

Statements for user profile standard development (for discussion)

Here are some statements for discussion that may help us in determining our strategies and planning of work. Please feel free to add your comments to each of the statements, or add statements yourself.

Gottfried

Erlend: My view on why we should choose RDF

The development of a set of preference items for user interface adaptation should be based upon various use cases, including a range of user groups and a range of contexts of use.

- Use cases need to be defined, and different kinds of adaptive user interfaces constructed for testing and evaluation purposes.
- This is what Cloud4All is doing through its SP1 and SP3, on various platforms. This takes time and resources, and requires the involvement of many partners.
- Other projects (aside from Cloud4All) may have done or may be doing similar activities. This working group is suitable for information exchange and coordination of these efforts.

The set and structure of preference items for user interface adaptation depends on the context of use (applications, platforms, assistive technologies). There is no universal set of preference items that would suffice for all existing contexts of use.

- This is an implication of the first statement.

The set of preference items for user interface adaptation will never be complete. New applications, assistive technologies and user interaction techniques will eventually require an extension of any existing set of items.

- What if we want to accommodate new interaction techniques that require new preference items? For example, consider the introduction of sign language output by avatars. We would need a new preference item for the speed of the sign language performance, for the preferred avatar, etc. Or for gesture-based input we would need additional preference items that specify a gesture alphabet.
- We need to define a framework that sets the structure of a user profile independent of its vocabulary items.
- The vocabulary (preference items) should be repository-based so that it can be updated on a regular basis.
- The framework (without vocabulary) should be defined in an ISO standard. Also, a process of approval for new vocabulary items for the repository should be specified in a second ISO standard.
- For an efficient development process, properties, structure and approval process can be developed in parallel.

I agree with the need for an extensible set of needs and prefs (n&p). This suggests to me that the structure of the list will be very important. This is why I have always argued for a model that starts with a gross category and then refines it. This means that where systems do not actually have implementations that use the new items, they will at least know where they fit. I recommend the approach we have taken for the ISO Metadata for Learning Resources (MLR) determination of classes etc and use maps that connect n&ps so from the beginning we know what is a refinement of what. That will make it easier in the long run to provide an extensible set.

The second point is that I think when you say repository, Gottfried, we could ask Mitsuharu Nagamori for a version of the schema registry that the Dublin Core community uses. Mitsuharu is familiar with these problems and the registry is already built and tested

The third point is that ISO does not work through registries, as I understand - so perhaps we have to think how to do this for the ISO standard. I am proposing an appendix to the MLR that will link the definitions there to the DC terms registry. This may be one way to do it?

A flat set of properties (key-value pairs) is most appropriate for practical reasons.

- In most cases, hierarchical constructs of user properties can be approximated by a flat set of properties with URIs as keys.
- See as an example the WURFL properties database for mobile devices. This simple collection of key-value pairs for thousands of mobile devices is a de-facto standard in industry today for the user interface adaptation based on mobile device characteristics. WURFL is regularly updated by a community. <http://wurfl.sourceforge.net/>
- Many standards in this area have a flat structure, and some use URIs as keys (e.g. Dublin Core, ETSI ES 202 746). This is a pragmatic approach, and allows for core properties and extension properties based on domain names. Also, using URIs as keys allows for formal definition of preference items (and constraints) in RDF. Note that doesn't mean that the preference values need to be coded in RDF.
- The core items will have a commonly defined domain name (namespace) as part of their URI. Third parties such as user groups and vendors can define their own extension items by using a different domain (namespace).

Agreed!

Comment..

I know some people like the ideas to appear on this list but please can we make sure that all ideas end up on the wiki. Otherwise I think we will have overlapping conversations and end up with a mess!

I am lucky enough to be spending a few days with Eric Miller. I think there are major advances in how the web works and what is coming available. My discussions with him only increase my feeling that we should be organising the needs and preferences as they are discovered - that is, they should be defined and we should have a clear 'mapping' of how they relate one to another.

Andy (in email) has talked about relationships between needs etc. This is an important aspect of a need so that I can have a broad wish and you can have a very detailed and specific version of that. It is also very important so that if your system does not know about the detailed need you have, at least it will know that it is a qualified version of one of the things your system does know about.

Finally, it lets us be assured that we can always cater for a new or more precise version of a need - ie we can have an extensible set.

My understanding of the pairs that Gottfried is talking about then allows us to include the relationships in the definitions.

So while I am not sure what Andy is advocating, I ask yet again if we can please learn about the needs and preferences by talking to as many people as possible who can inform us about what they are, and then can we start to organise them, please.

In general, the user properties stored in a user profile should be based on requirements rather than functional descriptions of the user. However, functional descriptions may help to efficiently derive requirements-based properties for the initialization or fine-tuning of user profiles.

- Our framework should be able to accommodate both requirements-based user properties and functional user properties.
- There will likely be stricter ethical constraints for functional user properties than for requirements-based user properties. For example, functional user properties could be permitted only in a local environment and temporary context (no storage on a central user profile repository).