Data-driven ontology engineering with OTTR: benefits and challenges derived from practical experience

3rd OTTR user forum, 2022-05-11

Moritz Blum, Basil Ell
Bielefeld University
{mblum,bell}@techfak.uni-bielefeld.de
Data-driven ontology engineering with OTTR

- DiProMag Ontology for the semantic description of experiments: production, characterization and prototypical application of magnetocaloric alloys
- A-Box and T-Box should be populated: parallel development of templates and ontology
- Manual and partly automated instantiation through domain-experts
  - Propose extensions/changes of the ontology
  - Understand the underlying structure
- Semi-structured data is given - some experiments still exploratory without a fixed parameter space
  - Measurement devices: input specification, settings, output
  - Production process: sequence of operations applied to objects
  - Goals and incentives

Fig: overview of the experiments carried out in two of our working groups

Read more about DiProMag on our project website: [dipromag.de](http://dipromag.de)
Methodology

Iterative process that allows backward steps -> simplifies and improves communication with domain experts, enforces a bottom up / data-driven approach
Communication with domain experts

- tables help to maintain an overview: datatype, parameter name, example
- documentation and variable naming
- coloring of rows to group concepts

<table>
<thead>
<tr>
<th>dpm:Material</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>otr:IRI ?material,</td>
<td>dpm:Co2CrAl0.0001Si</td>
</tr>
<tr>
<td>NEList<a href="">dpm:Element</a> ?elements,</td>
<td>(dpm:Co, dpm:Cr, dpm:Al, dpm:Si)</td>
</tr>
<tr>
<td>NEList<a href="">xsd:float</a> ?stoichiometry_portion_unit_at_percent,</td>
<td>(2,1,1,0.0001)</td>
</tr>
<tr>
<td>dpm:material_derived_from ?method_of_derivation,</td>
<td>dpm:addition_material_from</td>
</tr>
<tr>
<td>dpm:Material ?base_material,</td>
<td>dpm:Co2CrAl</td>
</tr>
<tr>
<td>dpm:User ?user,</td>
<td>dpm:BasilEll</td>
</tr>
<tr>
<td>xsd:date ?date,</td>
<td>“2022-05-11”^^xsd:date</td>
</tr>
<tr>
<td>xsd:string ?comment,</td>
<td>“Si for better… ”^^xsd:date</td>
</tr>
<tr>
<td>xsd:boolean ?verified]</td>
<td>“false”^^xsd:boolean</td>
</tr>
</tbody>
</table>
Workflows

- manual instantiation by domain experts requires workflows of template instantiations and naming conventions
- relations between templates define the workflows
- conceptual relationships are mirrored in the instantiated template signatures and the instantiated triples
Experience

- strictly defined workflows reduce the requirement of naming conventions
- T-Box is distributed across templates vs. central T-Box template(s): duplication and knowing where → Protegé in parallel to maintain overview
- good experience with Semantic Media Wiki (SMW) as central platform for managing template definitions, instantiations, and documentation
- the OTTR SMW extension was developed and is maintained by us
SMW: OTTR Template definitions

the two separate templates are connected by instantiations which use the same entities

page name
SMW: Forms for OTTR Instantiations

Add/Change here OTTR instances for the generated/edited page. 
*?*: optional argument,  "!": not a blank node ([] or :example),  "DFLT": default value available  
Add "none" or "ottr:none" for optional arguments or for arguments that should be replaced by the default value. 

Add instances of the 'dpm:CompoundExample'-Template

- **Compound:** dpm:Fe48Rh52
- **Elements:**
  - (dpm:Fe, dpm:Rh)

(of type ottr:RI)  
(of type NEList<dpm:Element>)  
-- elements in ( ) separated by ; --

Add another instance
SMW: OTTR
Template Instances

- templates are instantiated inside the Wiki by usage of the SMW template mechanism
- instance page shows triples, debug information, and SMW annotations like categories and properties defined on the template page
- retrieve data inside the wiki with #Ask

```
{{#ask: 
  [[-Has subobject::+:]] [[Predicate::Emmo:hasPart]] [[Subject::Dpm:Fe48Rh52]]
  |?Subject
  |?Predicate
  |?Object
  |format=table}}
```
Conclusion & Open Questions

- benefits from already existing structured data
- improved communication with domain experts and improved understanding by domain experts
- workflows or naming conventions are important to relate templates to each other
- SMW as a central platform for the collaboration

Open for discussion:

- How to ensure correctness of the ontology?
- How to manage versioning as changes to template signatures require changes inside the templates and a revision of the relations this template is involved in? If instances are created manually, such changes are work intensive.
- How to design templates if the structure of the modelled processes is not yet fully explored?