METADIRECTORY TOOL
UDS/MD

Roland Hedberg
<roland.hedberg@adm.umu.se>
PROBLEMS WE HAVE TRIED TO ADDRESS

- **A way to uniquely identify objects within UDS/MD;** how do you match information stored about one object in one source with information about the same object at another source, when each source may have its own unique identifier for the object?

- **Means by which information of the same kind can be addressed in a common way;** a target should not have to know under what name each source has a particular piece of information stored, and there must be one, and only one, attribute name for each piece of information that UDS/MD shall handle.

- **Harmonization of formats;** a target should not have to know all the different representations that might exist of the same type of information, therefore there must be a common format that local representations can be converted to and from.

- **Well defined input/output formats;** information changes, that occurs in sources, that should be reflected as changes in the targets has to be represented in a common, well defined format.
MORE PROBLEMS

- **Logic**: like when a set of attribute/values can be combined to form a new set of attribute/values
- **Dynamic data flow**: which information that should flow where should be changeable on the spur of the moment
- **Open data structure**: the system should be able to handle new types of information without much ado
- **Changeable environment**: sources and sinks should be possible to add and remove while the system is running
- **Replaying**: it should be easy to ‘replay’ a set of events
- **Logging**: should be possible to trace a packets way from a source to a sink.
- **Adapters**: facilitate the building of new adapters
FEATURES OF UDS/MD@UMU

- Events as RDF graphs
- Interface between UDS/MD and sources/sinks
  - DjB MailDir
- Makes a real effort not to loose packets
- ‘Pattern’ based routing
  - content ‘filtering’
- Object Identifier mapping
<uds:add> <uds:src> "primula" .
<uds:add> <uds:eid> "6648" .
<uds:add> <uds:person> _:a .
  _:a <uds:oid> "NIN:19570331-8575" .
  _:a <uds:guise> "anst1" .
  _:a <udsattr:norEduPersonNIN> "19570331-8575" .
  _:a <udsattr:givenName> "Per Arne" .
  _:a <udsattr:sn> "Nilsson" .
  _:a <udsattr:umuEduPersonEmploymentExtent> "100" .
  _:a <udsattr:norEduOrgUnitUniqueNumber> "1500" .
  _:a <udsattr:eduPersonAffiliation> "faculty" .
A. Spocp contains a policy engine that is very good at matching S-expressions against ‘rules’

B. RDF graphs can loss-less be transformed to and from S-expression

A + B => route server

Spocp can bind return information to rule
RETURN INFO

/ldap/((<uds:person>(* set (<uds:add>)(<uds:eid>)(<uds:src> \22kdb\22))))

== "http://127.0.0.1:1115/RPC2;uds:oid, udsattr:nuEduPersonNIN, udsattr:uid, udsattr:givenName, udsattr:sn"
OBJECT IDENTIFIER MAPPING

- One persistent ID, never reused or changed
- One or more ‘local’ IDs which may or may not change

11565371 (UDS/MD@UmU ID)
  - uid: peni0001
  - nin: 195703318575
  - dn: uid=peni0001,ou=person,dc=umu,dc=se
Work of the adapter

- Character set conversion
- Import from source or export to sink
- Attribute mapping
- Attribute value harmonization
- Safe delivery
- RDF construction/deconstruction
- Logic
LDAP Adapter

- Uses attribute options to keep guises separate
- add “ascii” versions
- handle ‘tokenization’
- Constructs attribute values from other attribute values
- Collect attribute values from other entries
displayName: Nilsson, Per Arne
cn: Per Arne Nilsson
cn: Per Nilsson
cn: Arne Nilsson
givenName: Per Arne
givenName: Per
givenName: Arne
sn: Nilsson
<uds:add> <uds:src> "primula" .
<uds:add> <uds:eid> "6648" .
<uds:add> <uds:person> _:a .
_:a <uds:guise> "anst1" .
_:a <udsattr:umuEduPersonEmploymentExtent> "100" .
_:a <udsattr:norEduOrgUnitUniqueNumber> "1500" .
_:a <udsattr:eduPersonAffiliation> "faculty" .
dn: uid=peni0001, cn=person, dc=umu, dc=se
displayName: Nilsson, Per Arne
cn: Per Arne Nilsson
cn: Per Nilsson
cn: Arne Nilsson
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
objectClass: eduPerson
objectClass: norEduPerson
objectClass: umuEduPerson
o;x-search: Umea Universitet
eduPersonOrgDN: dc=umu, dc=se
eduPersonPrincipalName: peni0001@umu.se
o:: VW1lw4UgVW5pdmVyc2l0ZXQ=
o;lang-en: Umea University
uid: peni0001
givenName: Per Arne
givenName: Per
givenName: Arne
norEduPersonNIN: 19570331-8575
sn: Nilsson
eduPersonOrgUnitDN; x-guise-anst1: umuSeOrgUnitID=1500, cn=org, dc=umu, dc=se
eduPersonOrgUnitDN: umuSeOrgUnitID=1500, cn=org, dc=umu, dc=se

umuEduPersonEmploymentExtent; x-guise-anst1: 100
umuEduPersonEmploymentExtent: 100
eduPersonAffiliation; x-guise-anst1: faculty
eduPersonAffiliation; x-guise-anst1: member
eduPersonAffiliation; x-guise-anst1: employee
eduPersonAffiliation: faculty
eduPersonAffiliation: member
eduPersonAffiliation: employee
ou; x-guise-anst1: Arkeologi och samiska studier
ou: Arkeologi och samiska studier
First stage is in operation
- Person information (employees, students)

Second stage just started
- More objects: roles, groups, courses, ...
- Guests
- More sources, sinks
- Admin GUI in operation