9 The Ontology UML Profile

UML profile is a concept used for adapting the basic UML constructs to a specific purpose. Essentially, this means introducing new kinds of modeling elements by extending the basic ones, and adding the new elements to the modeler's repertoire of tools. Also, free-form information can be attached to the new modeling elements. The Ontology UML Profile extends UML in a standard way to enable ontology modeling in the widely used UML modeling tools.

The OMG adopted specification (OMG ODM 2007) contains a formal specification of UML profile for RDFS and OWL. The profile presented in this chapter is our proposal from 2003 (Djurić et al. 2005b), which is intended to be practical and easy to use, and still be highly compliant with the OWL and RDFS standards. We believe that we succeeded in this goal; that is, we are presenting it in this chapter in a narrative form through examples. It is a good source for learning because the standard is very similar, and after going through these examples you will be able to quickly adapt to any changes in future versions and revisions of standard OUP.

9.1 Classes and Individuals in Ontologies

The Class is one of the most fundamental concepts in the ODM and the Ontology UML Profile. As we noted in the discussion about the essential concepts of the ODM, there are some differences between the traditional UML Class or the concept of a Class in object-oriented programming languages and an ontology class as it is defined in OWL (owl:Class). Fortunately, we are not trying to adopt UML as a stand-alone ontology language, since that might require changes to the basic concepts of UML (Class and others). We only need to customize UML as a support to the ODM.

In the ODM, the concepts that represent classes, i.e., RDFSClass, OWLClass, OWLAllDifferent, and OWLRestriction are modeled using the MOF Class concept. These constructs in the Ontology UML Profile are inherited from the UML concept that is most similar to them, UML Class.

D. Gašević et al., *Model Driven Engineering and Ontology Development*, 2nd edn.,
DOI 10.1007/978-3-642-00282-3_9, © Springer-Verlag Berlin Heidelberg 2009

However, we must explicitly specify that they are not the same as UML Class, which we can do using UML stereotypes. An example of Classes modeled in the Ontology UML Profile is shown in Fig. 9.1.

RDFSClass and OWLClass, ontology classes identified by a class identifier, have the stereotype «RDFSClass» or «OWLClass», OWLAllDifferent has the stereotype «OWLAllDifferent»; and OWLRestriction has the stereotype «OWLRestriction».

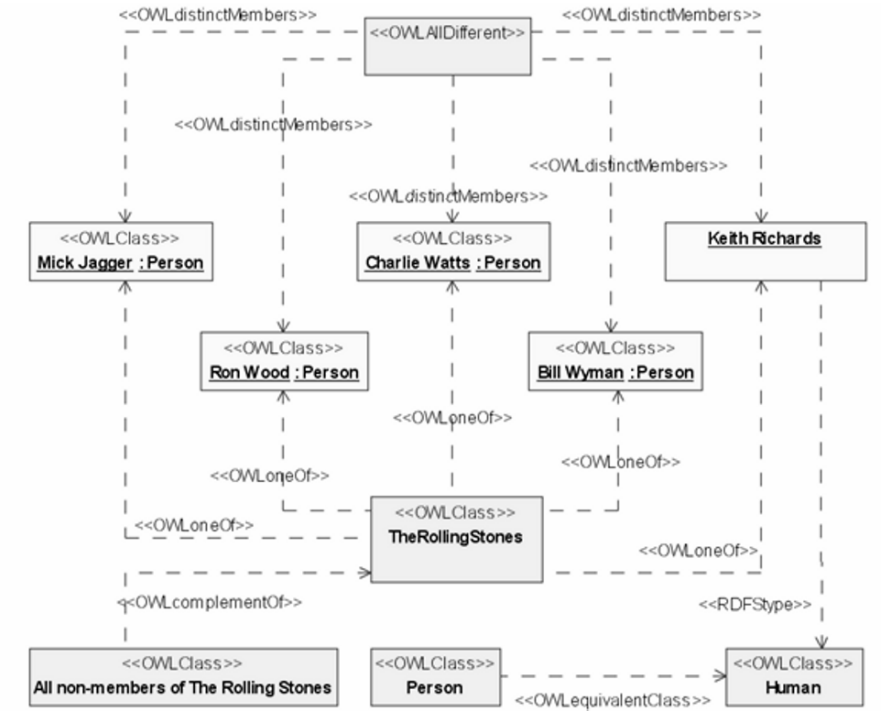

Fig. 9.1 Class diagram showing relations between ontology classes and individuals in the Ontology UML Profile

Figure 9.1 shows various types of ontology classes modeled in UML. The «OWLClass» Person is an example of an owl:Class class that is identified by a class identifier, while TheRollingStones is an enumeration. There is a class "All non-members of The Rolling Stones" that represents the complement of The Rolling Stones—all individuals whose type is not The Rolling Stones belong to this class. AllDifferent is an auxiliary class whose members are different individuals. Also shown are the «OWLClass» Human and the «equivalentClass» Dependency, which means that Person and Human are classes that have the same class description (i.e., all

Persons are Humans and vice versa). Note that in object-oriented modeling it would be highly unusual to model The Rolling Stones as a class rather than as an object of type RockNRollBand. However, ontology classes are not behavioral, but sets; and what would you call a set of all members of The Rolling Stones? Obviously—The Rolling Stones.

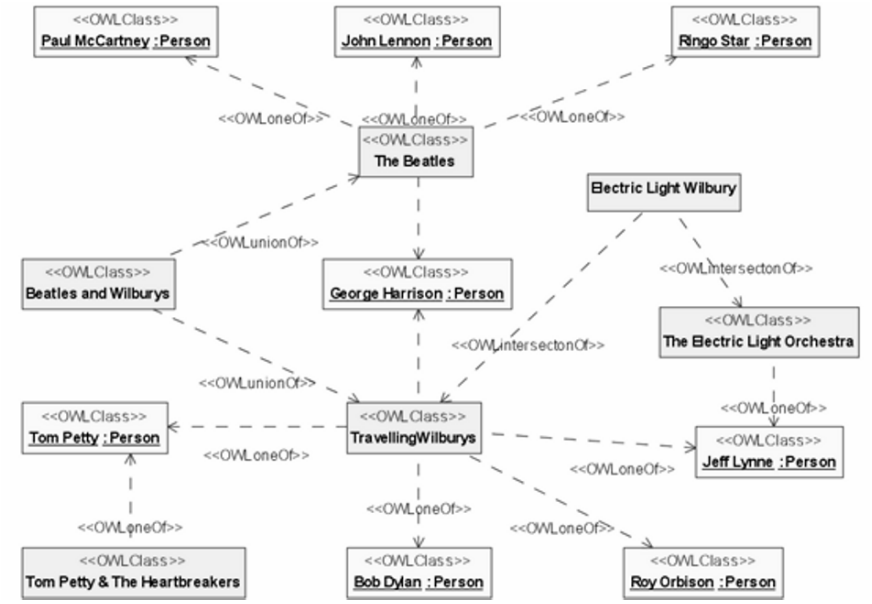

Fig. 9.2 Constructing union and intersection in an Ontology UML Profile

In the ODM, an instance of OWLClass is an OWLThing, an individual. An instance of RDFSClass is an RDFSResource, which means that it can be anything. In UML, an instance of a Class is an Object. OWLThing and UML Object have some differences, but they are similar enough; and so in the Ontology UML Profile, an OWLThing is modeled as a UML Object, which is shown in Figs. 9.1 and 9.2. The stereotype of an object must match the stereotype of its class («OWLClass» in this case). The «OWLThing» stereotype could be added as well. We can state that an individual has a type in three ways:

- By using an underlined name of an individual followed by ":" and its «OWLClass» name. For example, Mick Jagger:Person is an individual (OWLThing) whose type is Person. This is the usual UML method of stating an Object's type.

- By using a stereotype «RDFStype» between an individual and its «OWLClass». This method is also allowed in standard UML using the stereotype «instanceOf». For example, Keith Richards has «RDFStype» dependency link to Human, which is equivalent with Person («OWLequivalentClass»). Thus, he is also a Human, just like other members of The Rolling Stones.
- Indirectly, through logical operators on «OWLClass». If an «OWLClass» is a union, intersection, or complement, it is a class of individuals that are not explicitly defined as instances of it. For example, in Fig. 9.2 Bob Dylan is not explicitly defined as a member of the Beatles and Wilburys union class, but he is its member since he is a member of Travelling Wilburys, which is connected with the Beatles and Wilburys through an «OWLunionOf» connection. A similar thing applies to Jeff Lynne and the Electric Light Wilbury Class. Since he is a member of the Travelling Wilburys and The Electric Light Orchestra, he is a member of Electric Light Wilbury, an «OWLintersectionOf».

Again, do not confuse an instance-of relationship between a UML Object and a UML Class or between an OWLThing and an OWLClass (all in the M1 layer) with the relationship between, for example, an instance of OWLClass (M1) and an OWLClass concept (M2). The latter is a linguistic instance-of relation, an instance-of relation between concepts from different layers (the definition of OWLClass and a concrete OWLClass, for instance: Tom Petty & the Heartbreakers). The ontological instance-of relation is an instance-of relation between concepts that are in the same linguistic layer, but in different ontological layers (for instance, «OntClass» Person and the object George Harrison are at different ontological layers since Human is the class [type] of George Harrison). For a more detailed discussion of ontological versus linguistic instance-of relations, see Atkinson and Kühne (2003).

9.2 Properties of Ontologies

The concept of Property is one of the most unsuitable concepts in ontologies for modeling with object-oriented languages and UML. The problem arises from a major difference between Property and the UML concepts similar to it, Association and Attribute. Since Property is an independent, stand-alone concept, it cannot be modeled directly with Association or Attribute, which cannot exist on their own. Some authors (Baclawski et al. 2002a) have suggested extending UML with new constructs to support a

stand-alone Property, introducing aspect-oriented concepts into UML. In our view, this solution is rather extreme, since it demands nonstandard changes to UML.

Since Property is a stand-alone concept, it can be modeled using a stand-alone concept from UML. That concept could be the UML Class stereotype «RDFProperty», «OWLObjectProperty», or «OWLDatatypeProperty». However, Property must be able to represent relationships between Resources (Classes, Datatypes, etc., in the case of UML), which a UML Class alone is not able to do. If we look at the definition of a Property in the ODM more closely, we can see that it accomplishes representation of relations through its range and domain. We have found that in the Ontology UML Profile, the representation of relations in accordance with the ODM model should be modeled with the UML Association's or Attribute's stereotypes «domain» and «range». In order to increase the readability of diagrams, the «range» association is unidirectional (from a Property to a Class).

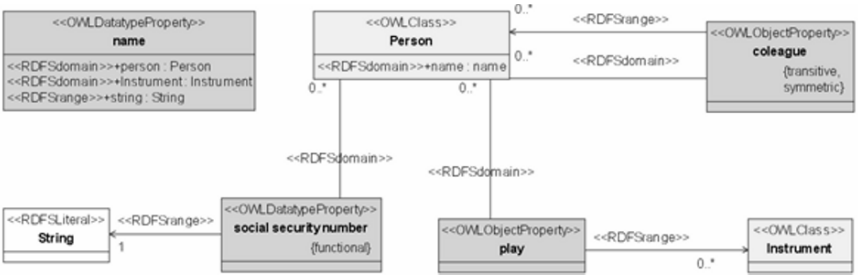

Fig. 9.3 Ontology properties shown in a UML Class diagram

OWL defines two types (subclasses) of Property—OWLObjectProperty and OWLDatatypeProperty. OWLObjectProperty, which can have only individuals in its range and domain, is represented in Ontology UML Profile as the Class stereotype «OWLObjectProperty». OWLDatatypeProperty is modeled with the Class stereotype «OWLDatatypeProperty».

An example of a Class Diagram that shows ontology properties modeled in UML is shown in Fig. 9.3. It contains four properties: two «OWLDatatypeProperty»s (name and socialSecurityNumber) and two «OWLObjectProperty»s (play and colleague) UML Classes. In cooperation with «RDFSdomain» and «RDFSrange» UML Associations, or «RDFSdomain» and «RDFSrange» UML Attributes, these properties are used to model relationships between «OWLClass» UML Classes. Tagged values describe additional characteristics; for example, the «OWLObject-Property» colleague is symmetric (if one Person is a colleague of another

Person, the other Person is also a colleague of the first Person) and transitive (if the first Person is a colleague of the second Person, who is a colleague of the third Person, the first and the third Person are colleagues).

There is an important issue that must be clarified with respect to this diagram. In UML, relations are represented by Associations (represented graphically as lines) or Attributes, which looks nice and simple. Ontology UML Profile diagrams may look overcrowded, since each relationship requires a box and two lines to be properly represented. The solution employed here uses standard graphical symbols, but UML allows custom graphical symbols for a UML profile. For example, a custom graphical symbol for a Property could be a tiny circle with lines, which reduces the space required on a diagram. Additional custom settings, such as distinct colors for «OWLClass» (green), «OWLObjectProperty» (orange), and «OWLDatatypeProperty» (orange), can be used to increase the readability of diagrams. For the sake of readability, the UML profile that we have used allows two styles of presentation of the domain and range of an «OWLDatatypeProperty». An example of the first style (a UML Class with two UML Associations) is socialSecurityNumber; and an example of the second one (a Class with Attributes as its domain or range) is name. The second style is allowed only for an «OWLDatatypeProperty» whose range multiplicity is equal to or less than one. So, if an «OWLDatatypeProperty» has a range multiplicity of 0..1 or 1, the style using Attributes can be used to reduce the clutter.

9.3 Statements

OWLStatement is a concept that represents concrete links between ODM instances—individuals and data values. In UML, this is done through Link (an instance of an Association) or AttributeLink (an instance of an Attribute). A Statement is a kind of instance of a Property, which is represented by a UML Class stereotype («OWLObjectProperty» or «OWLDatatypeProperty»). Since an instance of a Class in UML is an Object, a Statement in the Ontology UML Profile is modeled with the Object's stereotype «OWLObjectProperty» or «OWLDatatypeProperty» (the stereotype of an Object in UML must match the stereotype for its Class' stereotype). UML Links are used to represent the subject and the object of a Statement. To indicate that a Link is the subject of a Statement, LinkEnd's stereotype «RDFsubject» is used, while the object of the Statement is indicated with LinkEnd's stereotype «RDFobject». LinkEnd's stereotypes are used because, in UML, Link cannot have a stereotype.

These Links are actually instances of Property's «RDFdomain» and «RDFrange». In brief, in the Ontology UML Profile a Statement is represented as an Object with two Links—the subject Link and the object Link, which is shown in Fig. 9.4. The Persons represented, Mick Jagger and Keith Richards, are colleagues. Keith Richard also plays an Instrument, guitar.

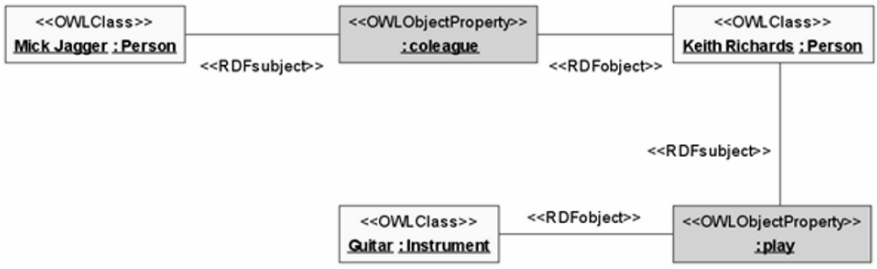

Fig. 9.4 Individuals and statements shown in a UML Object diagram

As with Properties of an ontology, the diagram's readability can be further increased by using distinct colors and custom graphical symbols. A tiny circle can be used instead of the standard box for representing a Statement in order to reduce clutter in a diagram.

9.4 Different Versions of the Ontology UML Profile

The ODM specification (and especially the part that deals with the Ontology UML Profile) is still under development, although it approaches the final specification. For that reason, the final version of the Ontology UML Profile will probably be different than the version we have described. However, the version described here should be very useful for getting a feeling for what it is like to create ontologies with UML. It is very easy to get accustomed to a similar profile once you have got a feel for working with one profile.

To show you what the differences could look like, we shall show you diagrams of an ontology similar to those which we have just talked about. There is another reason why we are showing these diagrams here. Some of the tools described and the discussion in this book refer to this older version of the Ontology UML Profile (Djurić et al. 2005b) which is called GOOD OLD AI Ontology UML Profile. This profile was later updated (but there is no point in updating tools until the official specification has been finished).

So, here is what classes look like (Fig. 9.5):

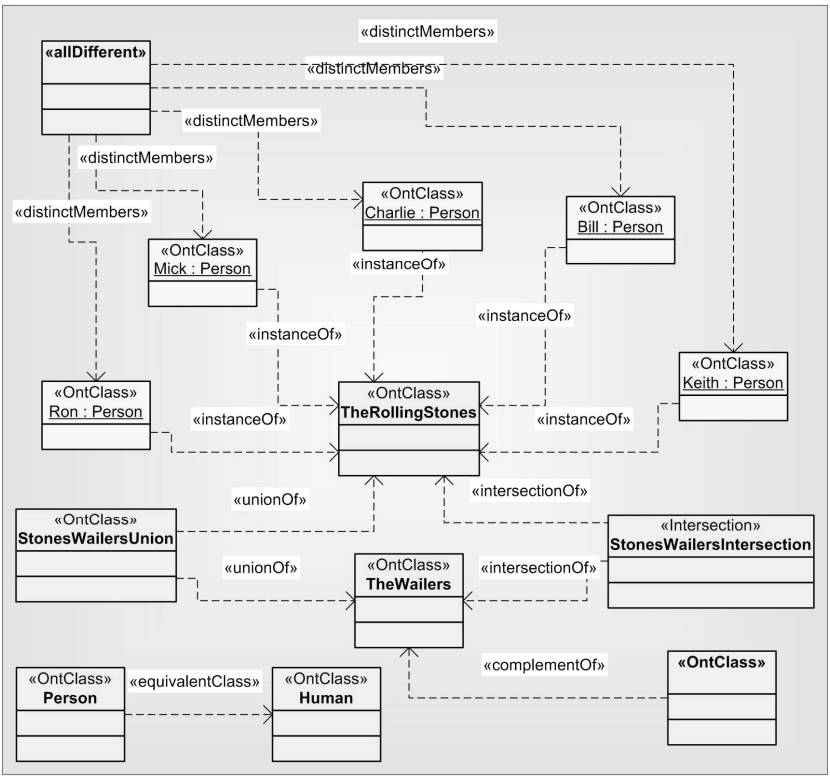

Fig. 9.5 Class diagram showing relations between ontology classes and individuals in the Ontology UML Profile

Properties (Fig. 9.6):

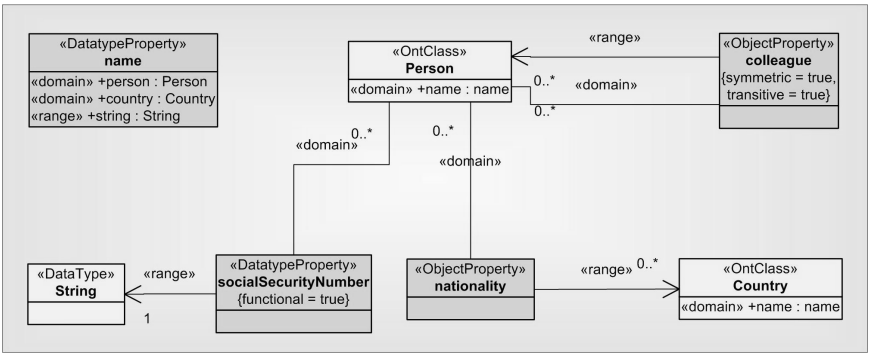

Fig. 9.6 Ontology properties shown in UML Class Diagram

Statements (Fig. 9.7):

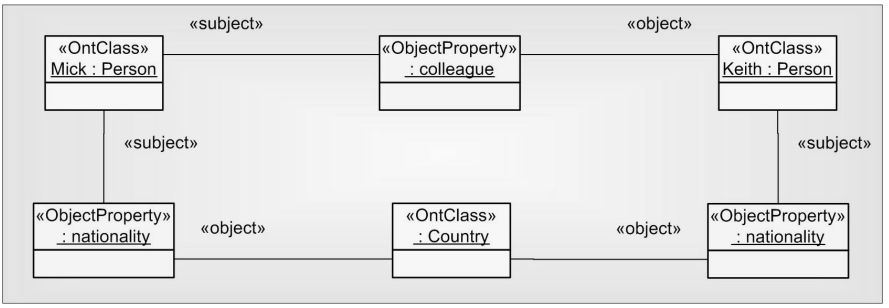

Fig. 9.7 Individuals and statements shown in a UML Object diagram

Of course, when the specification has been completed, you should look at the specification document for the exact details. Some of these details, especially the most important ones, will probably be the same or almost the same as those we have described in this chapter. However, there might be many less important details that are a little different. Something that is important, however, is that you can start from the examples that we have shown you and very quickly catch up with the specification.

http://www.springer.com/978-3-642-00281-6

Model Driven Engineering and Ontology Development
Gaševic, D.; Djuric, D.; Devedžic, V.
2009, XXI, 378 p. 183 illus., Hardcover
ISBN: 978-3-642-00281-6