DCMI ABSTRACT MODEL ANALYSIS: RESOURCE MODEL

Vicente Palacios
Universidad Carlos III de Madrid
Tel: +34 916249043
vicente.palacios@uc3m.es

Jorge Morato
Universidad Carlos III de Madrid
Tel: +34 916248848
jorgeluis.morato@uc3m.es

Juan B. Llorens
Universidad Carlos III de Madrid
Tel: +34 916249498
juan.llorens@uc3m.es

Jose A. Moreiro
Universidad Carlos III de Madrid
Tel: +34 916249238
joseantonio.moreiro@uc3m.es

Abstract:

The importance that DCMI abstract model is acquiring as a metadata model scheme reference has motivated the work of the present analysis. This research has as an objective to make easy its understanding and contribute, from our experience in the use of UML model language revision proposals that improve the notation and representation of some concepts included in the resource model that is included in DCAM.

Finally, the model in UML is included, incorporating the recommendations on notation and representations of the concepts of the original model.

Keywords:
DCMI Abstract Model, UML, Resource Model, Abstract Model Analysis.

1. Introduction

The DCMI Abstract Model (Dublin Core Metadata Initiative Abstract Model: DCAM) (DCMI, 2005) specifies an abstract model for metadata schemes, initially, for the Dublin Core scheme. The objective is to provide a reference model in which distinct scheme codifications could be compared at the same time that the understanding of the concepts that conforms the scheme and the interoperation with other codification are provided.

Since its approval, the Abstract model is becoming a model in which other metadata schemes, i.e. IEEE LOM, try to correspond their elements with the objective of improving the interoperation with Dublin Core.

The importance that the DCMI abstract model is acquiring as a metadata scheme model reference has motivated the work of the present analysis, which has as an objective to make easy its understanding and contribute, from our experience in the use of UML model

https://doi.org/10.23106/dcmi.952108542

language (Unified Modelling Language) (UML, 2003), revision proposals that improve the notation and representation of some of the concepts included in the DCAM.
More concretely, the main objectives of the present analysis are:

To analyse those concepts of abstract resource model that present difficulty in their interpretation. In this case, alternative representations and textual definitions shall be proposed, on the basis of the interpretation of the originals included in the DCAM.

The revision of the notation used in the graphic model representation. In concrete, the UML modelling elements in its version 1.5.

To adequate it to UML standard, the graphic representations included has been developed using modelling CASE tool umlCAKE (Reuse Company, 2006).

2. Analysis of Resource model

In this section, identified aspects, susceptible to revision are included. To make easy the identification of the concepts in the original model, the examples are shown using the names assigned in the DCMI abstract model. At the end of this section, a complete model shall be shown.
Firstly, the model represents the relation between the resource concept and the class concept as shown in Figure 1:

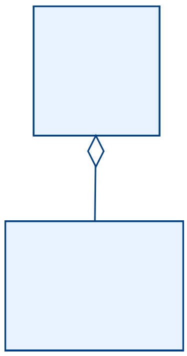


Figure 1: Graphic representation of the relationship between class and resource.

However, the textual definition of the relation indicates that *"Each resource could be a part of one or more classes"*, in other words, the contrary of what is shown in the previous graphic representation. In addition, the multiplicity expressed in the textual description *"one or more classes"* is more restrictive than that which is shown graphically "zero or more classes".
Interpreting the textual definition of the aggregate as correct and using a less restrictive multiplicity, due to the fact that a resource may not belong to any class, the correct graphic representation could be demonstrated in the following figure:

https://doi.org/10.23106/dcmi.952108542

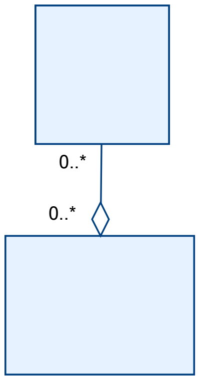


Figure 2: Graphic representation, the relation between class and resource proposal.

reso

As can be seen, the multiplicity has been included in the extreme of the aggregate corresponding to class composition. It has been established as "*0..**" due to the fact that, in the first place, a resource could belong to none or to more than one class. In this way, it would be necessary to define the relation between the concepts as: *"Each resource could be a part of zero or more classes and a class could aggregate zero or more resources"*.
Continuing with the relation in the resource concept, the aggregate with the concept property/value pair is represented graphically as shown in the following figure:

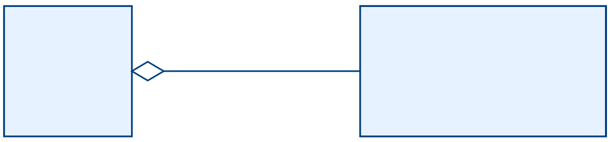
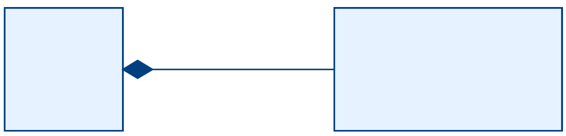


Figure 3: Graphic representation of the relation between resource and property/value pair

cla

In this case, the multiplicity of the extreme corresponding to the resource is neither graphically nor textually specified.
Accepting the relation between both concepts of aggregate as corrects and for those readers who are not very familiar with UML, it is necessary to establish if the composition restriction as exactly verified in the following, has been carried out.
Interpreting that property/value pair forms a part of a resource and is not shared by various resources and including the fact that a property/value pair does not make sense if it is not forming a part of a resource, it is verified how the restrictions that transform an aggregate into composition is complied. It is necessary to understand both restrictions, therefore, in the first, it has to be understood that even though two exactly equal property/value pair exist, both instances are distinct; in the second, it has to be understood that the property/value pair does not make sense if it is not forming a part of the resource it composes.
As a result, the relation should be represented as:



https://doi.org/10.23106/dcmi.952108542

resource

Figure 4: Graphic representation, proposal of the relation between resource and property/value pair

To make easy the interpretation of the relation, the multiplicity in the extreme of the resource composition has been specified as *"1..1"*. In this case, it could have been denoted in a short form *"1"* or, even left out (though the latter option is not recommended).
Once again, it would be necessary to update the textual definition as: *"Each resource is composed of zero or more property/pair value"*.
Going on with the analysis, the property/value pair aggregate are represented in the model as:

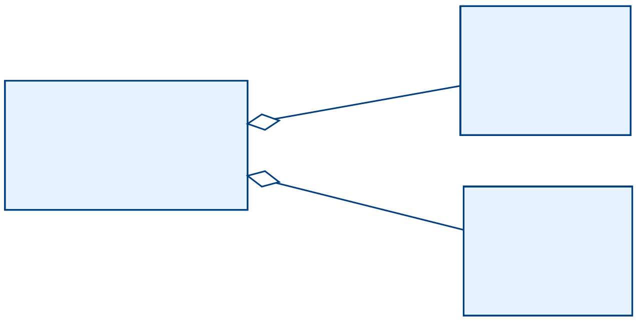


Figure 5: Graphic representation of the relation between property/value pair, value and property

In both aggregate, it would be necessary to verify if the extra restriction which converts them into compositions is complied. As much as for the value and as for the property, it complies that they only make sense forming a part of the aggregate property/value.
In respect to the "exclusivity" restriction of the aggregate, it could seem that a value aggregates more than one property/value pair, however, this affirmation is not certain if we bear in mind that the value is defined as many times as it forms part of the property/value pair. In other words, neither the value nor the property is shared by various property/value pair.
Bearing in mind the previous recommendation about the multiplicity restriction, the proposed graphic representation has been included in Figure 6:

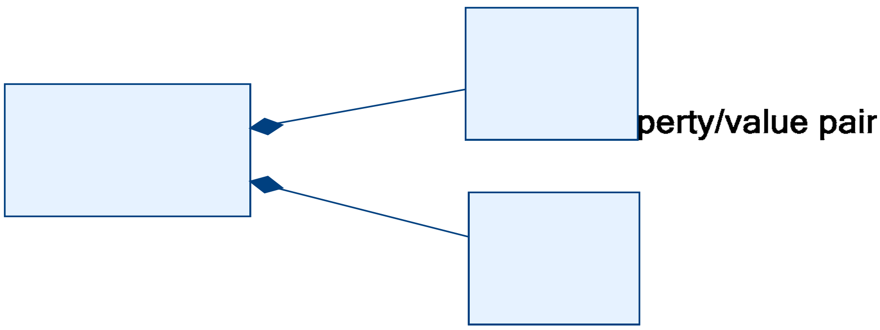


Figure 6: Proposed graphic representation of the relation between property/value pair, value and property

https://doi.org/10.23106/dcmi.952108542

In this case, the textual definition would be "Each property/value is composed of exactly one property and one value."
Continuing with the analysis we get to semantics associated with classes and properties.
The model graphic representation is shown in the following:

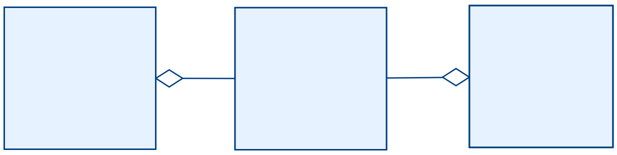


Figure 7: Graphic representation of the relations between semantic, class and property

This representation establishes that "Each property and class has some semantic evidence". It is evident that both the textual definition as well as the graphic representation are insufficient to represent the relations in a complete way.
First, we have to confront the meaning of the "semantic" concept. We know that the authors refer to the description of the meaning of the classes and properties, included typically in the textual form scheme. From our point of view, it is necessary to contemplate the possibility that a shared resource that could be from another scheme or an ontology would proportion the aforementioned semantic.
It is necessary, therefore, to distinguish between local semantics and shared semantics. From the view of the use of semantic, that is to say, assigning meaning to elements of a scheme, the semantic should be considered local, since it is assigned locally to the elements of the scheme. However, from the point of view of its definition, locally assigned semantic could be defined locally or from a shared resource, for example, an ontology.
In this way, the first modification on the semantic concept would be to add two same specialisation to represent both types of semantic, as shown in Figure 8:

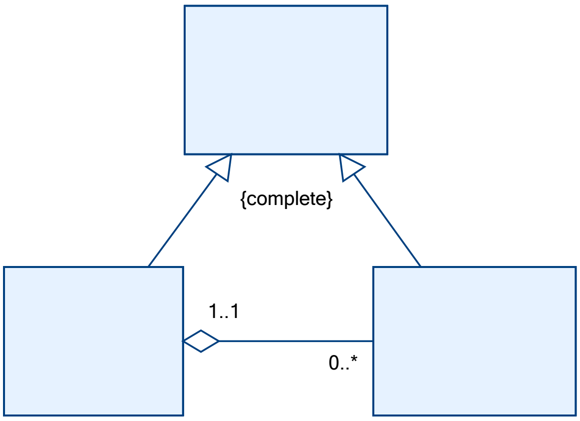


Figure 8: Graphic representation of the relation between semantic and its subclasses

We can say that semantic concept has been represented by an abstract class to denote that the definition of semantic should be carried out, obligatorily, in the way expressed by one of his sons.

https://doi.org/10.23106/dcmi.952108542

To go on, bearing in mind that the definition, for generalisation defect, presupposes disjoint subclasses (without labels) and that the subclasses set is not complete (all the possible subclasses have not been specified), the complete restriction has been included, to establish that the subtypes are the only two ways of expressing semantic.

Finally, to demonstrate the fact that local semantic could include references to externally defined concepts in the scheme, an aggregation has been included between local semantic and shared semantic concepts. An aggregate has been used since shared semantic instances make sense as independent elements when defined in external resources. The multiplicity of the relation in the extreme of the aggregate, denotes that shared semantic must be associated to an instance in local semantic to form part in the scheme. Also, semantic labelling existence is avoided restricting that a shared semantic instance could be associated to a local semantic instance at maximum. The multiplicity in the extreme of the aggregate denotes that a local semantic instance could aggregate none, if the definition is done locally (for example in a textual way), or various shared semantic instances. Permitting the semantic to be defined locally, without shared semantic, the interoperation and use of the scheme is made easy for those applications unable to interpret the proportioned semantic for a shared resource, also guaranteeing sharing with existing schemes that define the local semantic.

Semantic and its subtypes being defined, the relation with the class and property concepts should be analysed, proposing the graphic representation included in figure 9:

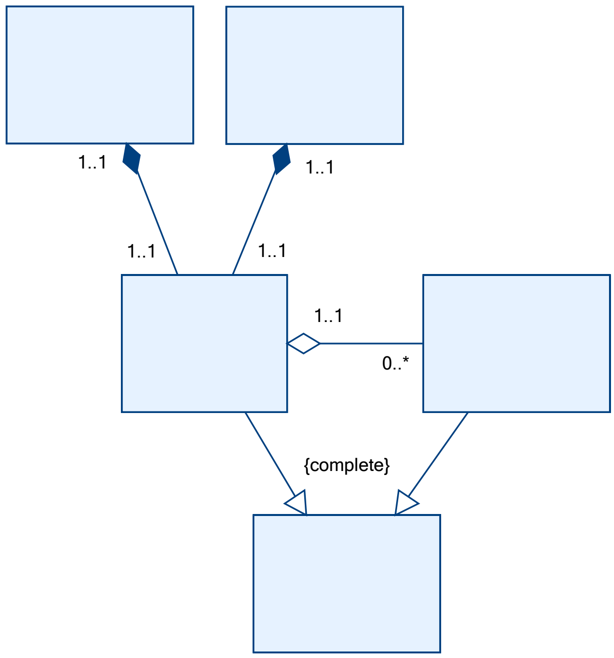


Figure 9: Graphic representation of classes and property semantic

As shown, the use of compositions are proposed to denote that local semantic is not shared and only makes sense as a part of a class or property.

As to the textual definition, in this case it is proposed: "Each property and class is composed of a local semantic in the scheme. The local semantic could be defined locally or aggregate the semantic concepts defined in shared resources as other schemes or

class https://doi.org/10.23106/dcmi.952108542 pro

ontologies. Shared semantic resources or shared semantic will always be associated to a unique local semantic".
Another analysed aspect is the concept represented as *shared semantics.* In difference to the previous proposal, in the original model, how the semantic shares the class and the subclass by the generalisation relation is defined. Also *refines class semantic* concept is included as shown in the following figure:

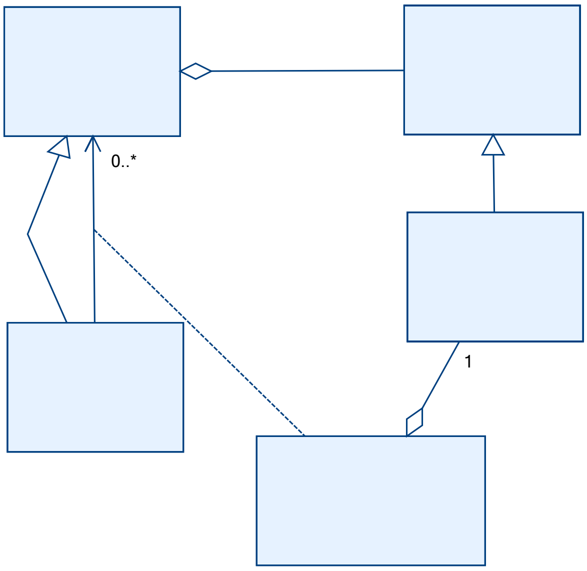


Figure 10: Graphic representation of semantic and classes related in the original model

class

Starting from the representation of the proposed semantic previously, and bearing in mind that the relation of generalisation implies that the subclass is a specific type of the parent class, being therefore a concept of the parent class and sharing its semantic; we think it unnecessary the representation of the originally represented concept as shared semantic and class refine semantic. Figure 11 includes the representation with the proposed changes:

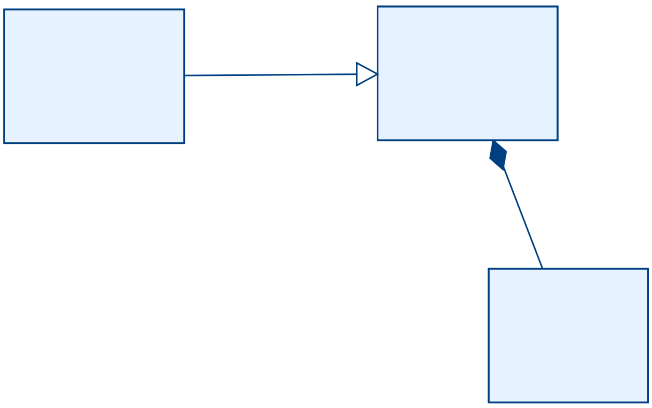


Figure 11: Proposed graphic representation for classes and subclasses semantics

With this representation the fact is denoted that one class has a local semantic and that one subclass, being a subtype of class inherits the composition defined in the parent which

re

https://doi.org/10.23106/dcmi.952108542

allows it to define its proper semantic. Also, as was mentioned previously, due to generalised relation the subclass is in last instance, the class type.
In a similar way as shown, the same logic can be used for the property concept.
As for the textual definition, it is proposed: "Each class could be related to one or more classes by a generalisation relation. In virtue of this relation, both classes share the semantic and relations of the parent class, in the way that all the subclass resource parts are from the father class".
In the case of the properties, the following textual definition is proposed: *"Each property can be related to one or more properties by a generalised relation. In virtue to this relation, both properties share the semantics and relation of the father property, in the way that the resources related to one value by one subproperty, are also related to the same value by the father property"*.
In addition to the commented aspects, how some of the elements in the model do not adjust to the UML notation is observed, as is the case of the representation of the multiplicity using the similar notation as used in the model Entity/Interrelation, this is: "0..n". To resolve this problem, as can be observed in the previous examples, the use of the proper UML notation is proposed as shown in figure 12:

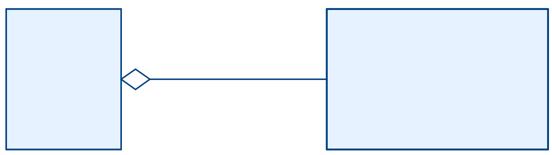


Figure 12: Example of multiplicity notation in UML

To make easy the interpretation of the multiplicity to readers less familiar with the UML notation, the use of summarized notation has been avoided. In this case, it would have been equivalent to represent the multiplicity as "*0..**" or simply "*".

resource

https://doi.org/10.23106/dcmi.952108542

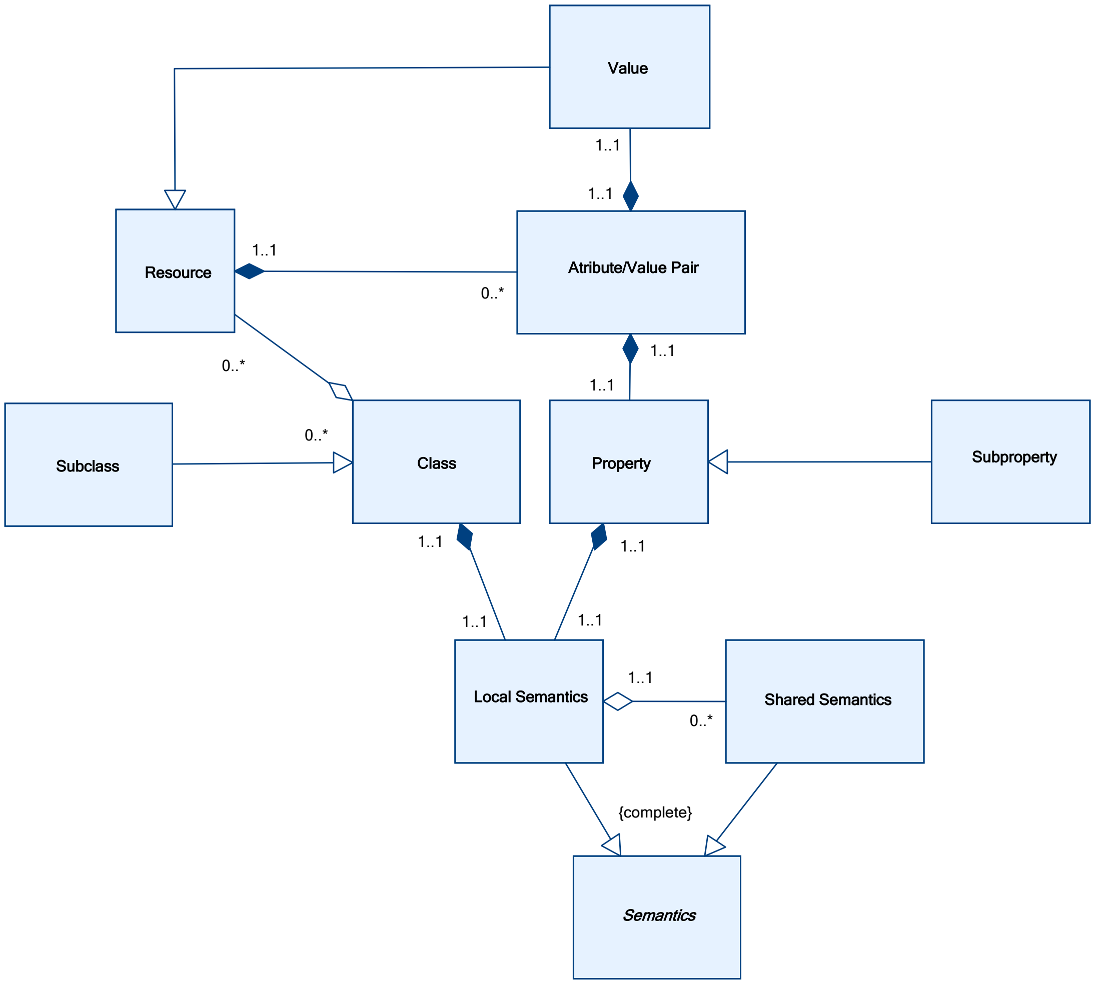


Figure 13: Graphic representation of the proposed resource model

Another aspect related with the notation and can make easy the interpretation of the model, is the specification of the multiplicity in the aggregate. In the model, the multiplicity of the aggregate in the extreme of the composed concept is not included and in some cases, is not included in the extreme of the component concept.
In the line of notion recommendation, we suggest the naming of classes in capital letters, proper of the notation UML with the purpose of distinguishing classes from other elements of the model as: objects, attributes, etc.
To conclude the analysis, Figure:13 shows the complete model with all the proposals made.

3. Conclusions
With the analysis and proposals presented, we hope to make easy the interpretation and use of the DCMI abstract model, presenting it with a formal notation and making clear those concepts susceptible to revision in relation to its definition and representation.
We hope that the proposals included could serve as an object of consideration for future revisions of the model.
At present, though it has not been included for extension reasons, the analysis of the

https://doi.org/10.23106/dcmi.952108542

description model, also included in the DCMI abstract model, is being developed. This analysis includes, in addition, the revision of coherence between the resource model and the description model.

References

1. DCMI (Dublin Core Metadata Initiative). DCMI Abstract Model. 2005. Available at <http://dublincore.org/documents/2005/03/07/abstract-model/> [Visited in april 2006].
2. OMG (Object Managment Group). Unified Modeling Language, version 1.5. Available at <http://www.omg.org/technology/documents/modeling_spec_catalog.htm#UML> [Visited in april 2006].
3. Reuse Company. umlCAKE: Herramienta CASE de modelado con UML. 2006. Available at <http://www.reusecompany.com> [Visited in april 2006].

https://doi.org/10.23106/dcmi.952108542