

URN registration and delegation system

December 9, 2007

1 Purpose

The use of URNs as values for directory attributes is undergoing a real explosion in numbers. Tracking the meaning of such long string values, as well as establishing a common vocabulary will soon become a daunting task.

Up to now, URN registries are simple static web pages maintained by hard working professionals, with the exponential growth of values, the time needed for this task will greatly increase, moreover, static html pages are not useful for machine interpretation, just for human use.

Delegation of namespaces is merely done by registering the URLs for the pages where the persons responsible for the different namespaces keep their values, this, also, is not very useful for machine interpretation.

If URNs values are stored in such way that both human and machine readable formats can easily be produced, the number of possibilities for presenting and using the information will dramatically increase.

Also, this will, for sure, be easier to maintain and to keep up to date.

2 Implementation

In theory, the back end storage for URN value information is not relevant.

In practice, we will implement an LDAP directory based solution, that, will also provide a standardized way for querying the repository directly, if the need arises and the administrators of the repository so decide.

Repositories will be distributed, creating a new one for each namespace delegation, in a way much like DNS.

2.1 Storage

We will implement the URN registry as LDAP directory objects, but this does not preclude other storage technologies if the necessary query interfaces are provided. Not

everybody is willing to expose their directories to the wild open Internet, though this do not have to be at all necessary.

Access to storage may be hidden behind a layer of web services.

2.2 Namespace delegation

The namespace delegation occurs any time a part of a URN value is assigned to a certain institution.

In order to find the registry that is responsible for a part of the namespace, we propose three methods:

1. DDDS for using the URN values as a means of finding the registry responsible for them. This should be implemented for sure to find the root namespace.
2. URLs for human interface location, in order for a web application to follow those URLs to take the users to the portal of each URN registry.
3. URLs for machine interface. This will be used by applications for issuing direct queries to the registries for presenting the information to the user inside the application, for example, in their native language. These URLs could point both to HTTP interfaces for web services and LDAP interfaces for direct queries, to the registry that owns the namespace.

2.3 Human interface

The human interface is, of course, a web application, that queries the registry that owns the namespace.

Once a namespace delegation entry is reached, there are two possible paths to follow:

1. In order not to confuse the user, the application queries the remote registry and presents the data to the user with the local look and feel. This is probably a less confusing experience for the user, that even allows to present the user with a localized version of the value, if it is available.
2. The application presents the registry human interface URL to the user. Following such link the user will arrive to the registry entry page. This has the risk of landing the user on a page she does not understand, but clearly presents the identity of the namespace owner. This way of operation should follow normal i18n procedures for selecting the user language and, if it is unavailable either on the request or inside the registry, it must present information in English.

2.4 Machine interface

As stated above, we will implement an LDAP based storage back end, but this is not an absolute requirement. Then, we propose two methods for machine access to the repositories:

1. Standard LDAP query from the human interface. This will probably be the method used for local queries. If the administrators of a registry decide to open directory access, such access could be protected in any way agreed upon by the participating registries.
2. Web services functions for getting both URN delegation and URN description objects. The machine interface URI of participating registries must provide a web service with two functions: `getURNdelegation` and `getURNdescription`. These functions return the solicited information if the visibility permissions allow. We will use the SiLeDAP Ajax/PHP directory management framework to implement this interface, as well as the human one.

3 Operations

Here, we describe technical operations. The policy level operations should be described in a policy document.

3.1 Search

NOTE: search operations at delegation level should provide a means of mapping URN values to the corresponding delegation objects.

3.2 List

Either delegations or descriptions can be listed as a list of search operation URLs for each URN value.

3.3 Registration

There should be a local means of entering URN information in the local repository. These operations are outside the scope of the present document.

4 Protocol

Communications will take place over REST using URLs for locating a given URN data.

Search URL: urnRegHttps/search?urn=UNRValue&format=FORMAT

List URL: urnRegHttps/list?type=objectType&format=FORMAT

format being human or machine, default being human, if not present

type being delegation or description, description being default, if not present

If there is no operation, the entry point to the registry should be presentend in human format.

If the URN parameter is mmissing in search operations, a search from should be presented.

5 Definitions

5.1 Attributes

urnRegOwner Internet domain of the institution that owns the URN namespace.

urnRegHttpsURI HTTPS URL pointing to the entry point for webservises querying of the URN registry, for a given namespace. It can produce both human an machine responses, human being the default.

urnRegLdapURI LDAP URL pointing to the entry point for issuing queries to the URN registry, for a given namespace. With strong suggestion of using either TLS or LDAPS.

urnRegPolicyURI HTTP URL pointing to a delegation policy document that informs a human about how delegations are made under a given namespace.

urnRegRegistrarURI HTTP URL pointing to the namespace registrar home page.

urnRegValue registered URN value.

urnRegValueDescription description of the value, for humans. This attribute allows for multilingual representatins of the value description according to RFC 3866.

urnRegValueVisibility informs about the level of visibility of the URN. Initially, we have considered three levels: external, internal, application. External values can be presented to queries outside the organization. Internal values are for intra institutional use, and as such should not be presented to external queries. Application values are only pertinent to a given application and should not be presented.

urnRegValuePresentation human readable representation for the URN value. This attribute allows for multilingual values according to RFC 3866. For example, a URN for populating schacHomeOrganizationType of a person from the University of Salamanca (Spain) is urn:mace:terena.org:schac:schacHomeOrganizationType:int:university. The values for this attribute could be uRVP;lang-en: university, rRVP;lang-es-ES: universidad, uRVP;lang-de: Universität, and so on.

urnRegValueUser this attribute is mainly for internal use. Once there is an URN value registry, it becomes useful to register applications that depend upon a given value, mainly for change impact assesment.

5.2 Object classes

urnRegDescription contains the following attributes: urnRegValue, urnRegValueDescription, urnRegValueVisibility, urnRegValuePresentation and urnRegValueUser.

urnRegDelegation contains the following attributes: urnRegValue, urnRegOwner, urnRegHttpsURI urnRegLdapURI. urnRegRegistryURI and urnRegPolicyURI.

5.3 Web service functions

getURNdata get all information about a given URN value, regardless of it being a delegation or a normal URN.

getURNlist gets the list of URN values stored in the repository. Input: delegations, descriptions. Output: the list of URN values from the repository.

5.4 LDAP schema definitions

5.5 XML structures