

# The Template Instance Pattern

Csongor Nyulas, Tania Tudorache, Samson Tu

Stanford Center for Biomedical Informatics Research, Stanford University, US  
{nyulas, tudorache, swt}@stanford.edu

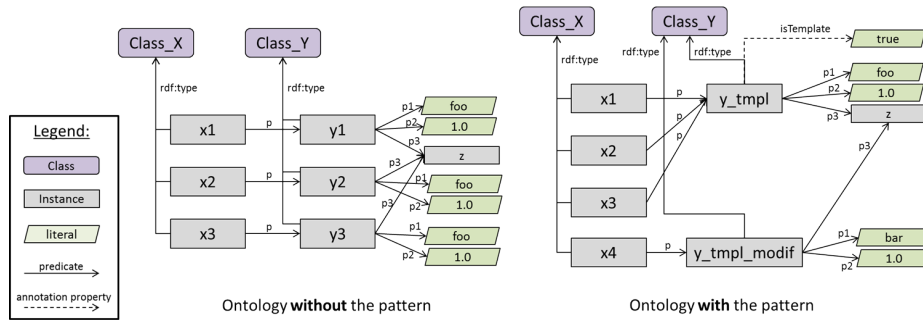
**Abstract.** We present the Template Instance Pattern, a content design pattern that marks an instance, which is used as a value for a property, to be a “template”. A template instance is intended to be immutable (none of its properties can be changed). If the content (i.e., any of the property values) of the template instance needs to be changed, a template flag will indicate to an ontology editor that it should create a clone of the template instance, and replace the value of the property with the desired value in the newly created clone. This pattern is especially useful for cases in which an ontology makes abundant use of reified relations (represented as instances), which are repetitive and that increase significantly the size of an ontology. We created this pattern as a result of building a large real world medical ontology that makes excessive use of reification.

## 1 Background

Ontologies make use of reified relations to model n-ary relations or describe additional properties of a relation (e.g., confidence level, provenance, and so on). For example, biomedical ontologies often need to qualify facts about the domain with scientific evidence: a definition of a disease would need to include links to scientific papers that sourced or endorsed that definition, or relationships between different entities need to carry a probability. Some ontologies, such as the ICD-11 [2, 1] make extensive use of reified relations. Practically, all relations in the ICD-11 ontology are reified. One issue with reification is that it creates the intermediate reified instances that do not contribute any real content, but they rather provide a structure that groups together all properties of a reified relation. An ontology that makes abundant use of reified relationships is likely to be very large in size due to the “clutter” introduced by the reified instances. A larger ontology may be harder to maintain, and may create challenges for ontology editors or reasoners. The template instance pattern proposes a way of reducing the number of reified instances and the related property assertion axioms in an ontology, especially for the cases in which the reified relations are identical for multiple entities.

## 2 Solution Description

The pattern proposes to use the same “template” instance as the value of a reified property for multiple subjects (rather than having multiple copies of the same reified instance). We give an example in Figure 1 to make it easier to understand. The individuals  $x_1$ ,  $x_2$  and  $x_3$  have a reified property  $p$  that has as values the reified individuals,  $y_1$ ,  $y_2$  and  $y_3$ , respectively. In OWL, we would have the following object property assertions:



**Fig. 1.** An abstract example of reified modeling without (left side) and with (right side) the Template Instance Pattern.

```
(x1 p y1)
(x2 p y2)
(x3 p y3)
```

The properties for the reified individuals,  $y_1$ ,  $y_2$  and  $y_3$  are all identical (the underscore represents the index for  $y$ : 1, 2, or 3):

```
(y_ p1 foo)
(y_ p2 1.0)
(y_ p3 z)
```

The pattern proposes to create a template instance,  $y\_tmpl$  that has the common property values (right hand side of Figure 1):

```
(y_tmpl p1 foo)
(y_tmpl p2 1.0)
(y_tmpl p3 z)
```

The template instance would be used as the value for the reified property for  $x_1$ ,  $x_2$  and  $x_3$ :

```
(x1 p y_tmpl)
(x2 p y_tmpl)
(x3 p y_tmpl)
```

In addition, we will also add one annotation property, `isTemplate:true`, on  $y\_tmpl$  to mark that it as a template instance. The intention is that the template instance is immutable, i.e., the property values of the template instance cannot be changed.

In the case that a user would like to change a property value of the reified instance, a clone of the template instance would be created and the change would occur on the clone. An ontology editor would use the `isTemplate` annotation property to check that a certain instance cannot be modified, and it should rather create a clone.

Figure 1 gives an example of this situation. Say that at a given time  $t_0$ , there was an additional  $x_4$  individual that had the template instance as the value for  $p$ . At  $t_0$ :

Linearization	Is part of?	Is grouping?	Linearization Parent	
01 Morbidity	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	12 XII Diseases of the skin	
02 Mortality	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	12 XII Diseases of the skin	
03.1 Primary Care - High Res. Set.	<input type="checkbox"/>	<input checked="" type="checkbox"/>	12 XII Diseases of the skin	
03.2 Primary Care - Low Res. Set.	<input type="checkbox"/>	<input checked="" type="checkbox"/>	12 XII Diseases of the skin	
04 Research	<input type="checkbox"/>	<input checked="" type="checkbox"/>	12 XII Diseases of the skin	
05 Spec. Adapt: Mental Health	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Click here to select a parent	
06 Spec. Adapt: Dermatology	<input checked="" type="checkbox"/>	<input type="checkbox"/>	12 XII Diseases of the skin	
07 Spec. Adapt: Musculoskeletal	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Click here to select a parent	
08 Spec. Adapt: Neurology	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Click here to select a parent	
09 Spec. Adapt: Paediatrics	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	12 XII Diseases of the skin	
10 Spec. Adapt: Occupational Health	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Click here to select a parent	

**Fig. 2.** Proposed usage of the Template Instance Pattern for representing views in the ICD-11 ontology. Each row in the table represents a reified instance (which represents a view), and each column represents a property value of the reified instance. Each class in the ontology can be part of one or several views.

```
(x4 p y_tmpl)
(y_tmpl p1 foo)
(y_tmpl p2 1.0)
(y_tmpl p3 z)
```

Later, at time  $t_1$ , the user decides to change the value  $p_1$  from  $foo$  to  $bar$ . The ontology editor (or the user, if there is no support in the tool), will see the `isTemplate` annotation property on `y_tmpl` as set on `true`, and it will create a clone of it, `y_tmpl_modif`, by cloning also all the object and data property axioms. Then, it will change the value of  $p_1$  on `y_tmpl_modif`. At  $t_1$ :

```
(x4 p y_tmpl_modif)
(y_tmpl_modif p1 bar)
(y_tmpl_modif p2 1.0)
(y_tmpl_modif p3 z)
```

As a result of using the pattern, we can reduce the number of data and object property axioms in the ontology. This “deflation” of the ontology is especially significant, if there are many repetitive values in the ontology, with few changes, and if the number of properties of the reified instance is large.

### 3 Example of Usage

As we have mentioned before, the ICD-11 ontology that describes diseases and their properties makes extensive use of reification. A class in the ICD-11 ontology represents a disease, which has several properties (e.g., title, definition, synonyms, signs and symptoms, etiology, manifestation, etc.) that are encoded as property axioms. Each disease class can participate in one or several “views” that will be extracted from the ICD-11

ontology. A view (in ICD language called “linearization”) is a portion of the ontology that is relevant for a particular sub-domain or use case (e.g. Morbidity, Mortality, Primary Care, Dermatology, etc.). Figure 2 shows an example of these views. For each disease class, the specifications of the views it participates in, are represented as reified instances (the table rows in Figure 2). Each of the view specification instances contains five additional properties (partially shown as columns in the table). Each disease class has to specify their inclusion in ten different views. Each view is represented as a reified instance. The ontology currently contains over 40.000 disease classes, each having 10 view specifications, and each of the view specifications containing additional 5 properties. Currently, we have over 400.000 reified view specifications, and over 2 million property assertion describing them. Many of these reified view specification instances are identical. We plan to use the Template Instance Pattern to reduce significantly the number of the reified instances in the ontology. For example, for the initial modeling (that contains no user modifications), we would have only 10 reified view specification instances (reduced from 400.000) and 400.000 property assertions (reduced from 2 million). This “deflation” of the ontology size is significant, would allow us to more easily maintain the view specifications, and would have a beneficial impact on the performance of the ontology tools used for editing ICD-11.

## 4 Conclusion

We presented the template instance pattern that was created from the need to address scalability and maintainability issues in the development of the ICD-11 ontology. The pattern proposes the use of template instances as values for reified properties, and their flagging using an annotation property that can be used to provide support for this pattern in ontology tools. We also described an example of the pattern and its operationalization in a generic ontology tool.

## References

1. T. Tudorache, S. Falconer, C. Nyulas, N. Noy, and M. Musen. Will Semantic Web Technologies Work for the Development of ICD-11? In *The 9th Intl. Semantic Web Conference (ISWC 2010)*, pages 257–272. Springer, 2010.
2. World Health Organization. The 11th Revision of the International Classification of Diseases (ICD-11). <http://www.who.int/classifications/icd/revision/en/index.html>. Last accessed: August, 2012.