

# Property Clustering in Semantic MediaWiki

## Define Your Own Classes and Relationships

Dr. Gero Scholz

IVU Traffic Technologies AG, Berlin, Germany

**Abstract.** Semantic MediaWiki (SMW) currently has an atomic understanding of properties: they are seen as annotation marks which can be arbitrarily attached to articles. As a next step towards an object oriented representation of knowledge we introduce a concept of *property clustering*. This makes it possible to define a formal meta model for a knowledge domain. We support class inheritance and typed relations between objects. As a proof of concept we provide an implementation which is based on a set of templates and a few existing MediaWiki extensions. A graph of the meta model can be generated automatically. We offer different models for entering information based on templates and forms. A demo website (<http://semeb.com/dp1demo/SMWpc>) is available.

Keywords: *Semantic MediaWiki, Semantic Forms, Class, Relation, Inheritance, Meta Model, Proof of Concept, Demo Implementation*

## 1 Introduction

Currently in SMW every possible combination of properties can be assigned to every article. It is possible to assign multiple values for the same property to the same article. The difference between *relations* and *values* which was part of the SMW concept in older versions has been dropped in favor of more generalized *properties* in the latest SMW release. All this leads to a fairly universal, generic concept. In short, SMW offers a concept of *weak typing* expressed by arbitrary bundles of properties taken from an ocean of all possible attributes which might be useful for annotation.

But people do not primarily perceive objects as conglomerates of attributes. Instead they classify objects and use well defined names for these classifications. Classes in essence are *named clusters of properties*. Consequently, this article introduces a concept of *strong typing* which we call SMWpc. The 'pc' might translate to 'property clustering' or to 'personal classes'. The latter interpretation would emphasize that the design of classes always depends on the perspective of authors and readers.

SMWpc is a proof of concept which is already usable for small wikis. It is based on SMW, a few other MediaWiki extensions and some tricky MW templates. To improve performance and robustness a more professional implementation should be made by extending the current php source code of SMW.

## 2 Idea and Concept

The graph in Fig.1 describes the general idea of SMWpc.

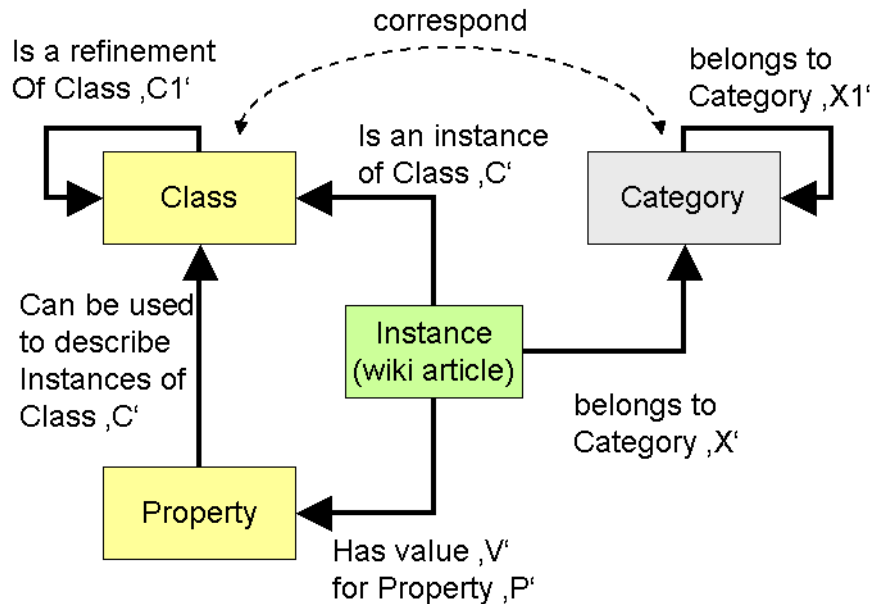


Fig. 1. Meta Model of SMWpc

In SMWpc MediaWiki articles are seen as *instances of a classes (objects)*. A class is formally described in a meta model using special *meta properties*. Each class in SMWpc corresponds to a traditional MediaWiki category which is named after the class. There are no freely floating properties in SMWpc. Instead properties are always tied to classes. A special meta property is used to describe class inheritance.

## 3 Meta Model

A full version of the meta model can be found on the website. The most important meta property is *.obj is a*. It states that an article describes an object of a certain class. The meta property *.prop describes* ties a property to its class. Note that one property can be tied to many classes. *.class extends* is used to define (single) inheritance. It is a good design principle to use templates for the assignment of property values. Via *.prop assigned by* we establish a reference between

a property and its associated 'assignment template'. Sometimes the value of a property can be algorithmically derived from the values of one or more other properties. We use *.prop derived from* to express this. The meta property *.prop refers to* allows to express that a property of a class is to be understood as a reference to an object of another class. *.prop reverse* offers a second name for the same relation if used in the opposite direction. The properties *.prop unique* and *.prop mandatory* express the cardinality of properties, i.e. they state if zero, one or many values will be allowed for a certain property.

Apart from these essential features there are other meta properties which can help you to attach color schemes or icons to classes and properties. There is also a meta property that links an edit form to a class. And last not least there are class-specific templates which produce a nice common layout for all objects belonging to the same class.

As you may have noted all SMWpc meta properties start with a prefix like *.obj*, *.class*, *.prop*, .. to make clear that they do not belong to the application domain of the wiki. It would be a good idea to use the same convention for SMWs existing meta properties like 'has type', 'allows value' etc. There should be a clear separation of namespaces between the meta model and the application domain of a wiki. Technically speaking all SMWpc meta properties are normal SMW properties. This allows to use the concept of reflection (introspection) in the implementation of SMWpc. SMW should consider to follow the same strategy. It would be of great value to operate on the *information model* of a wiki in the same query language that you use to operate on its *contents*.

## 4 Focus of SMWpc

The initial version of a wiki typically contains a small, weakly structured collection of articles which have some commonalities. Once a wiki grows the designer of the wiki can use SMWpc to create a formal meta model which supports queries and helps to enter information in a more structured way. It is important to closely monitor the ratio between the size of the 'information model' and the total amount of information in a wiki. Encyclopedic wikis will have a lower ratio than specialized wikis with closer scope and more elaborated relationships between the articles. Most often there will be a perceived lack of semantic structure in a wiki. But there is also a (small) danger of over-engineering when a small wiki is started with a very rigid structure.

The main focus of SMWpc is on small and medium-size wikis (less than 10.000 pages) which have a dedicated focus. Their user communities agree on a common scheme for classification of articles and they want better support for collecting highly structured information. An example could be a wiki in the area of molecular genetics but it could also be a wiki about pets where you have classes like *species*, *food*, *disease* etc. It is quite clear that a property named *symptom* belongs to class *disease* and not to *food* or *species*. With SMWpc there is a way to express this. While it may make a lot of sense to have multiple values for the symptoms of a disease, there should only be a single value for the property

*maximum age* of class *species*. The property *likes* must contain a reference to an instance of *food* and not to a *disease*. With SMWpc you can express all this and much more.

## 5 Example

We set up an example which deals with students, their subjects of study and their hobbies (playing games and playing musical instruments). The example tries to demonstrate all features of SMWpc. So do not pay too much attention to the contents. The information model generated by SMWpc is shown in Fig.2. For more information please go to the website (<http://semeb.com/dp1demo/ClassStudent>).

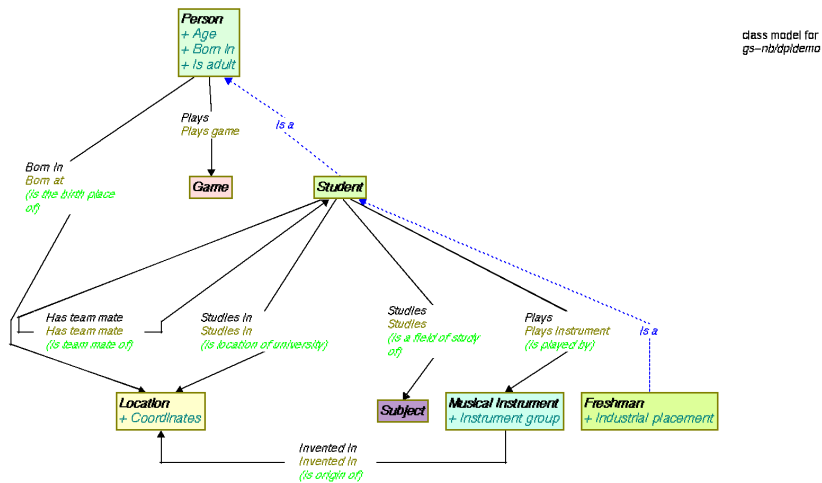


Fig. 2. Sample class model of an SMWpc application

## 6 Conclusion

SMWpc is a first step in the direction of true object oriented semantic modeling with MediaWiki. There are lots of features which can be improved and added in future. And there is much more functionality already available than could be shown and explained here. We hope that the idea of SMWpc will be adopted by the Semantic MediaWiki community. Integration of SMWpc concepts into SMW would create a more robust solution with better performance. Adding SMWpc concepts to SMW would enlarge the scope of SMW significantly. It would be a pure add-on, so no current functionality would be lost.