



Proposed Architecture and Workflow for Managing Decentralized Information on Organizations Using a Central Registry File

Background

In the context of the initiatives set in place by the Expert Consultation on International Information Systems for Agricultural Science and Technology (Rome, 2005), FAO, GFAR and Wageningen International have proposed an Application Profile (a metadata format) for describing organizations. The Application Profile was presented at the Content Management Taskforce meeting in Wageningen (March 2007) and has been available from the FAO AIMS website (http://www.fao.org/aims/ap_applied.jsp) since 21/03/2007 for comments and feedback.

During the same meeting in Wageningen, GFAR also raised the issue of what the overall workflow should be in managing information on organizations in a decentralized way in order to avoid the proliferation of databases and the problem of duplication, and proposed a distributed architecture with a central Registry File in order to manage information on organizations uniformly on a global scale. GFAR has been asked to follow up with a project proposal in collaboration with FAO and other members of the Content Management Task force.

Proposed architecture

The project is based on a distributed architecture with a central Registry File in order to manage information on organizations uniformly on a global scale.

The elements of the proposed architecture are:

- 1. Data providers:**
the organizations themselves: each organization should describe itself and the description should be in the form of an XML record (compliant with the Organizations Application Profile) stored in a file. If the organization has the capacities, they can create the XML record and register the URL of the file with the central Registry File, otherwise they can use the services of a gateway provider (see below). The description should always be editable by the organization.
- 2. Registry File:**
through a simple web application, URLs would be appended / updated to a central Registry File. The format of the Registry File would be that of XML records with only two elements per record: a unique identifier (see below the paragraph on Unique Identifiers) and the URL for retrieving the XML description. An additional element could be a reference email (see below, "Open issues: spamming and duplicates").
- 3. Gateway providers:**
organizations that can provide facilities for other organizations, like:
 - web tool for creating the XML record;
 - web hosting for the XML file with related tools for updating the record and registering it with the Registry File.
- 4. Service providers (harvesters)**
all the organizations / services that want to provide information services based on the descriptions of the organizations.
Service providers would look up the Registry File, harvest all the URLs, read the data that they need and manage those data as they see fit.
Since the XML descriptions are based on the Organizations Application Profile and created by the owners themselves, they will be very informative and allow for the implementation of very structured information services.

Workflow

1. Case 1: organizations with the capacities to create the record and host it:
 - a) the record is created and placed at a public URL on the organization's web server; the record must contain the ID, which can be generated based on a specific algorithm (see later on), preferably using the tools provided by gateway providers;
 - b) through the web tool available with the central server where the registry File is hosted, the organization registers the URL of the XML file; the only input of the web tool would be the URL and the identifier.
The application would check the validity of the XML file at that location, check the ID, create a new record / update the record in the Registry File;
 - c) subsequently, the record can be updated at its location with no need to update the Registry File;
 - d) only in cases when the URL changes, the Registry File should be updated through the same web tool, providing the identifier and the new URL.

1. Case 2: organizations with no hosting and / or technical capacities:
 - a) the record can be created through the web tools provided by a gateway service;
 - b) the record can be also hosted with the gateway service: the gateway service automatically generates the ID, registers the new URL and returns the identifier to the owning organization;
 - c) subsequently, the organization can update its record through the same tool on the gateway provider's website, using its identifier, with no need to update the registry File;
 - d) only in cases when the URL changes, the Registry File should be updated (this would be done by the gateway provider).

Unique Identifiers and URNs

Theoretically, the Registry File could just consist of a list of URLs, with no need for identifiers, since the URLs, though not permanent, are unique.

However, attaching a unique identifier to an organization allows to: a) change the URL of the record without creating a second entry in the registry file (while leaving the previous no more valid URL there); b) (to a certain extent) avoid duplication; c) performing faster harvesting; d) most important of all: create and maintain relations between the records (impossible with URLs since they can change): notice that the Application Profile for describing Organizations has an element with different attributes for describing different types of relations with other records (organizations), therefore unique identifiers are an issue for the AP as well.

In order to avoid heavy manual work loads for any of the involved organizations, unique identifiers should not be manually assigned, but automatically generated based on data exposed in the XML file.

The main issues concerning unique identifiers are:

1) The criteria for generating unique identifiers

Rather than establishing an assigning authority, we propose to devise an algorithm so that the ID can be generated also locally and is repeatable. Every feature of this architecture should enhance the distributed approach.

Examples of generated IDs:

1. *[acronym]-[country/ISOcode]-[stringFromName]-[checkDigit]*: where the stringFromName could be a string generated using some characters from certain fields (e.g. the name, using the italian Tax ID algorithm using 12 letters): example: GFAR-ITA-GLBLOARSRCHE-B;

2. *[domainext].[domain].[path]*: (with slashes replaced by dots, example: org.egfar.public.description.xml): of course if the URL changes the ID does not. This option though seems too tied to a location (URL).

2) Metadata implementation

Usually, metadata store unique identifiers in an ID-type attribute. This also allows to refer to records using IDREF-type attributes in other records.

We propose to adopt this standard solution in the first phase and then, if URNs are adopted, a URN containing the unique identifier could be used in the Dublin Core identifier element (URNs cannot be stored in ID-type attributes, as these only accept non-colonized names)..

3) Future solutions: URNs?

since there is an agreement to adopt standards whenever possible, a way to ensure that IDs are universally

unique (not only in our system) would be using the generated IDs to create URNs (Uniform Resource Names), which, by definition, "are intended to serve as persistent, location-independent, resource identifiers", and matching them to the corresponding URLs.

The proposed syntax (following the syntax defined in "[URN Syntax](#)" by IETF) is:

`urn:[namespace]:[uniqueString]`

Elements:

urn: the protocol;

[namespace]: the "Namespace Identifier": this can be whatever we choose, a string identifying the context of organizations in the field of agriculture: in view of the future wide use of URNs and of the setting up of a naming authority, choosing a namespace that can be easily reserved by GFAR / FAO / the ARD Community would be a good idea; for the moment, we propose and use "agri";

[uniqueString]: the "Namespace Specific String": this would be the string generated according to the criteria indicated in par. 1, with a specific syntax to be defined; considering the foreseen usage of the "agri" namespace for other sub-domains (documents, projects etc.), the first element in this syntax would be the sub-domain "org".

Proposed solution for URNs:

"urn:agri:"*NSS*

NSS = **"org:"***identifier*

identifier = *acronym/initials* - *countryISOcode* - *stringFromName* - *checkDigit*

Examples:

urn:agri:org:GFAR-ITA-GLBLOARSRCHE-B

urn:agri:org:FAO-ITA-FDOOXNTNSAI-T

Open issues

1) Spamming, errors and duplicates

These issues are all related to the possibility for anyone to register a new URL with the Registry. Spammers could just register URLs of well formed XML files containing any kind of unwanted texts and links; or the organizations themselves could register different URLs by mistake (with different gateways or just at different times without providing the identifier). It is true that the identifier is generated from the data exposed and theoretically two registrations of the same organization should generate the same identifier, but even a slight difference in the description could generate a different identifier and the duplication would go unnoticed.

A solution against spamming could be that of storing a reference email in the Registry and sending a request for confirmation before activating the record.

Storing a reference email could also help in avoiding duplicates: an automated procedure could check for suspect duplicates and send requests for confirmation. Of course emails could change in time but if there are duplicate registrations the most recent one most probably has a valid email.

2) Email exposure

Even if there are no reference emails in the Registry, emails are exposed in the single XML descriptions. It is true that organizations usually publish a contact email on their website, exposing themselves to the same risks of spamming, but free access to the Registry file allows anyone to easily harvest a lot of contact emails which may be misused.

Should read-access to the Registry file be limited to "registered" information services?

Feedback and comments

Before proceeding to the next steps, we would very much appreciate feedback on this proposal, particularly on the following points:

- Unique identifiers: criteria: do you agree with one of the criteria indicated in this document (if yes, which one?) or do you have other suggestions?
- Solutions for duplicates and spamming: do you agree on storing reference emails in the Registry?
- Limit permissions to harvest through the Registry?

This paper will be used for the prototype in January 2008, so only comments received before then will be considered for the prototype.

Prototype

The prototype will consist of a suite of web tools for managing the system: a tool for creating XML descriptions compliant with the Organizations Application Profile and for generating unique identifiers, a tool

for registering / updating records in the Registry – with both a user interface for single organizations and a “web-services”-like interface for gateway automated services, a simple harvesting test application. The Registry will be hosted by GFAR and most of the tools will be available both on the EGFAR website and the FAO AIMS website. Any information system wishing to create new services can do so by either using the basic web services provided by EGFAR or harvesting directly from the Registry.

Next steps

- 1) A test phase with a few “pioneer” organizations and possibly an example of service implementation from a service provider.
- 2) Launch: identification of the largest number possible of organizations willing to participate in the launch of the system. The larger the number of organizations involved in the launch, the more easily other organizations will wish to participate.
- 3) Full implementation of the system and focus on advanced information services.

Outline of the architecture and workflow

