LADOK object'
  EQUALITY caseIgnoreIA5Match
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 )

attributetype ( 1.2.752.43.8.2.5
  NAME 'ladok-semester'
  DESC 'Semester according to LADOK standard format'
  EQUALITY integerMatch
  SINGLE-VALUE
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 )

attributetype ( 1.2.752.43.8.2.6
  NAME 'ladok-courseInstancePace'
  DESC 'Course pace in %'
  EQUALITY integerMatch
  SINGLE-VALUE
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 )
```

```
attributetype ( 1.2.752.43.8.2.7
  NAME 'ladok-courseInstanceForm'
  DESC 'Undervisningsform (distans etc)'
  EQUALITY caseIgnoreIA5Match
  SINGLE-VALUE
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 )

attributetype ( 1.2.752.43.8.2.8
  NAME 'ladok-courseInstanceTime'
  DESC 'Undervisningstid (dag,kväll)'
  EQUALITY caseIgnoreIA5Match
  SINGLE-VALUE
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 )

attributetype ( 1.2.752.43.8.2.9
  NAME 'ladok-courseInstanceStartDate'
  DESC 'The date (week) the course starts in.'
  EQUALITY generalizedTimeMatch
  ORDERING generalizedTimeOrderingMatch
  SINGLE-VALUE
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.24 )

attributetype ( 1.2.752.43.8.2.10
  NAME 'ladok-courseInstanceEndDate'
  DESC 'The date (week) the course ends by.'
  EQUALITY generalizedTimeMatch
  ORDERING generalizedTimeOrderingMatch
  SINGLE-VALUE
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.24 )

attributetype ( 1.2.752.43.8.2.11
  NAME 'ladok-courseInstanceType'
  DESC 'Kurstyp (ETYPBEN)'
  EQUALITY caseIgnoreIA5Match
  SINGLE-VALUE
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 )

attributetype ( 1.2.752.43.8.2.12
  NAME 'ladok-roleType'
  EQUALITY caseIgnoreIA5Match
  SINGLE-VALUE
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 )

attributetype ( 1.2.752.43.8.2.13
  NAME 'ladok-groupType'
  EQUALITY caseIgnoreIA5Match
  SINGLE-VALUE
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 )

attributetype ( 1.2.752.43.8.2.14
  NAME 'ladok-personType'
  EQUALITY caseIgnoreIA5Match
  SINGLE-VALUE
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 )

attributetype ( 1.2.752.43.8.2.15
  NAME 'ladok-OrgIdentifier'
  DESC 'The department identifier from LADOK'
  EQUALITY integerMatch
  SINGLE-VALUE
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 )

attributeType ( 1.2.752.43.8.2.16
  NAME 'ladok-OrgDN'
  DESC 'The organizational reference (DN) of an object'
  EQUALITY distinguishedNameMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
  SINGLE-VALUE )
```

```
attributeType ( 1.2.752.43.8.2.17
    NAME 'ladok-OrgUnitDN'
    DESC 'The organizationalUnit reference (DN) of an object'
    EQUALITY distinguishedNameMatch
    SYNTAX 1.3.6.1.4.1.1466.115.121.1.12 )

attributeType ( 1.2.752.43.8.2.18
    NAME 'ladok-PrimaryOrgUnitDN'
    DESC 'The primary organizationalUnit reference (DN) of an object'
    EQUALITY distinguishedNameMatch
    SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
    SINGLE-VALUE )

attributeType ( 1.2.752.43.8.2.19
    NAME 'ladok-identifierURN'
    EQUALITY caseIgnoreIA5Match
    SINGLE-VALUE
    SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 )

## Objektklasser

objectclass ( 1.2.752.43.8.1.1
    NAME 'ladok-StructuralObject'
    DESC 'The base class of all LADOK structural objects'
    SUP top
    STRUCTURAL
    MUST ( cn ) )

objectclass ( 1.2.752.43.8.1.2
    NAME 'ladok-Object'
    DESC 'The base class of all LADOK auxiliary objects'
    AUXILIARY
    MAY ( description )
    MUST ( ladok-identifierURN ) )

objectclass ( 1.2.752.43.8.1.3
    NAME 'ladok-Course'
    DESC 'A course (KURS)'
    SUP ladok-Object
    AUXILIARY
    MUST ( ladok-credits $ ladok-ECTScredits )
    MAY ( ladok-OrgDN $ ladok-OrgUnitDN $ ladok-PrimaryOrgUnitDN ) )

objectclass ( 1.2.752.43.8.1.4
    NAME 'ladok-Test'
    DESC 'A test (PROV2)'
    SUP ladok-Object
    AUXILIARY
    MUST ( ladok-credits $ ladok-ECTScredits ) )

objectclass ( 1.2.752.43.8.1.5
    NAME 'ladok-CourseInstance'
    DESC 'A course-instance (KTILL2)'
    SUP ladok-Object
    AUXILIARY
    MUST ( ladok-semester ) )

objectclass ( 1.2.752.43.8.1.6
    NAME 'ladok-CourseInstanceProperties'
    DESC 'Extra data about a course-instance.'
    SUP ladok-Object
    AUXILIARY
    MAY ( ladok-courseInstanceStartDate $
        ladok-courseInstanceEndDate $
        ladok-courseInstancePace $
        ladok-courseInstanceType $
        ladok-courseInstanceTime $
        ladok-courseInstanceForm ) )
```

```
objectclass ( 1.2.752.43.8.1.7
  NAME 'ladok-RelatedObject'
  AUXILIARY
  MUST ( ladok-associatedObjectURN ) )

objectclass ( 1.2.752.43.8.1.8
  NAME 'ladok-Person'
  AUXILIARY
  SUP 'ladok-RelatedObject'
  MUST ( ladok-personType ) )

objectclass ( 1.2.752.43.8.1.9
  NAME 'ladok-Group'
  AUXILIARY
  SUP 'ladok-RelatedObject'
  MUST ( ladok-groupType ) )

objectclass ( 1.2.752.43.8.1.10
  NAME 'ladok-Role'
  AUXILIARY
  SUP 'ladok-RelatedObject'
  MUST ( ladok-roleType ) )

objectclass ( 1.2.752.43.8.1.11
  NAME 'ladok-Locality'
  AUXILIARY
  MAY ( 1 $ st $ street ) )

objectclass ( 1.2.752.43.8.1.12
  NAME 'ladok-Organization'
  AUXILIARY
  MUST ( ladok-OrgIdentifier ) )
```

Exempel

Kursen AI1020: 'Introduktionskurs i Astronomi' vid Stockholms universitet. Observera att värdet av dn, ladok-orgDN och ladok-orgDN är beroende av hur katalogen är strukturerad och inte är specificerat av schemat. Exemplet utelämnar alla operationella attribut och använder den experimentella objektclassen namedObject [NamedObject] som strukturell objektclass istället för ladok-StructuralObject. I exemplet används en namngivning av entryt som inte är relaterat till ladok-identifierURN för att illustrera att namngivningen är helt oberoende av den globalt unika identifieringen av objektet som ges av objektets URN.

```
dn: cn=Grundkurs,ou=Astronomi,ou=Kurser,dc=su,dc=se
cn: Grundkurs
ladok-identifierURN: urn:ladok:info:US:c-AI1020
ladok-credits: 50
ladok-ECTScredits: 75
description: Introduktionskurs i astronomi
description/lang-en: Introductory course in Astronomy
ladok-orgUnitDN: ou=Astronomi,o=Stockholms universitet,c=SE
ladok-orgDn: o=Stockholms universitet,c=SE
objectclass: ladok-Course
objectClass: ladok-Object
objectClass: namedObject
objectClass: top
```

Att göra

1. Fler exempel.

Användning av schemat

Det beskrivna schemat kan användas på många sätt. Detta avsnitt specificerar vissa sökningar som en klient kan förvänta sig fungera och vara så hastighets-optimerade som möjligt.

Referenser

[NamedObject] *Named Object LDAP ObjectClass*. ???.

[codex-directory-bcp] *CodeX: Rekommendationer för drift och strukturering av katalogsystem*. 2003. Leif Johansson.

[RFC2141] *URN Syntax*. May 1997. R. Moats.

[RFC2234] *Augmented BNF for Syntax Specifications: ABNF*. November 1997. D. Crocker and P. Overell.

[RFC2252] *Lightweight Directory Access Protocol (v3): Attribute Syntax Definitions*. December 1997. M. Wahl. A. Coulbeck. T. Howes. S. Kille.