Adoption of the Classical Theory of Definition to Ontology Modeling

Patryk Burek

Onto-Med Research Group, University of Leipzig, Germany

burek@informatik.uni-leipzig.de

Abstract. Ontology modeling requires modeling languages expressive enough to represent various definition types. A definition type which seems to be of particular significance is that provided by the Classical Theory of Definition. In this paper we investigate if and how far the Classical Theory of Definition is adopted by some of the ontology modeling formalisms, namely by UML, ORM and DL. Moreover, we provide a means for representing some crucial issues in the context of the Classical Theory of Definition which seem to have no representation in the formalisms discussed. Among them are the identification of essential, peculiar and incidental predications and the representation of subsumption in the manner of the genus-differentia definition.

Keywords: Knowledge Representation, Ontology Engineering, Knowledge Modeling.

1 Introduction

The backbone of ontology modeling is the construction of a taxonomy of concepts founded on subsumption links. It seems so far that there is no agreement on the nature of subsumption [2], [17] and on the rules of taxonomy evaluation [5]. Moreover, one can observe that the definitional status of the concepts in most of the domains is not equal. There are various techniques used for defining concepts and there are tacit assumptions often lost during the ontology engineering/knowledge modeling process. The development of data models/ontologies not suffering from deformation of the input knowledge is still a difficult task. Of some help here may be the theories of concepts, definitions and categorization developed across philosophy, linguistics and cognitive science. Here we concentrate on the oldest of these theories, namely on the Classical Theory of Definition. The Classical Theory of Definition seems especially promising for modeling taxonomies, since it provides an interesting genus-differentia pattern for representing subsumption and permits making more explicit some of the tacit assumptions underlying the concept definitions.

In this paper we examine how far the Classical Theory of Definition is adopted by some of the languages proposed for ontology modeling. The formalisms investigated include Unified Modeling Language (UML), Object Role Modeling (ORM) and Description Logic (DL). Arguments for adopting UML, which was developed for software engineering, in ontology engineering are proposed in [4], [7]. It seems that

ORM, which is, like UML, a commonly used software engineering technique, may be of some use in ontology engineering for the same reasons. DL has recently become a prime candidate for ontology modeling especially in the context of the Semantic Web.

The purpose of this work is not to provide a comparison or a ranking of the formalisms discussed: our work concentrates only on the following issue: how far the Classical Approach is adopted in each of these formalisms and what consequences it may have on ontology modeling.

For those aspects of the Classical Theory of Definition that are not reflected in the formalisms discussed we propose a framework of definitional tags. Our framework of tags is not intended as a new formalism for ontology modeling but rather as the general extension pattern of the formalisms discussed. Among the issues supported by the framework introduced are the identification of essential, peculiar and incidental predications and the representation of subsumption in the manner of the genus-differentia definition.

In the case of UML, only class diagrams will be investigated and only classes will be interpreted as *ontological* concepts. The notions of concept, UML class and ORM entity type are used here as equivalent.

We use the notions of concept extension and intension as they are generally accepted in the literature. By concept extension we understand the set of all objects for which a concept can be truly predicated. By concept intension we understand a set of its *defining characteristics* [15].

Section 2 discusses the basic tenets of the Classical Theory of Definition. In section 3 the distinction between essential, peculiar and incidental components of definitions is discussed. In section 4 the genus-differentia definition is analyzed, and in section 5 conclusions are presented. The overall structure of sections 2, 3 and 4 is as follows: first the preliminaries are presented, secondly DL, UML and ORM are analyzed with respect to the given issues and finally our proposal is introduced.

2 Basic Tenets of the Classical Theory of Definition

We consider here the Classical Theory of Definition in a broad sense dating back to ancient Greek philosophy. The following two fragments encode the Classical Theory of Definition: [1]

> Most concepts (esp. lexical concepts) are structured mental representations that encode a set of necessary and sufficient conditions for their application. [8]
>
> The Classical Theory of Definition has two principal tenets: that there are intensional definitions for each of the class terms which we use; and that a 'proper' intensional

[1] The first paragraph refers to the Classical Theory of Definition, concerned with the structure of definition, while the second refers to the Classical Theory of Concepts concerned with the structure of concepts. For our purposes this distinction is not relevant so we treat both theories as aspects of the same approach, called later the Classical Approach or the Classical Theory of Definition.

> definition states in the definiens the logically necessary and sufficient conditions for the application of the definiendum. [15]

Both fragments have in common the optimistic assumption that concepts are definable by the necessary and sufficient conditions of concept application.

Furthermore, both fragments reveal the second tenet of the Classical Approach concerning the definition's structure: the definition is a compound of sufficient and necessary conditions for the concept application. The definition consists of the definiendum, the definiens and the copula jointing them. The definiendum contains the concept to be defined. Of interest for our purposes here are the definitions where the definiendum contains only the defined term – explicit definitions.

The definiens defines the definiendum and is understood here as the conjunction of the true predications about the definiendum, although other functional patterns of the definiens [10] may be investigated in the future. The definiens and the definiendum are linked by the copula being the equivalence functor, which indicates that the definiens provides both sufficient and necessary conditions.

2.1 Analysis: Basic Tenets in DL, UML and ORM

Generally in knowledge modeling the first tenet of the Classical Approach is commonly accepted - knowledge is definable and presentable in the form of intensional definitions. The second tenet seems to be accepted too. We will see that the classical structure of the definition is adopted in UML, ORM and DL.

In DL concepts are defined in the Terminology Box (TBox) by explicit definitions in the classical manner. The definitions can be equivalences with only the defined concept in the definiendum [1].

In UML definitions have a graphical form. Each class is represented by a rectangle divided into compartments separated by horizontal lines (Fig.1) [12]. The top compartment containing only a class name can be interpreted as the definiendum. The middle list compartment holds the list of attributes, the bottom one the list of operations. Together with associations assigned to the class they provide the intension specification and can be interpreted as the definiens.

In UML classes are defined by necessary and sufficient conditions. The full list of class attributes, operations and associations delimits precisely the direct class extension. Hence we see that UML meets also the second tenet of the Classical Approach.

In ORM concepts, called entity types, are also defined in a graphical form (Fig 2) [6]. Each entity type is presented as an oval with a name of the entity type in it. The oval with the name can be interpreted as the definiendum. Entity types play roles in facts. Facts are depicted as rectangles divided into named boxes. Each box represents the role of the entity linked with it by an arc. The list of all roles played by an object provides the necessary and sufficient conditions for instantiation of the entity.

We see then that all three formalisms permit the definitions of the classical structure. UML and ORM moreover seem to restrict the definition types to only the classical one, while DL allows also another, less restrictive, definition type in form of the inclusion axiom.

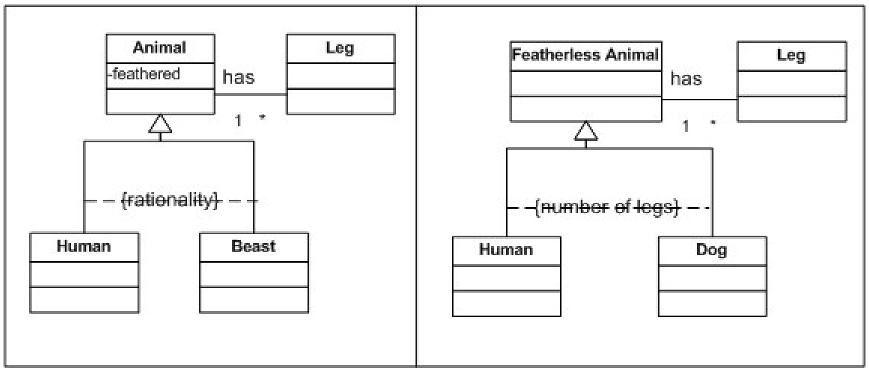

Fig. 1. An UML diagram with a discriminator

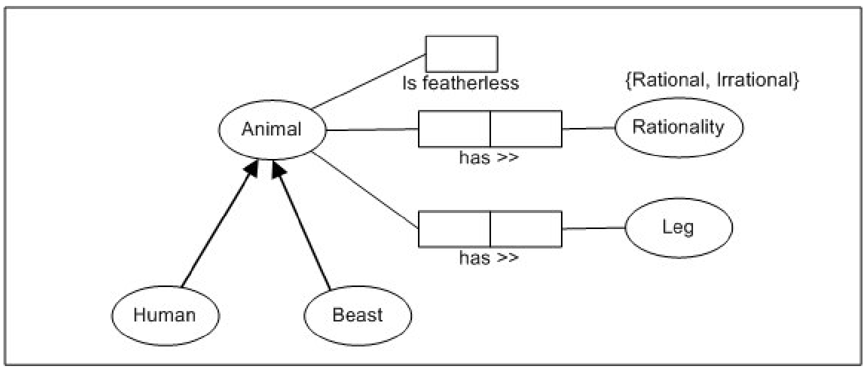

Fig. 2. An ORM diagram

3 Account of Essence

Originally, the classical definition not only provided necessary and sufficient conditions, but according to Aristotle it was the account of essence [9], [13]. An account of essence is the answer to the question "what makes a thing what it is?". The sentence "a human is an animal" tells something which is fundamental (essential) for *human*, but on the other hand saying that "human is civilized" is not fundamental for *human* but rather something that merely happens to be universally true for *human* and only for him. The second predication, in contrast to the essential, is called peculiar. The predication reflecting nonessential features that are not shared by all referents of the concept is called incidental [9], [13].

It is questionable whether one can reveal the essence of things, and whether such a demand is not the result of the excessively high ambitions of some philosophers. This question is, however, outside the scope of our interest here. The point here is to distinguish statements that are true parts of definitions from those which only happen to be true. Hence we understand an essence in a "lite" version, not as the true objective nature of things, but as the intentions of a definition's author. The essence of a concept is what an author of the concept's definition believes is fundamental for an object to be understood as a member of the concept's extension.

3.1 Analysis: Account of Essence in DL, UML and ORM

The distinction between essential, peculiar and incidental predications is partially adopted in systems performing classification, where the properties that are true parts of definitions are separated from those nonessential. The nonessential properties are shared by all members of the concept's extension but are not considered as parts of the concept's definition. The classification algorithm requires that the position of a concept in a taxonomy does not depend on contingent facts, even those which are universally true. In CLASSIC [3] nonessential properties are modeled not as parts of a concept's definition but as rules that do not take part in the classification. Brachman suggests in [1] to model nonessential properties in DL not as parts of definitions in TBox but in ABox. Nonessential properties are treated then not as definitional knowledge but as assertional knowledge.

Here one may raise the question of whether the assertion that "Socrates is a human" and the assertion that "a human is civilized" have the same relevance for the definition of the human. It seems that the second statement although not essential for humans still plays a definitional role while the first does not. The ABox does not provide any means to separate the assertions involved, even non-essentially, in definitions, from those not involved in definitions at all. The definitional relevance of nonessential predication seems to be of particular importance in cases where revealing the essence of concepts is problematic. In many domains, including scientific and technical domains, definitions are not intended to provide an account of the essence. It seems that authors of those definitions do not intend to point at any fundamental features of concept referents. In such cases, if we strictly follow the above suggestion many concepts would lack definitions in TBox completely and all knowledge stated about them would be contained only in ABox, where nonessential but definitional knowledge would be mixed with purely assertional knowledge.

We observe then that the explicit representation of the distinction between essential, peculiar and incidental properties in DL runs into some problems. Neither in UML nor ORM do we find a means to represent it.

3.2 Proposal: Definitional Tags

The essential, peculiar and incidental properties can be introduced by the definitional tags. The following tags can be introduced: [essential], [peculiar], and [incidental]. These tags can be assigned to any predication present in the definiens, for example:

$$\text{human} \equiv [\text{essential}]\text{animal} \wedge [\text{essential}]\text{rational};$$

$$\text{human} \equiv [\text{essential}]\text{animal} \wedge [\text{peculiar}]\text{rational}.$$

In both examples the extension of *human* is the same. The intentions seem to be equal as well, since both definitions use the same, *animal* and *rational,* predications.

What distinguishes these two definitions is the relevance of the predication *rational*. In the first definition *humans* are essentially rational animals (*humans* are essentially *animals* and essentially *rational*). In the second example *humans* are essentially *animals* but the fact that that they are *rational* is not essential, although common to all *humans*. It seems that although both definitions have the same extension and intension the assumptions and the intents behind them are different. The definitional tags [essential], [peculiar], [incidental] permit us to grasp these differences.

4 Genus-Differentia Definition

A well-known representative of the Classical Approach is the genus-differentia definition. It was introduced by Aristotle and later elaborated by medieval philosophers [9], [14]. It has a peculiar structure, where the definiens is composed of two elements: the genus and the differentia. The genus subsumes the defined concept and its extension should be possibly the closest to the extension of the defined concept, thus the genus should be the nearest (genus proximum). The differentia specifies the features distinguishing the referents of the defined concept from the referents of other concepts subsumed by the same genus. An example of the genus-differentia definition is the Aristotelian definition of a human: "a human is a rational animal". Here an *animal* is the genus and *rational* is the differentia which distinguishes *humans* from *beasts*.

4.1 Analysis: Genus-Differentia Definition in DL, UML and ORM

DL permits definitions of the form: $human \equiv animal \wedge rational$. However the elements of the conjunction are not recognizable in any way as the genus and the differentia. Hence we see that DL does not permit us to explicitly identify the roles played by the components of the genus-differentia definition.

In UML, a generalization with a discriminator can be interpreted as the genus-differentia definition. The generalization associates the class to its parent class. The parent class can thus be interpreted as the nearest genus. The discriminator names the partition of the parent class. The discriminator joins the generalization arc with other generalization arcs taking part in the same partition. In our example on the left-hand side of Fig.1, *rationality* is a discriminator.

The discriminator name must be unique among the attributes and association roles of the parent class. Multiple occurrences of the same discriminator name are permitted and indicate that the children belong to the same partition. Hence one could say that the discriminator names the feature to which the differentia refers. However, it does not name the differentia itself nor does any other element of UML. All attributes, operations and associations of the child class distinguish it form the parent. However, they can not all be treated as the differentia, since some of them may be shared with other children in the same partition.

In ORM subtype entities are introduced by subtype links which are directed line segments from subtypes to supertypes and by the definitions written under the diagram. The following two definitions specify the subtype entity *human* on Fig. 2: Each *human* is an *animal* which is *rational*; Each *human* is an *animal* which is *featherless* and has two *legs*. The first example in the context of the diagram presented in Fig. 2 can be read: Each *human* is an *animal* who has *rationality* of level "rational". "Rational" is a value of a reference mode of the entity type *rationality* related to the supertype a*nimal.* It plays the role of the differentia in the definition of *human*. In the second example, the differentia is a conjunction of two conditions. The first one is a unary fact *featherless* related to the supertype a*nimal*, the second is the number restriction on the *has (legs)* fact.

The above examples show that the definition pattern adopted in ORM allows every diagram element to play the role of the differentia. However, in ORM the differentia is not marked out in the diagram, and is present only in the additional definition below the diagram. Neither the value *rational,* the fact *featherless* nor the number restriction 2 on the fact *has (legs)* are marked in any way on the diagram as the differentia. Their application in the definition of an entity type *human* cannot be read from the diagram. Moreover, we can observe that the discriminator is not present in ORM at all.

4.2 Proposal

We propose to decompose the notion of differentia to the notions of relationally interpreted differencing principle and difference. Then we introduce the definitional tags for representing subsumption by the pattern of the genus-differentia definitions.

4.2.1 Differencing Principle

The notion of the genus is present in all three formalisms discussed and is crucial for ontology modeling. However, the notion of the differentia is not so popular. Two different aspects seem to be of importance in the context of the differentia.

The first one is the discriminator present in UML. The discriminator can be interpreted as the principle of the class partition. Hence we call the discriminator the *differencing principle* applied to the parent class. A close look to the differencing principle shows that it could be understood as the attribute or the role of the parent class. This corresponds to the UML requirements for the uniqueness of the discriminator name among the parent class attributes and roles.

Furthermore we can observe that the discriminator refers to some other element of the class model. In the right-hand side of Fig. 1 the discriminator *number of legs* refers to the multiplicity of the association ending. UML, however, does not in any way point to the model element that is used as the discriminator. The discriminator in UML is independent from all other model elements.

Instead of treating the discriminator as independent from other model elements we suggest interpreting it as a relational entity. We interpret the discriminator as the role of some model element in the partition of the given parent class. We say that some model element is applied to the given class as a differencing principle. In this case the multiplicity of the association end *number of legs* plays the role of the differencing principle applied to the class *animal*.

The differencing principle is the ground on which subclasses of the class to which it is applied are distinguished. In other words the differencing principle is a branching point of an ontology or an ontological choice. By applying differencing principles to categories, the hierarchical structure of the ontology is built.

The differencing principle can be interpreted as the attribute of the class it is applied to or as a question grounding the partition of that class. *Is rational?* and *how many legs does it have?* are both questions underlying the partition of the class *animal*. The differencing principle presented in the form of the question may be particularly fruitful. A linguistic analysis of the question structure can provide the additional properties of differencing principles and can help in further analysis of subsumption links based on them.

4.2.2 Difference

The second relevant aspect of the differentia, in addition to the differencing principle, is the differentia itself, which is the difference distinguishing the child class. It justifies the introduction of a child class in the context of a given partition. The differentia cannot be identified explicitly in UML or DL but only in ORM's subtypes definitions. The differentia can be interpreted as the value of the differencing principle or the answer to the question stated in the differencing principle. In the Aristotelian definition of human being, *rationality* is the differencing principle and the value *rational* is the differentia.

The differentia firstly distinguishes the child class from the parent class. *Human* is distinct from *animal* since it is *rational*. Secondly, the differentia distinguishes a class from other children classes in the given partition (with the same underlying differencing principle). This, however, only holds for the classification, while in the case of a typology the children concepts may overlap.

4.2.3 Genus-Differentia Tags

We propose to interpret the hierarchical structure based on the genus-differentia definition as the application of a differencing principle to a parent class, where the differencing principle is the role of some element of the model or has the references extending the model. We propose a genus-differentia subsumption pattern based not on a binary subsumption relation, graphically presented by the arrowed edge linking the parent class with the child class, eventually with a discriminator, but as a relation with the following four arguments:

1. Genus as the supertype;
2. Species as the subtype;
3. Differencing principle as the role of some other model element or as the external reference;
4. Differentia - the property of the species interpreted as the corresponding value of the differencing principle or as the answer to the question stated by it.

These four elements of the subsumption can be identified by the following three tags: [gen], [spec], [diff ref=””]. The tags [gen] and [spec] identify, respectively, the supertype and the subtype; the differentia is tagged with [diff ref=””] where ref=”” refers to the element playing the role of the differencing principle. For example the definition

“A human is a rational animal” annotated with the tags would have the following form:

[spec]human ≡ [gen]animal ∧ [diff ref=”rationality”]rational;

where [spec]human and [gen]animal state, respectively, that *human* is a species and *animal* is a genus. [diff ref=”rationality”]rational states that *rational* is the differentia with the underlying differencing principle *rationality* which is present in the model or refers to an external source.

5 Conclusions

In the current paper we analyzed how far the Classical Theory of Definition is adopted by the selected ontology modeling languages. We analyzed UML, ORM and DL with respect to the issues relevant for the Classical Theory of Definition. We have observed that some of the tenets of the Classical Approach are accepted in all three formalisms discussed, for example the optimistic assumption of definability of knowledge by both sufficient and necessary conditions. However, we have found that some of the issues of the Classical Approach are not supported by any of the formalisms discussed, like the distinction between essential, peculiar and incidental predications, and some are supported only partially, like the genus-differentia definition. To enable the representation of these issues we have proposed a framework of the definitional tags.

We believe that the Classical Approach adopted by the tags offers several advantages for modeling the taxonomical structures of concepts. Among them the following:

1. Tags [essential], [peculiar], [incidental] identify the tacit intuitions of the author of a definition; separate the essential knowledge from the nonessential; and permit concepts to be distinguished even if their intension and extension are equal.
2. Tags [spec], [gen], [diff ref=””] bind a parent concept and a child concept with the principle of partition and make the subsumption more explicit:
   - [ref=””] identifies the differencing principle, which is a branching point of the taxonomy and serves as the principle of partition. It seems particularly useful in the case of multiple partitions of parent concept;
   - [diff ref=””] labels the characteristics that distinguish the concept in the given partition and that makes the concept a member of that partition. In the case of the multiple subsumption it enables us to state the features by which a concept is assigned to each partition;
   - [ref=””] reduces redundancy in the case of UML and ORM. Instead of adding a new independent model element, one already present in the model is identified as the differencing principle;
   - [ref=””] permits the definitional dependencies in UML and ORM to be traced. It enables one to identify which concepts are suitable as the foundations for the definitions of other concepts. In our second example, we observe that for defining the concept *human,* the concept of *rationality* is needed;
   - [ref=””] identifies the borderlines of the model. In some cases the differencing principle may refer to a concept that is not present in a model. By treating a dif-

ferencing principle as the relational entity, we can state explicitly that the model refers at this point to the external source.

The tags introduced are not intended as a new ontology modeling formalism but rather as a general modeling pattern that could be embodied as an extension to the formalisms discussed. The framework is not intended to be a normative theory of taxonomy evaluation either as it is in the OntoClean approach [5]. Nevertheless, the important task which exceeds the scope of this paper, however, is to compare the tags presented with the meta-properties of OntoClean.

Acknowledgments

I am indebted to Professor Heinrich Herre, Rafał Graboś, Frank Loebe, Hannes Michalek for fruitful discussions and to Hesham Khalil, Evan Mellander and the anonymous reviewers for feedback on earlier versions of this paper.

References

1. Baader, F., et al. (eds.): Description Logic Handbook. Cambridge University Press (2002)
2. Brachman, R.: What isa is and isn't: an analysis of taxonomic links in semantic networks. IEEE Computer, 16(10) (1983)
3. Brachman, R. et al.: Living with CLASSIC: When and how to use a KL-ONE-like language. In Sowa J (ed.): Principles of Semantic Networks. Morgan Kaufmann (1991)
4. Cranefield S., Purvis M.: UML as an ontology modeling language. Proc. IJCAI-99 (1999)
5. Guarino, N., Welty C.: Ontological Analysis of Taxonomic Relationships. In Leander, A., Storey, V. (eds.): Proc. ER-2000. Springer-Verlag LNCS (2000)
6. Halpin, T.: Object-Role Modeling. In Bernus, P., Mertins, K., Schmidt, G. (eds.): Handbook on Architectures of Information Systems. Springer-Verlag, Berlin (1998)
7. Kogut P., et al.: UML for Ontology Development. http://www.sandsoft.com/docs/ker4.pdf
8. Laurence, S., Margolis. E. Concepts and Cognitive Science. In Laurence, S., Margolis. E. (eds.): Concepts Core Reading. MIT Press (1999)
9. Kotarbiński, T.: Wykłady z Dziejów Logiki. PWN. Warszawa (1985)
10. Rickert, H.: The Theory of Definition. In Seger J., C.: Essays on Definition. John Benjamins Publishing Company. Amsterdam / Philadelphia (2000)
11. Robinson, R.: Definition. Oxford University Press (1963)
12. Rumbaugh, J., Jacobsson, I., Booch, G.: The Unified Modeling Language Reference Manual. Addison Wesley Longman Inc. (1999)
13. Smith, R.: Aristotle's Logic. In Edward N. Zalta (ed.): The Stanford Encyclopedia of Philosophy. URL = http://plato.stanford.edu/archives/fall2003/entries/aristotle-logic. (2003)
14. Sowa J.: Knowledge Representation. Brooks/Cole (2000)
15. Swartz, N.: Definitions, Dictionaries, and Meaning. URL=http://www.sfu.ca/philosophy/swartz/definitions.htm#part2.1 (1997)
16. Temmerman, R.: Towards New Ways of Terminology Description. John Benjamins Publishing Company. Amsterdam / Philadelphia (2000)
17. Woods, W.: Understanding subsumption and taxonomy: A framework for progress. in Sowa, J.F. (ed.) Principles of Semantic Networks. Morgan Kaufmann, San Mateo, CA (1991)