

MIWP-6: Registers - Discussion #2615

[Register Federation] Fields for the Register exchange format

26 Nov 2015 09:34 am - Daniele Francioli

Status:	New	
Priority:	Normal	
Assignee:		
Description		
The proposed mandatory and optional fields for the RoR (Register of Registers) are described in the Registry federation requirements page .		

History

#1 - 26 Nov 2015 09:39 am - Daniele Francioli

- Description updated

#2 - 26 Nov 2015 09:47 am - Daniele Francioli

Here you can post your comments related the Fields for the exchange format.

#3 - 02 Dec 2015 02:37 pm - Christian Ansorge

Thank you very much.

In the meantime I also had an discusson with Michael L who explained me your reasoning for removing the pointers on the concept level.

2 quick reactions from my side:

1. One of our requirements is that we can point from the concept level. This come from the situation that we handling concepts extending Inspire and concepts not extending Inspire (eg. Coming from reporting) in the sam codelists. When just pointing from the level of the concept scheme we could not differentiate between those two categories. This was the reason why we were pushing so much for multiple inschemes from concept level.

2. I had a look at the voaf and it is indeed what we were looking at (at least for the level of conceptscheme). Why not usinf voaf:extend which seems even better suited? But I had only a very quick look and maybe miss the reasoning behind voaf:relieson

When back in the office I will have anothr look on voaf

Cheers

Chris

#4 - 02 Dec 2015 03:37 pm - Daniele Francioli

Thank you for your comments.

1. In that case probably it's better to replicate voaf:reliesOn at the concept level.

2. We decided to use the voaf:reliesOn instead of the voaf:extend because basically the second one can only express extension, whereas with the reliesOn we can potentially also express other type of relations (like subset or profile).

Below I've copied the definition of the two terms from [1]:

extends - Indicates that the subject vocabulary extends the expressivity of the object vocabulary by declaring subsumption relationships, using object vocabulary class as domain or range of a subject vocabulary property, defining local restrictions etc ...

relies on - Indicates that the subject vocabulary **uses** or **extends** some class or property of the object vocabulary

Cheers,

Daniele

[1] <http://lov.okfn.org/vocommons/voaf/v2.3/>

#5 - 02 Dec 2015 05:16 pm - Christian Ansorge

Thank you Daniele,

Just to come back on the reliesOn on concept level. In the voaf definition the domain is vocabulary. Do you see it as feasible to use the voaf:reliesOn also on concept level? Maybe interpreting concept as a subset of voaf:vocabulary? Sorry if this question is maybe a bit immature or stupid, I still have to check the vocabulary in detail.

Cheers

Chris

#6 - 02 Dec 2015 08:59 pm - Michael Lutz

I think there is a bit of a misunderstanding here. What Christian meant (I think) that it is important to be able to say that a concept belongs to different concept schemes, e.g. the value <http://inspire.ec.europa.eu/codelist/CurrentUseValue/publicServices> in the example is in the concept scheme <http://inspire.ec.europa.eu/codelist/CurrentUseValue> and in the concept scheme CurrentUseValue (the extension). These relationship should be expressed by the inScheme relation as discussed previously.

Then, we may want (and in the case of empty code lists need) to express that one concept scheme has a dependency on another one. This should be expressed by the reliesOn relation.

#7 - 02 Dec 2015 09:07 pm - Michael Lutz

Another issue I would like to raise: Do we need to repeat any statements about the concepts that are re-used from another register, e.g. their prefLabel, definition, etc. but also their broader/narrower relationships? Maybe it is enough to just list their ids and any statements going beyond what is already stated in the original register, e.g. the fact that they are also in the concept scheme CurrentUseValue (the extension).

Thus, probably, we could simplify the statement above to

```
http://inspire.ec.europa.eu/codelist/CurrentUseValue/publicServices">  
http://www.w3.org/2004/02/skos/core#Concept"/>
```

Alternatively, we could have a full description for all concepts.

#8 - 07 Dec 2015 03:54 pm - Christian Ansorge

Ok, just let's come back on the elements needed. We started our work on the exchange file with the requirement (by JRC registry team) that the exchange file should be as slim as possible allowing the basic usecases SEARCH ITEM and REGISTER ITEM. For the search item we had some discussions in Lisbon if we can include (and index) the label or not in the exchange format. In the end we considered it as necessary for effective searching. Furthermore we do not intend to replace the national and local registers. By blowing up the scope of the exchange file we risk to enter quickly the level where we would infact fully replace the local registers (just that the URI still points externally but all elements would be in the RoR) which risks to undermine the local effort to promote, implement and manage their registries. I would be careful on this topic, because I (as MS) would ask why we have to set up a register in the first place as anyway everything is hosted by the RoR? The business case for a local registry would than be extremely limited. Our goal should be to make concepts discoverable but to link back to the local level as soon as possible.

I would take the discussion for additional elements from the angle of the use cases. What we need to serve the use cases (to search the RoR for something; to make items available on the RoR)

- Definition - The definition would be indeed useful for effective search functionality but in my point of view there is also a risk. The definition can become quite extensive which might result in larger size of the exchange files (as slim as possible?). Furthermore is there a inconsistency, as definition element is often not provided anyway. I think here we have to take a decision.

- PrefLabel, Broader, Narrower, etc. - I don't see this elements as necessary for the operation of the RoR. Our use case is to list registered concepts from the federation entities and to make them (to a certain extent) searchable. I see the use of this attributes on the level of the local thesaurus but not so much on the level of the RoR. The [HILUCS list](#) in LU is a good example for a hierachical codelist. In the end all elements (Tier 1 - Tier 3) point to the same codelist. So from the RoR point of view (as we come from the perspective of a flat list of terms) we just need the direct counter part of a list (be it internally hierachically structured or not), in the case above it would be the HILUCS codelist. Therefore I would suggest to disregard the statements like prefLabel, broader or matches as they are irrelevant for the search usecase.

Just my thoughts on the topic, for sure there are other opinions. :-)

#9 - 11 Dec 2015 02:34 pm - Michael Noren

Christian, while I won't argue on the need for local registries, from the goal of information re-use I don't see them as the key ingredient. If we put the practical implementation aside for a moment, my main goal as a user of the federation is probably to find extensions I can re-use. To assess if I can re-use them I need to know enough detail about them, e.g. the definition.

Comparing this scenario to the problem DCAT is trying to solve for open data, where previously the user had to search each local portal separately, trying to understand the navigation and search logic of each, probably also in the national language of the hosting country for the portal. To solve this, the open data catalogues should be published in DCAT, similar to what we aim to do with our common exchange file, so it can be indexed and made searchable on any other portal (for example the RoR in our case). The user is then free to use the portal he/she understand and finds usable. Here we can of course argue what is enough information that needs to be supplied for anyone to implement a useable search function.

The second part is when I have found my dataset (DCAT) or code list extension, how can I get access to the full information? With DCAT there is the distribution information, but that might just link to a landing page and then you are still faced with navigating a national portal (with all the issues of that). In our case it would be the link to the national registry, where there needs to be a page for the linked URL with the full data. Similarly to the DCAT scenario, again you have to face all the issues of using a new portal (navigation, presentation, search logic, language...).

In our case it seems the difference between the basic metadata and the full data is quite minimal, so why not require enough information to make it usable on its own and avoid the above problems? If we do that anyone can setup a search portal (or harvest the full data for own use), countries can join the federation even if they don't have registry, or a registry that cannot be linked to like we require, and people can more easily find extensions and

directly assess if they can use them. The downside is that each contributor has to provide a little more information.

If you don't have the information it will be empty in the exchange file, but also in the other scenario when the user follows the link to the national registry for the full information, so no difference really. It will be more data to interchange, but the amounts we are talking about is still really small. It will be extra work for the participants to add this data to the exchange file - yes, but if you have a registry you probably generate this file anyway, and if you don't, indeed you will need to do extra manual work, but unlike the other scenario, you can still participate in the federation.

It got a bit long, but does it make sense?

#10 - 14 Dec 2015 09:47 am - Daniele Francioli

- Description updated

#11 - 13 Jan 2016 12:15 pm - Daniele Francioli

- Description updated

#12 - 18 Jan 2016 09:42 am - Heidi Vanparys

Michael Lutz wrote:

Another issue I would like to raise: Do we need to repeat any statements about the concepts that are re-used from another register, e.g. their *prefLabel*, *definition*, etc. but also their broader/narrower relationships? Maybe it is enough to just list their ids and any statements going beyond what is already stated in the original register, e.g. the fact that they are also in the concept scheme `CurrentUseValue` (the extension). [...]

I would say that repeating any statements about the concepts that are re-used from another register is not needed. It may actually introduce inconsistencies, if a property is changed in one registry and this change is not reflected in another registry.

#13 - 18 Jan 2016 11:47 am - Daniele Francioli

- Description updated

#14 - 18 Jan 2016 11:48 am - Daniele Francioli

- Description updated

#15 - 18 Jan 2016 11:48 am - Daniele Francioli

- Description updated

#16 - 20 Jan 2016 10:59 am - Daniele Francioli

- Description updated

#17 - 20 Jan 2016 11:06 am - Daniele Francioli

- Description updated

#18 - 22 Jan 2016 11:53 am - Daniele Francioli

- Description updated

#19 - 22 Jan 2016 11:55 am - Daniele Francioli

- Description updated

#20 - 22 Jan 2016 12:13 pm - Daniele Francioli

- File *eionet_DesignationSchemeValue_example.rdf* added

- File *inspire_DesignationSchemeValue_example.rdf* added

- Subject changed from [Register Federation] Fields for the exchange format to [Register Federation] Fields for the Register exchange format

- Description updated

#21 - 22 Jan 2016 12:14 pm - Daniele Francioli

- *File Register_exchange_format_proposal_annotated.docx added*

- *Description updated*

#22 - 22 Jan 2016 01:44 pm - Daniele Francioli

- *Description updated*

#23 - 05 Feb 2016 05:24 pm - Michael Noren

As discussed in #2661, don't we need a "status" for the concepts and also for the conceptscheme? Maybe it's my lack of understanding, but how else can one something be deprecated/deleted/non-valid? There is no method for deleting a published extension from the federation (and it should not be possible since it might be used by someone), so the only method left is to update the extension with a new status?

#24 - 10 Feb 2016 03:54 pm - Christian Ansorge

Michael and I had some internal discussions about the current RoR Architecture. Thnaks to Michael to writing it down:

Add status property?

How are updates to code lists and code list values provided beyond updates of existing information? If a concept is provided in harvest #1, but is not there in harvest #2, what happens then, will the RoR simply delete it or will it continue to exist as it was in harvest #1? If someone is using (referring to) this concept, just making it disappear is perhaps not great, so here it might be useful with a status element that the provider can use to indicate that it is deprecated or similar?

#25 - 11 Feb 2016 02:25 pm - Michael Lutz

Hi Christian and Michael,

we are proposing the following behaviour in the RoR/federation for the different use cases. In general, we think it is important to distinguish from what information is available in the RoR (federation) and what information is available in the local registers and between guidelines for making your register available in the federation and guidelines for "good" register management.

Use case: Browse register relations

The RoR will store only a reference (URI) to the registers and relations between them. In the browse interface (the one already available here: <http://inspire-regadmin.jrc.ec.europa.eu/ror/>) you can then see only the reference to the Registers and relations. No information related to the items or any other metadata will be stored in the RoR database.

In this context, deleting a register from the registry descriptor means that you want to remove the register from the federation.

Federated search use case

The search index will be created using the following (draft) elements:

- URI
- Label
- Definition

Even in this case, the information are not stored directly in the RoR system. During the harvesting, the information (listed above) read from the RDF

exchange files are used to create the search index (then passed to a search engine that is an external application - like Apache SOLR). You can basically search and filter for one of these fields. We are evaluating if the status of the item could be one additional field to be added to the search index (and then to the Register exchange file). It could be useful to search only on "valid" items.

In case you remove a specific item from the Register RDF file, it will be deleted from the search index. Please note that removing items from a local register is a "bad practice" for the management of registers (which we could address in one of the FAQs - e.g. Q "what do I do if I no longer need a value in my register?" A "Don't delete it, but mark it as retired or superseded").

The JRC Registry Team

#26 - 11 Feb 2016 04:11 pm - Christian Ansorge

Hej,

Thanks for the quick response.

- I think the core of what we wanted to say is, that we should look into the need for a status property on the concept level. In our initial proposal we had the status of the concept as mandatory property and we still think this information is essential and should be provided. If it will not be provided (missing in the latest proposal for a exchange file at least) we would need clear rules of what to provide and what happens if concepts suddenly are no longer provided. But as you said you are considering it, I am confident to get it in again. :-)

- Concepts which are (for whatever reason) no longer provided in updates of registers, should maybe be flagged as retired (as no other information available) and not be removed from the search index.

- Of course we will not delete concepts, but give them the status retired or superseded. Our question came from the background if we then shall only provide valid concepts or all concepts of a register (including superseded, invalid and retired). Here we need another decision on what we actually include in the exchange and what in the search index.

Cheers

Chris :-)

#27 - 11 Feb 2016 05:31 pm - Michael Lutz

I think that for the search use case, it is indeed important that the register exchange file is complete, i.e. also include deprecated and superseded items (we can discuss whether invalid ones should also be included), so that they can be found in the RoR (if you include them in your search). The status is probably not included in the proposed exchange format because it is not relevant for the browse extensions use case and because we have not yet looked at the search use case in detail.

So I would propose to include it for now and see whether we really need it (if not, we can still throw it out later).

Michael

#28 - 12 Feb 2016 10:36 am - Christian Ansorge

:-)

#29 - 24 Feb 2016 09:43 am - Daniele Francioli

- *Description updated*

#30 - 24 Feb 2016 09:43 am - Daniele Francioli

- File deleted (*eionet_DesignationSchemeValue_example.rdf*)

#31 - 24 Feb 2016 09:44 am - Daniele Francioli

- File deleted (*inspire_DesignationSchemeValue_example.rdf*)

#32 - 24 Feb 2016 09:44 am - Daniele Francioli

- File deleted (*Register_exchange_format_proposal_annotated.docx*)