Software Engineering Data Modeling Based on OWL

Donghu Gu, Hongyu Li, and Jinsheng Zhang

Yunnan Technology and Business University, Kunming, Yunnan 6517001, China

m13577050829@163.com

Keywords: Ontology Primitives; Ontology Element Attributes; Data Modeling Engineering

Abstract: In ontology modeling, because the concept classification structure is not clear and lacks theoretical guidance, the ontology modeling method lacks the ability to customize the specific ontology description language. In view of this situation, this paper combines ontology based theory with UML ontology commitment, and proposes a core ontology metamodel based on ontology based theory and UML metamodel extension and its extension method. UML's profile extension technology is used to eliminate OWL (Web Ontolo gy). Language) The key technology of the contradiction between complex symbology and ontology modeling operability requirements, the mapping relationship between UML modeling elements and OWL Lite syntax elements is established, and the Web ontology description language is taken as an example to the core ontology. The scalability of the metamodel and the effectiveness of the extension method were verified.

1. Introduction

OWL (Web Ontology Language) is a W3C-recommended Web Ontology Description Language Standard, a semantic markup language provided for publishing and sharing ontology on the WWW. OWL developed from DAML+p IL, the basic syntax and function are very similar, but its language mechanism is greatly enhanced. [1]There are more vocabulary used to describe classes and attributes, with rich semantic and relational logic representation capabilities. In addition, it emphasizes the representation of knowledge and the application of inference rules, and is the most representative ontology description language in current Semantic Web research.

OWL provides a large number of semantic primitives based on description logic to describe and construct various ontology, and has more mechanisms than XML, RDF, etc. Three sub-language with increasing expressiveness are provided for different needs: OWLLite; OWL DL and OWL Full

(1) OWL Lite: The semantic expression ability is simple, and is limited to hierarchical classification of concepts and simple attribute constraint description. Used for users who only need one classification level and simple attribute constraints.

(2) OWL DL: Based on description logic, supports users who need to perform maximum semantic representation on the inference system. The inference system guarantees computational completeness (ie, all conclusions are guaranteed to be calculated) and deterministic (ie, all calculations are completed in a limited amount of time). [2]Constraints are widely expressed and express all constraints of the OWL language, but only under specific constraints.

(3) OWL Full: Grammatical freedom, support for users who need to maximize expression on RDF without computational guarantee. Allowing an ontology to add vocabulary to a predefined (RDF, OWL) vocabulary, but without the support of an inference system, any inference software cannot support all of the semantic features of OWL Full.

The relationship between these three sub-language is: each legal OWL Lite is a legal OWL DL; each legal OWL DL is a legal OWL Full; each valid OWL Lite conclusion is a valid OWL DL conclusion; each valid OWL DL conclusion is a valid OWL Full conclusion.

When constructing the OWL ontology, users can choose different sub-language according to the requirements of expressiveness and complexity. When choosing a sub-language, the following should be considered:

(1) The degree of expression of the constraint. OWL Lite is suitable for simple constraints, and

OWL DL constraints have a wide range of expressions.

(2) Inheritance of the RDF model mechanism. OWL DL does not allow you to define the type of the type and the mechanism for assigning attributes to the type, and OWL Full implements this in this regard.

(3) OWLFuII's support for reasoning is unpredictable. Because the implementation of OWL Full's reasoning software is not fully supported, the computability is not guaranteed, so it is difficult to apply in reasoning applications.

2. Ontology Basic Theory

Ontology primitives and ontology meta-attributes are two important achievements of the ontology basic theory. The former reflects the ontological concept modeling ability and the essence of form. The latter is the attribute of ontology attribute and an important supplement to the ontology primitive.

2.1. Ontology primitives

BWW (Bunge-Wang-Weber) [5] proposes highly abstract primitives describing concepts. The ontology analysis framework of these primitives is based on Aristotle's ontology analysis theory, which is scientific and universal, and in computer science and The field of information has been fully affirmed [6]. BWW contains the following main primitives:

2.1.1. Things

Any objective existence can be regarded as a thing, and a thing can be a concrete thing or a conceptual one. There is no concept of "individual or instance" in the ontology primitive, and the object primitive itself can represent abstract concepts and instances in the class.

2.1.2. Class (concept)

A collection of things with the same characteristics.

2.1.3. Features and attributes

The feature is an objective existence, and the feature always exists regardless of whether the subjectively aware of the feature of the thing. The observation model of things is called a conceptual thing, and its characteristics are called attributes [7]. Attributes are classified into intrinsic attributes and relational attributes. The former relies on only one thing, the latter depends on multiple things, and the attribute primitives are used to express relationships between concepts.

2.1.4. Rules

The constraint on the value of the attribute and its combination is to limit the attributes of the thing and the relationship between things. BWW also includes primitives such as functions, interactions, and combinations. Bung suggests function primitives as an optional representation of attributes [7]. Interaction is a relationship based on associated attributes. Combination is a kind of thing. Special form of existence. Therefore, this paper believes that the basic ontology modeling requirements can be met through the above core primitives.

2.2. Comparison of Ontology Description Language Features

The above ontology description language is analyzed and compared according to different characteristics of the ontology description language. Comparison column

Table is shown in Table 1.

Table 1 Comparison of Ontology Description Language

Language	Expressive power	Reasoning complexity	Remarks
RDF / RDFS	weak	Very low	Establish a basic definition of ontology
DAML+ OIL	Strong	high	Strong expressive ability and high degree of inference complexity
OWL Lite	general	low	OWL's minimal version, effectively controlling complexity
OWL DL	Strong	high	Increase expressiveness within the scope of reasoning
OWL Full	Very strong	Uncontrollable	Provides strong expressiveness regardless of the limitations of reasoning

3. Ontology Core Metamodel and Extension Method

In this paper, the main ontology primitive is regarded as the core structure of COMM, and the ontology meta-property is used as the attribute of the primitive primitive, and the primitive of the class is constrained. Figure 1 shows the COMM heavy UML extension metamodel.

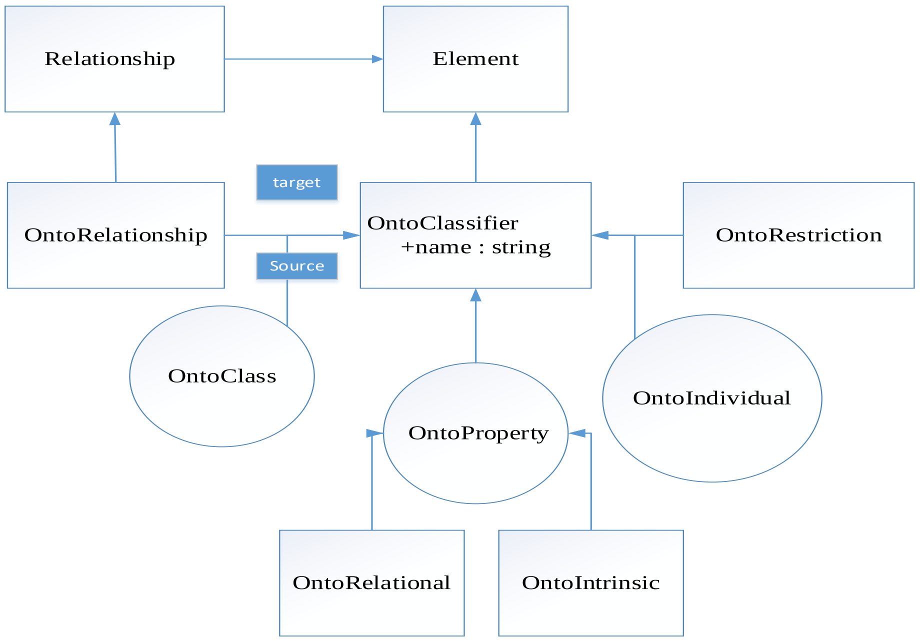

Figure. 1 Comm core metamodel Figure 1

OntoClassifier and OntoRelationship inherit UML top-level classes to mask attributes and associations related to OO modeling. OntoClass can have multiple internal properties and has a name attribute as an identifier. OntoRelationship has a target and a souce association, indicating the connection relationship between OntoClassifiers. OntoClass, OntoRestriction, OntoProperty, OntoIndividual correspond to classes, rules, attributes, and instances in ontology primitives, and OntoRestriction and OntoPropety are COMM first-level elements. OntoIntrinsic and OntoRelational correspond to the intrinsic properties and associated properties of ontology primitives, respectively. OntoMetaProperty describes the ontology meta-attribute whose type of the kind property is defined by the enum class MetaPropertyKind.[4] The OntoIntrisic type is described by OntoType, and there is a class one-way association between OntoRelational and OntoClass to indicate the type of the associated attribute. According to the above expansion results, it can be seen that COMM has universal ontology modeling ability and incorporates ontology element attributes into COMM, which can provide guidance for conceptual modeling and further standardize the organization of conceptual structure in ontology. In order to make COMM have the ability to customize a specific ontology language, this article provides an extension method, in which the customized ontology language is Specific Ontology Language (SOL). The COMM extension method is as follows:

3.1. Analysis of SOL core structure

If the SOL construct has a correspondence with the core primitive, it is described using the COMM metamodel.

3.2. Perform SOL ontology analysis

The SOL itself is called the ontology, which is called the language ontology. Corresponding to the concept construct in OL and the class primitive, it is called the language ontology class; the structure representing the relationship between the ontology classes in OL corresponds to the attribute primitive, which is called the language ontology attribute. By using ontology of primitives and attribute primitives to perform ontology analysis on OL, you can obtain the essence of SOL providing ontology modeling form and modeling ability.

3.3. Extended COMM

UML is extended to extend COMM to further meet the need for ontology modeling using SOL.

3.4. Constraint extended semantics

Use OCL to constrain the extended semantics of COMM as needed.

4. Example Analysis

To verify the effectiveness of COMM customization capabilities and extension methods, this section takes OWL as an example and follows the COMM extension method to complete OWL customization. OWL is the Web ontology description language officially launched by W3C based on the development experience of DAML-ONE, OIL, DAML+OIL. Therefore, using OWL as an analysis example is typical. This article does not distinguish between the different constructs in the 3 seed languages of the OWL specification[5].

4.1. Analysis of the OWL core structure

Table 2 shows the correspondence between ontology primitives, COMM, and OWL constructs.

Table 2 Comparison of ontology primitives and OWL word formation

Ontological primitive	OWL word formation	COMM element
Object	Individual	OntoIndividual
Characteristic	-	-
Intrinsic property	DataProperty	OntoIntrinsic
Associated attribute	ObjectProperty	OntoRelational
Class	Class	OntoClass
rule	Restriction	OntoRelational

As can be seen from Table 2, OWL is an ontology description language with ontology completeness. Individual, Class, DataProperty, ObjectProperty, and Restriction are their core ontology constructs. There is no "individual" concept in ontology primitives. Thing in OWL is the parent class of all classes, which is different from the meaning of the primitives of things. Therefore, it is more appropriate to correspond the primitives of things to individual. In the OWL customization process of COMM, the corresponding structure of OWL will be described using the COMM metamodel elements corresponding to the ontology primitives in Table 1.

4.2. OWL ontology analysis

Think of OWL as the language ontology, called the OWL language ontology. Individual, Class, DataProperty, ObjectProperty, Restriction are the language ontology classes of the language ontology, representing instances, classes, attributes, and constraints, respectively.

This is consistent with the analysis in the OWL core construct. minCardinality, maxCardinality, cardinality, hasValue, AllValuesFrome, SomeValuesFrom can be considered a special subclass of Restriction. OWL's built-in "Property" such as inverseof, onProperty, is used to describe the relationship between OWL language ontology classes. For example, invserof is used to represent the relationship between objectProperty, onProperty is used to represent the relationship between restriction and property and restriction and class, so OWL language ontology attributes include:

1) Subclassof, equivelantClass, disjointWith, UnionOf, complementOf, intersectionOf, one of, indicating the horizontal or vertical relationship between classes [3].

2) sameAs, differentFrom, describes the relationship between individuals.

3) subPropertyof, equivelantProperty, inversOf, description attribute

5. Conclusions

This paper combines ontology basic theory with UML ontology commitment, designs COMM based on ontology basic theory, and proposes COMM extension method. Take OWL as an example to verify and analyze the effectiveness of COMM scalability and extension methods. The example analysis shows that COMM has better scalability and its extension method can effectively customize the main structure of OWL.

References

[1] B. Li, S. Ji and D. Qiu, et al. GENERATING TEST CASES OF COMPOSITE SERVICES BASED ON OWL-S AND EH-CPN[J]. International Journal of Software Engineering and Knowledge Engineering, 2010, 20(07):921-941.

[2] R. Liu, C. Hu and C. Zhao. 2008 Ninth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing - Model Checking for Web Service Flow Based on Annotated OWL-S[J]. 2008:741-746.

[3] Hilaire V, Gaud N, Galland, Stéphane, et al. Proceedings of the 2010 ACM Symposium on Applied Computing - SAC \"10 - An approach based upon OWL-S for method fragments documentation and selection[J]. 2010:938.

[4] Bruijn J D, Rubén Lara, Polleres A , et al. OWL DL vs. OWL Flight: Conceptual Modeling and Reasoning for the Semantic Web[J]. Fensel Dieter, 2005:623-632.

[5] Sheng B, Zhang C, Yin X, et al. Common intelligent semantic matching engines of cloud manufacturing service based on OWL-S[J]. The International Journal of Advanced Manufacturing Technology, 2016, 84(1-4):103-118.

[6] Babu S, Lakshmi N V S S R, Sivakumar B. An analysis of decision – making structure for self-organizing system based on software engineering[J]. Computers & Electrical Engineering, 2017, 57:81-90.

[7] Živadin Micić, Marija Blagojević, Živadin Micić, et al. Knowledge acquisition in information technology and software engineering towards excellence of information systems based on the standardisation platform[J]. Computer Standards & Interfaces, 2016, 44:1-17.

[8] Igamberdiev M, Grossmann G, Selway M, et al. An Integrated Multi-Level Modeling Approach for Industrial Scale Data Interoperability[J]. Software & Systems Modeling, 2016:1-26.