

1 Ontology to Describe the Input and Output Data of the Method for Building Domain Ontologies from User-Generated Classification Systems

In this document we describe the ontology (see figure 1) designed to model the input and output data of the processes, activities and tasks making up the method for building domain ontologies from user-generated classification systems. This ontology, developed in OWL-DL ¹, was designed by reusing existing ontologies, and following ontology design best practices defined by the World Wide Web Consortium W3C such as the part-whole best practices which are described later in this section.

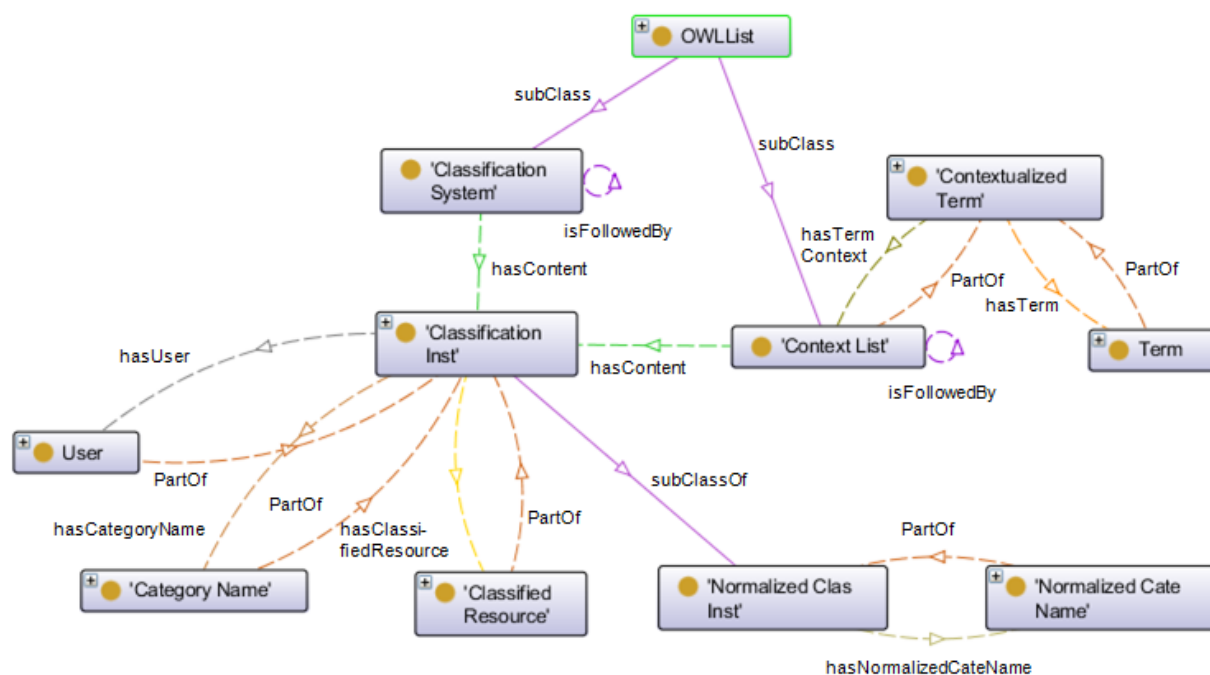


Figure 1: Ontology modeling the data structures presented in the method.

The classification system is represented by the class *ClassificationSystem* which is modeled as a list of classification instances, which in turn are represented by the class *ClassificationInst*. To model a list we reuse the ontology OWLlist ² which has been proposed in [?]. In this ontology a list is a sequence of nodes where each node contains data and a link to the next node in the list. Thus to model the list a node class is created as a subclass of OWLlist, and then the *hasContent* and *isFollowedBy* object properties have to be defined. While the *hasContent* relates the node with the class containing the data, the *isFollowedBy* defines the class of the next node in the list. Therefore in the case of the *ClassificationSystem* class we defined it as *subClassOf* OWLlist and set the *hasContent* property to a *ClassificationInstance* class and the *isFollowedBy* property

¹OWL specification is available at: <http://www.w3.org/TR/owl-guide/>

²OWLlist is available in <http://www.co-ode.org/ontologies/lists/2008/09/11/list.owl>

to a `ClassificationSystem` class.

A classification instance consists of a user, a category name and classified resource which are represented in the ontology by the corresponding classes. The *consists of* relation has been modeled in two ways. first we have created the object properties *hasUser*, *hasCategoryName* and *hasClassifiedResource* to specify the classification instance parts. On the other hand, we have modeled the *partOf* relation between the constituent parts and the classification instance following the best practices in this respect provided by the W3C³. One of the best practices recommends to define the *partOf* as a taxonomy. To do so, first the *partOf* is defined as an object property. Then each of the constituent parts has to be declared as subclass of a restriction over the range of values that the *PartOf* can take, which in this case is `ClassificationInst`. As we require that all users, category names and classified resources are part of at least one classification instance, we create the restriction using an existential quantifier. In addition we require that users, category names, and classified resources only can be part of classification instances, and hence we create another restriction using an universal quantifier. All the *partOf* relations presented in this ontology follows the recommend best practice and use both, existential and universal quantifiers when defining the restriction over the range of the *partOf* object property.

A normalized classification instance, represented by the class *NormalizedClasInst*, is a classification instance plus a normalized category name, which is represented by the class *NormalizedCateName*. Therefore the `NormalizedClasInst` class is subclass of `ClassificationInst`, and is related to the class `NormalizedCateName` by means of the object property *hasNormalizedCateName*. In addition, the `normalizedCateName` class is defined as *partOf* of the `NormalizedClasInst` class.

A contextualized term, represented by the class *ContextualizedTerm*, consists of a term and its context, which in turn are represented by the class *Term* and *ContextList*. The *consists of* relation has been modeled using the object properties *hasTerm* and *hasTermContext*. On the other hand, we have defined a *partOf* relation between the constituent parts `Term` and `ContextList`, and the `ContextualizedTerm` class. The `ContextList` is a list of classification instances. Therefore it is defined a subclass of `OWLlist`. its *hasContent* and *isFollowedBy* object properties are set to the class `ClassificationInstance` and to the class `ContextList` respectively. Finally, we have included a standalone *TransformationInstance* class to represent the instances which has been transformed in the Data Preprocessing activity of the method. Note that is not possible to define in advance the semantics of this class since it depends on the data structure which is defined during the execution of the method. And therefore the ontology engineer must adapt this ontology according to the circumstances defined by the method execution.

In the following we present the definitions of the the different classes and object properties making up the ontology.

³Part-whole best practices published in : <http://www.w3.org/2001/sw/BestPractices/OEP/SimplePartWhole/simple-part-whole-relations-v1.3.html>

2 Classes

OWLList

This class is imported from the ontology list.owl and it represents a lists. The list is defined by specifying the content of each node in the list (hasContent object property) and a link to the next node in the list (isFollowedBy object property).

ClassificationSystem

This class represents the classification systems as a list of classification instances. It is defined as:

ClassificationSystem \sqsubseteq OWLList
ClassificationSystem $\sqsubseteq \forall$ hasContents ClassificationInst
ClassificationSystem $\sqsubseteq \forall$ isFollowedBy ClassificationSystem

ClassificationInst

This class represents a classification instance which is an individual classification of a resource in the classification system. That is, the relation between a user who has used a category name to classify a resource.

User

This class represents a user in the classification system. A user is, at least and only, part of a classification instance. In addition users, category names, and classified resources are disjoint. It is defined as:

User $\sqsubseteq \forall$ partOf ClassificationInst
User $\sqsubseteq \exists$ partOf ClassificationInst
User $\sqsubseteq \neg$ CategoryName
User $\sqsubseteq \neg$ ClassifiedResource

ClassifiedResource

This class represents a classified resource (e.g., Web pages, pictures, and users) that users classify in the system. These resources are, at least and only, part of a classification instance. In addition, classified resources, category names and users are disjoint. It is defined as:

ClassifiedResource $\sqsubseteq \forall$ partOf ClassificationInst
ClassifiedResource $\sqsubseteq \exists$ partOf ClassificationInst
ClassifiedResource $\sqsubseteq \neg$ CategoryName
ClassifiedResource $\sqsubseteq \neg$ User

CategoryName

This class represents the category names (e.g., tags or list names) that users assign to classify resources in the system. These category names are, at least and only, part of a classification instance. In addition, category names, classified resources, and users are disjoint. It is defined as:

CategoryName $\sqsubseteq \forall$ partOf ClassificationInst
CategoryName $\sqsubseteq \exists$ partOf ClassificationInst
CategoryName $\sqsubseteq \neg$ ClassifiedResource
CategoryName $\sqsubseteq \neg$ User

NormalizedClasInst

This class represents the normalized version of a classification instance, where the category name has been turned into a normalized category name. This class is subclass of the ClassificationInst class. It is defined as:

NormalizedClasInst \sqsubseteq ClassificationInst

NormalizedCateName

This class represents the normalized version of a category name. A normalized category name is, at least and only, part of a normalized classification instance. It is defined as:

NormalizedCateName $\sqsubseteq \forall$ partOf NormalizedClasInst
NormalizedCateName $\sqsubseteq \exists$ partOf NormalizedClasInst

ContextualizedTerm

This class represents a contextualized term. That is a given term and the list of classification instances where this term appears as a category name.

Term

This class represents a term. A term corresponds to a normalized category name which has been extracted from the classification system. A term is, at least and only, part of a contextualized term. It is defined as:

Term $\sqsubseteq \forall$ partOf ContextualizedTerm
Term $\sqsubseteq \exists$ partOf ContextualizedTerm

ContextList

This class represents the context of a given term. The context is defined as a list of classification instances where the term was used as a Category Name. The list is defined by specifying the hasContents object property as classification instances, and the isFollowedBy as another ContextList. In addition, a context list is, at least and only, part of a contextualized term. It is defined as:

ContextList \sqsubseteq OWLList
ContextList $\sqsubseteq \forall$ hasContents ClassificationInst
ContextList $\sqsubseteq \forall$ isFollowedBy ContextList
ContextList $\sqsubseteq \exists$ partOf ContextualizedTerm
ContextList $\sqsubseteq \forall$ partOf ContextualizedTerm

TransformationInstance

This class represents the different ways that a normalized classification instance can be turned into during the preprocessing activity. According to the defined transformation at execution time this class semantics has to be defined by means of its association with existing or new classes.

3 Object properties

hasContents

This object property defines the content of each node in the list. Its semantics is defined in the list.owl ontology.

isFollowedBy

This object property defines the class of the next node in the list. Its semantics is defined in the list.owl ontology.

partOf

This object property defines a part of relation. Its inverse relation is *hasPart*. It is defined as:

$$\text{partOf} \equiv \text{hasPart}^{-}$$

hasPart

This object property defines a has part relation. Its inverse relation is *partOf*. It is defined as:

$$\text{partOf} \equiv \text{hasPart}^{-}$$

hasUser

This object property defines the relation has user which is stated always between a classification instance (domain) and a user (range). It is defined as:

$$\begin{aligned} \exists \text{ hasUser } \text{Thing} &\sqsubseteq \text{ClassificationInst} \\ \top &\sqsubseteq \forall \text{ hasUser } \text{User} \end{aligned}$$

hasCategoryName

This object property defines the relation has category name which is stated always between a classification instance (domain) and a category name (range). It is defined as:

$$\begin{aligned} \exists \text{ hasCategoryName } \text{Thing} &\sqsubseteq \text{ClassificationInst} \\ \top &\sqsubseteq \forall \text{ hasCategoryName } \text{CategoryName} \end{aligned}$$

hasClassifiedResource

This object property defines the relation has classified resource which is stated always between a classification instance (domain) and a classified resource(range). It is defined as:

$$\begin{aligned} \exists \text{ hasClassifiedResource } \text{Thing} &\sqsubseteq \text{ClassificationInst} \\ \top &\sqsubseteq \forall \text{ hasClassifiedResource } \text{ClassifiedResource} \end{aligned}$$

hasNormalizedCateName

This object property defines the relation has normalized category name which is stated always between a normalized classification instance (domain) and a normalized category name (range). It is defined as:

$$\begin{aligned} \exists \text{ hasNormalizedCateName } \text{Thing} &\sqsubseteq \text{NormalizedClasInst} \\ \top &\sqsubseteq \forall \text{ hasNormalizedCateName } \text{NormalizedCateName} \end{aligned}$$

hasTerm

This object property defines the relation has term which is stated always between a contextualized term (domain) and a term (range). It is defined as:

$$\begin{aligned} \exists \text{ hasTerm } \text{Thing} &\sqsubseteq \text{ContextualizedTerm} \\ \top &\sqsubseteq \forall \text{ hasTerm } \text{Term} \end{aligned}$$

hasTermContext

This object property defines the relation has context term which is stated always between a contextualized term (domain) and a context list (range). It is defined as:

$$\begin{aligned} \exists \text{ hasTermContext } \text{Thing} &\sqsubseteq \text{ContextualizedTerm} \\ \top &\sqsubseteq \forall \text{ hasTermContext } \text{ContextList} \end{aligned}$$