Proposal for Handling Partially Controlled Vocabularies

Authors	Tony Proctor
Created	2013-03-13
Title	Partially_Controlled_Vocabulary
Version	1.0
Type	Technical
Language	English
Description	Proposal for handling partially controlled vocabularies for types
Suggested Keywords	Tag-Values, Namespaces, Software-vocabulary

Contents

1. Abstract

Proposal to adopt a common framework for allowing extensible sets of tag values such as types. By allowing predefined sets to be extended, this would support the notion of *partially controlled vocabularies.*

The proposal eliminates the possibility of clashes, avoids the need for a registration scheme, and ensures the data is still transportable between differing products.

2. Proposal

Genealogy and family history involve a number of partially controlled vocabularies for its types, sub-types, and other taxonomies – collectively referred to here as “tag values”. These may include event types/categories, person roles/relationships, place types/categories, name styles, property names, source types, etc.

GEDCOM allowed some extensibility but only by prefixing the tag with an underscore. That simply declared it as foreign and provided no way to avoid clashes.

STEMMA implements partially controlled vocabularies using XML namespaces. Custom tag values may be defined by declaring a new namespace using the standard xmlns attribute in the Dataset header element. The prefix associated with that namespace can then be used to introduce custom tag values without any fear of clashing. For instance:

```
<Dataset Name=’Example’
        xmlns:MyEvents=’http://mydomain.com/myevents’>
...etc...
<Event>
        <Type> MyEvents:xyz </Type>
```

This mechanism uses the same XML namespace feature that prevents clashes between element names and attribute names from different origins. XML tag names are deemed to belong to a given namespace and may be qualified as a Fragment identifier. For instance, http://stemma.parallaxview.co#Dataset is the namespace-qualified name of the STEMMA <Dataset> element. Similarly, http://stemma.parallaxview.co/event-type#Union is a namespace-qualified Event-type. The main differences here are that these namespaces apply to tag values, and they are local to the associated Dataset.

An XML parser normally discards any such prefixes once the XML has been loaded since they basically just connect names (and values) to their respective namespace declarations. The exception to this is when they have been employed in the value of attributes, or in element data, as is the case in STEMMA and SOAP. The prefix-to-namespace mapping then has to be retained and made available to the program loading the XML. This is why the tag-value namespaces are expected to be associated with the enveloping Dataset element rather than any of elements below it.

3. Not Covered or Not Required

The illustrations here use STEMMA syntax since that model has already successfully used this scheme. However, the specific syntax is not a mandatory part of the proposal.

This proposal describes a solution based around XML concepts such as namespaces. If the Data Model standard specifies additional serialisation formats then we should strive to accommodate the same basic principle in those formats.

The illustration uses the general scheme to define custom evidential properties for a person. However, these also need support for data-types and cardinality and so must be the subject of a separate proposal.

4. Illustration

This STEMMA example shows a Person that has multiple custom roles in relation to a club social event. It also defines a custom property to accommodate his membership details.

```
<Dataset Name='Multi_Role_Example'
xmlns:roles='http://mydomain.com/roles'
xmlns:props='mailto:name@emaildomain.com?subject=properties'>

<ExtendedProperties>
        <PersonProperties>
                <PropertyDef Name='props:MemberID'  Type='Integer'/>
        </PersonProperties>
</ExtendedProperties>

<Event Key='eClubSocial'>
        <When><Date><Value> 1960-06-09 </Value></Date></When>
        <Type> Social </Type>
</Event>

<Person Key='pGordonBennet'>
        <EventRef Key='eClubSocial'>
                <Property Name='Roles'>
                        <Item> roles:Photographer </Item>
                        <Item> roles:Host </Item>
                </Property>
                <Property Name='props:MemberID'> 2314 </Property>
        </EventRef>
</Person>

</Dataset>
```

In other words, Gordon Bennet was both the host and the photographer at the club meeting, and his membership ID was 2314.

5. Use Cases

It will be impossible to predefine all acceptable tag values in a standard and so the Data Model must accommodate an easy way of extending the predefined set, and in such a way that the data remains transportable between different products.

Without a reliable and easy-to-use framework then vendors and users may resort to private methods outside of the standard.

6. References

STEMMA Extensible Vocabulary.
http://www.familyhistorydata.parallaxview.co/home/extensibility/extended-vocabularies.

STEMMA Extensible Properties.
http://www.familyhistorydata.parallaxview.co/home/extensibility/extended-properties.