Introduction to XD and XDP

Karl Hammar

2017-05-28
eXtreme Design

• "a family of methods and associated tools, based on the application, exploitation, and definition of Ontology Design Patterns (ODPs) for solving ontology development issues" – Presutti et al.
• Agile, iterative, pair development, testing emphasis
• Requirements written as user storys formalised as Competency Questions, Contextual Statements, Reasoning Requirements
• Tight customer integration
• Key steps: find ODP, instantiate ODP, integrate solution
Project initiation and scoping
Identify CODP catalogues
Collect requirement stories

Select story
Elicit requirements
Select set
Match and select ODPs
Instantiate and integrate ODPs

Test module
All req:s covered?
No
Yes
Release module
Release new version
Integrate partial solutions, evaluate, revise

All req:s covered?
No
All stories covered?
No
Yes
XD for WebProtégé (XDP)

Fork of WebProtégé including tooling to support some XD steps:
• Find ODPs
• Instantiate ODPs (template-based or specialisation-based)
• Integrate ODPs into solution (basic alignment)

Also includes visualization, courtesy of code from the VisualDataWeb project and new UI tabs for advanced editing

Some restrictions of WebProtégé:
• No reasoning
• ODP namespaces cloned, not imported
Graphical representation

General description

<table>
<thead>
<tr>
<th>Name</th>
<th>Bag</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intent</td>
<td>To model bags of items (elements). The Bag is characterized by a collection that can have multiple copies of each object.</td>
</tr>
<tr>
<td>Solution description</td>
<td>The Bag is characterized by a collection that can have multiple copies of each object. This is performed through the Item entity. The Item is linking exactly one resource through the relationship ItemContent.</td>
</tr>
</tbody>
</table>
Select the appropriate Content Ontology Design Pattern instantiation method from the choices below. For a discussion on their respective attributes and effects, see [http://goo.gl/dv8pA3](http://goo.gl/dv8pA3)

### Template-Based Instantiation

In this method the CODP building block is treated as a template that is instantiated into the target ontology module by way of copying and renaming its constituent classes and properties. Advantages of this method include that CODP-level generic concepts that may be off-putting to less experienced modellers are not included in the final ontology, but only the CODP structure is kept. Disadvantages include that future alignment to other ontologies using the same CODPs may be complicated, as the IRIs of COPD-level concepts are not kept.

### Import-Based Instantiation

In this method the original CODP is imported into the target ontology module, and instantiation is performed via specialization of CODP classes and properties using subsumption axioms. Advantages of this method include increased traceability and ease of alignment with other CODPs, as IRIs of CODP-level concepts are maintained.
Please provide labels for the ODP entities below that make sense when adapting the ODP to your domain.

<table>
<thead>
<tr>
<th>Classes</th>
<th>===&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>item</td>
<td>My item class</td>
</tr>
<tr>
<td>(collections) Bag</td>
<td>My bag class</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Object Properties</th>
<th>===&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>item content</td>
<td>my item has some content</td>
</tr>
<tr>
<td>item of</td>
<td>is item in my bag</td>
</tr>
<tr>
<td>has item</td>
<td>my bag has my item</td>
</tr>
</tbody>
</table>
Prefix: owl: <http://www.w3.org/2002/07/owl#>
Prefix: rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
Prefix: xsd: <http://www.w3.org/2001/XMLSchema#>
Prefix: rdfs: <http://www.w3.org/2000/01/rdf-schema#>

Ontology: <wptmp:entity>

ObjectProperty: <wptmp:entity#is item in my bag>
   Domain:
      <wptmp:entity#My item class>
   Range:
      <wptmp:entity#My bag class>

ObjectProperty: <wptmp:entity#my item has some content>
   Domain:
      <wptmp:entity#My item class>

ObjectProperty: <wptmp:entity#my bag has my item>
   Domain:
      <wptmp:entity#My bag class>
Hands-On Session

- Using XDP and ODPs, construct an ontology covering a set of requirements and structuring a set of provided data, in the policing domain.
- Inspired by a real-world project and real-world data.
- Goal: try out and learn about the method, the tooling, and look at some ODPs.
CAVEATS

• ODP Portal, tooling, etc. are mirrored from the Internet, to account for the lack of Internet connectivity. Some IRIs you see here do not exist in the real world, or lead to content that is not the same as what you see here.

• Further: ODP quality varies greatly in the real world: stale IRIs, bad documentation, bad illustrations, dependency on remote references, etc.

• Thus: four specifically suggested patterns have been tampered with in order to simplify modelling. E.g., merged import closure, added missing documentation, added some common-sense assumptions.

• Finally: this is beta-quality research software. Expect some bugs.
Get started

Poll: Who wants some Google Refine/OpenRefine introduction as well?

• WiFi SSID “ODP Tutorial”, password: “eswc2017”
• Instructions: [http://ontologydesignpatterns.org/instructions.txt](http://ontologydesignpatterns.org/instructions.txt)
• Data: [http://ontologydesignpatterns.org/data.zip](http://ontologydesignpatterns.org/data.zip)
• XDP Instance: [http://wp.xd-protege.com](http://wp.xd-protege.com)
• ODP Portal: [http://ontologydesignpatterns.org](http://ontologydesignpatterns.org)
• WebVOWL Instance: [http://vis.xd-protege.com](http://vis.xd-protege.com)