Reasonable Ontology Templates

Martin G. Skjæveland[1] Henrik Forssell[1] Johan W. Klüwer[2] Daniel P. Lupp[1] Evgenij Thorstensen[1] Arild Waaler[1]
[1]Department of Informatics, University of Oslo, [2]DNV GL, Norway

Overview

- Modular macros for OWL/RDF — in OWL/RDF
  - Good for ontology construction and maintenance
  - Clear interface, consistent pattern use, hide complexity
  - Reasoning support
  - DRY — don't repeat yourself
- Leverage W3C stack and tools
- Implementation, OWL vocabulary, and template library available at `http://ottr.xyz`

Preliminaries

- An *(ontology) template* has a head and a body:

$$\underbrace{\underbrace{\mathcal{T}}_{\text{name}}\ \underbrace{(p_1, \ldots, p_n)}_{\text{parameters}}}_{\text{head}} :: \underbrace{\mathcal{O}}_{\text{body}}$$

- A *template instance* $\mathcal{T}(a_1, \ldots, a_n)$ specifies a substitution of the template's parameters in the body with the instance's arguments.
- The template body is a regular ontology that can contain template instances and where parameters may be used as placeholders for concepts, roles, individuals and data values.
- A template instance is *expanded* by recursively replacing template instances with substituted template bodies.

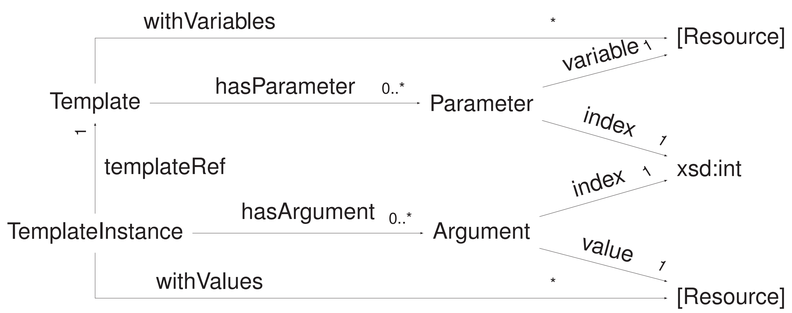
Extensible Framework

$$\mathcal{T}(p_1, \ldots, p_n) \underset{\text{lowering}}{\overset{\text{lifting}}{\rightleftarrows}} \mathcal{O}$$

- Bulk data formats: XLSx, XML (XSD + SAWSDL), RDF, OWL
- Bulk lifting and lowering transformations: XSLT, SPARQL

UiO : University of Oslo

OTTR OWL Vocabulary: `http://ns.ottr.xyz`



Examples

- Template $PartOf(Whole, Part) :: \{ Whole \sqsubseteq \exists\, hasPart.Part \}$ serialised with OTTR vocabulary:

```
 ### head:
<http://draft.ottr.xyz/i17/partof> a ottr:Template ;
  ottr:hasParameter [ ottr:index 1; ottr:variable :Whole ] ,
                    [ ottr:index 2; ottr:variable :Part  ] .
    ## or: ottr:withVariables ( :Whole :Part ) .
  ### body:
:Part  a owl:Class .
:Whole a owl:Class ;
  rdfs:subClassOf [ a owl:Restriction ;
    owl:onProperty ex:hasPart ; owl:someValuesFrom :Part ] .
```

- Template instance $PartOf(Hammer, Handle)$ serialised with OTTR vocabulary:

```
[] ottr:templateRef <http://draft.ottr.xyz/i17/partof> ;
   ottr:hasArgument [ ottr:index 1; ottr:value :Hammer ] ,
                    [ ottr:index 2; ottr:value :Handle ] .
    ## or: ottr:withValues ( ex:Hammer ex:Handle ) .
```

- . . . expands to

```
:Handle  a owl:Class .
:Hammer a owl:Class ;
  rdfs:subClassOf [ a owl:Restriction ;
   owl:onProperty ex:hasPart; owl:someValuesFrom :Handle ] .
```

Examples

- Terse input and iterations with lists + use of 3 "official" OTTRs:

```
<http://draft.ottr.xyz/pizza/NamedPizza> a ottr:Template ;
  ottr:withVariables ( :pizza ( :toppings ) )  .
  ### body:
:pizza rdfs:subClassOf p:NamedPizza .
[] ottr:templateRef ottr-owl:SubObjectSomeValuesFrom ;
   ottr:hasArgument [ ottr:index 1; ottr:value :pizza ] ,
            [ ottr:index 2; ottr:value p:hasTopping ] ,
            [ ottr:index 3; ottr:eachValue ( :toppings ) ] .
[] ottr:templateRef ottr-owl:SubObjectAllValuesFrom ;
   ottr:withValues ( :pizza p:hasTopping _:alltoppings ) .
[] ottr:templateRef ottr-owl:ObjectUnionOf ;
   ottr:withValues ( _:alltoppings ( :toppings ) ) .
```

- The instance

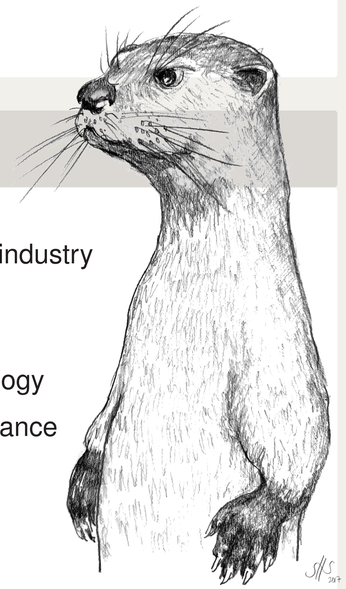
  $NamedPizza(Napoletana, \langle Tomato, Mozzarella, Olive, Caper, Anchovies \rangle)$

  . . . expands to:

  $Napoletana \sqsubseteq NamedPizza$
  $Napoletana \sqsubseteq \forall\, hasTopping.(Tomato \sqcup Mozzarella \sqcup \ldots)$
  $Napoletana \sqsubseteq \exists\, hasTopping.Tomato$
  $Napoletana \sqsubseteq \exists\, hasTopping.Mozzarella$
  $Napoletana \sqsubseteq \exists\, hasTopping. \ldots$

Future Work

- Large-scale evaluation
  —prototype successfully tested in industry
- Language design and extensions
- Support template pre-conditions
  —with also template head as ontology
- Methods for template library maintenance
  — with ontology interrelationships
- Protégé plugin
- Template-based visualisations



Poster by Martin G. Skjæveland and Daniel P. Lupp