Creating semantic relations for user manuals on Semantic Web using SKOS

SIMONA-ELENA VÂRLAN
Assist. at. Department of Computer Science*, Phd Student at. FEAA**
University “Vasile Alecsandri” of Bacău*, University „Al. I. Cuza” of Iaşi**
Calea Mărăşeşti no. 157
ROMANIA
varlan_simona@yahoo.com

Abstract: The aim of the paper is to create a standard model representation using SKOS for user manuals to benefit from an automatic mechanism of information and explanations representation necessary for each software applications. The SKOS (Simple Knowledge Organization Systems) is a standard for representing concept schemes for different types of KOS (Knowledge Organizations Systems). This method of knowledge representation and semantic data modeling can be an infrastructure upon which applications can be built for automatic data processing. From this can benefit in particular applications for economic transactions such as applications for expert systems in making decisions and solving problems. Creating a standard for representing user manuals any software application will be able to parse document content and manipulate and link data with other data on web.

Key-Words: Semantic Web, KOS, SKOS, XML, RDF, OWL

1 Introduction

The idea is to manipulate data with any other computer program not only through a browser [1]. All the controled vocabularies play an important part in development of a web of data [2] by linking concepts within the documents on the web (linking data on the web). SKOS vocabulary purpose is to provide an effort to minimize costs and maximize results when working with applications using controlled vocabularies. SKOS provides a simple yet powerful medium for representing structured knowledge capable of being understood by the computer for utility in the Semantic Web.

A user manual fits into KOS categories so we can use a specific semantic web language like SKOS for knowledge representation to create a standard model representation for user manuals. Semantic Web already offers semantic languages like RDF (Resource Description Framework) and OWL (Web Ontology Language). RDF is a specific language to create statements about resources, but provides only the minimum semantic level required for metadata assertions. OWL language on the other hand, provides the necessary semantic to describe resources, but require effort, expertise and therefore costly, because it is a language-oriented modeling classes requiring a very precise sense.

To describe the concepts and conceptual schemes we used SKOS Core Vocabulary [3], [4] because SKOS is a simple language like RDF that does not require much effort and expertise, but is sufficient for defining complex conceptual structures and to support semantic search like OWL does. With SKOS practically we can create sets of classes from any kind of organized date.

2 Problem Formulation

Each software application is made accompanied by a user’s manual, in different ways. User manuals are actually a set of structural terms so we could create an automatic mechanism for representation of specific information of these manuals. Achieving this is the objective of this paper.

If we create a standard representation of information for the user manuals than any software application can parse, analyze and process retrieved information and we can link data on web more easily. This reduces time and effort.

3 Problem Solution

A manual or user guide is a technical communication document intended to assist users. A manual is based on concepts that are different issues covered; each can have a label and a definition. It can use classes of concepts that contain the same domain concepts. For example, a user guide should contain a series of instructions and these instructions can be grouped into one class. A manual also contains a glossary

1 Semantic Web: http://www.w3.org/2001/sw
of terms used, and a glossary representation is made also in SKOS format.

To cover all the requirements and characteristics of representation of a user guide can be used in combination with other semantic vocabularies such as OWL, RDF, FOAF (Frind of a Frind). SKOS has been designed for use in a multilingualism environment therefore containing tags for specifying the language used in the labeling and description of concepts. SKOS provides a set of semantic relationships between concepts, including skos: narrower, skos: broader and skos: related. We use to represent relationships between concepts in the same area property skos: related.

User manuals require a number of fields designed to provide additional information on paper and general notes, source data, historical data, and examples. SKOS vocabulary also includes properties such as: skos: note, skos: editorialNote, skos: definition, skos: scopeNote, skos: changeNote, skos: historyNote.

User manuals using SKOS representation requires compliance with a basic model. Since both SKOS and user manuals are oriented conceptual models, the choice of representation in SKOS vocabulary is the best solution. SKOS vocabularies contain similar instances of type skos: Concept, which combines a web resource, often represented by a URI. SKOS concepts have lexical labels and documentation about these labels, and may refer to other concepts using a variety of semantic relations. All these language properties can be adapted to the requirements of the standard representation of user manuals.

For a SKOS representation of user manuals we must initially define the basic concepts with which we work. These concepts are characteristics of guidelines that must be mapped to SKOS.

Is presented in Table 1 a list of such mappings:

<table>
<thead>
<tr>
<th>RDF Properties</th>
<th>Meaning of property / Commentary</th>
</tr>
</thead>
<tbody>
<tr>
<td>rdf:about</td>
<td>URI for instance skos:Concept</td>
</tr>
<tr>
<td>dc:title</td>
<td>Title of the user manual</td>
</tr>
<tr>
<td>doc:version</td>
<td>Doc version</td>
</tr>
<tr>
<td>skos:definition</td>
<td>Definitions for concepts</td>
</tr>
<tr>
<td>skos:prefLabel</td>
<td>Preferred lexical label for Item</td>
</tr>
<tr>
<td>skos:altLabel</td>
<td>Alternative lexical label to name the item</td>
</tr>
<tr>
<td>skos:related</td>
<td>Association relationship between concepts</td>
</tr>
<tr>
<td>skos:note</td>
<td>Details that are not for general public</td>
</tr>
<tr>
<td>dc:date</td>
<td>Historical document, design, etc..</td>
</tr>
</tbody>
</table>

Table 1: List of mappings

The list above contains a number of RDF tags representing sufficient minimum information necessary to develop an environment for a user manual. This is due to the heterogeneity and multiple possibilities of arrangement and structuring of information using a limited set of tags reduces overall assembly complexity.

The definition requires the use of tags in a minimum-sufficient (as few possible tags) so that the representation of information may require a small amount of work but to provide a large volume of granularity. It is therefore necessary to obtain a separation between data definition and data format required for generating comprehensible user information, leading to a greater simplicity of implementing the system. For each property separately described the significance and the role it carries in SKOS representation.

Next we show SKOS representation of the conceptual scheme for a user manual.

SKOS representation is as follows:

```xml
<?xml version="1.0" encoding="utf-8"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:skos="http://www.w3.org/2004/02/skos/core#"
  xmlns:dc= "http://purl.org/dc/elements/1.1/"
  xmlns:doc="http://www.w3.org/2000/10/swap/pim/doc#">
  <skos:Concept
    rdf:about="[http:// Web address used for reference ]"
    dc:date="[ date of last modification ]"
    dc:title="[ manual title ]"
    doc:version="[ version ]">
    <skos:prefLabel >
      { Technical name / Short name for the documented segment }
    </skos:prefLabel>
    <skos:altLabel>
      { Full name of the documented }
  </skos:Concept>
</rdf:RDF>
```
SKOS code above uses namespaces required for validation of RDF properties. These are: rdf, skos, dc and doc. If file validation is successfully completed, it will display a tree representation of the tags contained. Figure 1 capture this tree representation. As you can see in the left corner is the main tag, below its properties, and then child tags.

Fig.1: Tag Tree

Skos file used as an example here, meet the standard described above, having added only the necessary information described in terms of the manual. This is an example for describing the HTML tag used within it. Being a demo application only some tags were described, in the near future the example will be developed.

This is the SKOS file used in the above example:

```xml
<?xml version="1.0" encoding="utf-8"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:skos="http://www.w3.org/2004/02/skos/core#"
  xmlns:dc= "http://purl.org/dc/elements/1.1/"
  xmlns:doc="http://www.w3.org/2000/10/swap/pim/doc#">

  <skos:Concept
    rdf:about="http://www.w3.org/TR/html401/"
    dc:date="24 Dec 1999"
    dc:title="Manual HTML"
    doc:version = "4.01">

    <skos:prefLabel>
      HTML 4.01
    </skos:prefLabel>

    <skos:altLabel>
      Hypertext Markup Language 4.01
    </skos:altLabel>

    </skos:Concept>

</rdf:RDF>
```
4 Conclusion

SKOS offers representation structure and content of controlled vocabularies in a way that can be understood by the computer. Using SKOS Core Vocabulary can define complex conceptual structures used to generate metadata as ontology languages such as OWL does, but in a much simpler and less demanding in terms of expertise and effort.

We don’t expect a massive use of SKOS Core, but is one of the best ways to create controlled vocabularies because it offers the advantages: eliminating ambiguities between different concepts with the same meaning easily extend the syntax, could be combined with other semantic languages to cover all of the different vocabularies.

Acknowledgment

This work was supported by CNCSIS-UEFISCSU, project number PN II-RU code 188/2010.

References:

[13] M. Frederiksen, SKOS Output from Binary Relations Weblog, see http://www.wasab.dk/morten/blog/archives/author/morten/skos.rdf