Proceedings of the 6th Győr Symposium and 3rd Hungarian-Polish
and 1st Hungarian-Romanian Joint Conference on Computational Intelligence

# Uncertainty Models in Ontology Management

**L. Kovács**

**University Miskolc, Department of Information Technology**
**Miskolc-Egyetemvaros, 3515, Hungary**
**E-mail: kovacs@iit.uni-miskolc.hu**

Abstract: The knowledge base is the kernel module of any intelligent information systems. The main tool in knowledge representation is ontology which is used to perform an explicit conceptualization of the problem domain. A unique characteristic in modelling of agent's knowledge is the representation of subjective knowledge. This model may contain some uncertainty factors. This paper provides a survey on the different uncertainty models and presents a new vector-valued representation formalism for truth values. The proposed model is tested in an implementation developed in the OWL ontology language.

*Keywords: ontology, uncertainty representations, many-valued logic*

## 1. Introduction

The term 'ontology' is the science on structures of objects, properties and relations in every area of reality. According to a widely accepted definition for ontology in information science, "an ontology is a formal, explicit specification of a shared conceptualization " [1]. In this context, conceptualization refers to an abstract model of some phenomenon in the world that identifies that phenomenon's relevant concepts. These properties mean that the concepts and the constraints on their usage are unambiguously defined and the ontology description is machine understandable. The core element of ontology models is the term ‘concept’. This term has no exact global definition; every discipline has developed its own definition of concept. From the viewpoint of engineering, the term 'concept' is used as an identifier or a descriptor for a cluster of objects. In this sense, the concept describes besides the naming also the properties of the cluster. In the history of knowledge engineering, a great variety of data structure were developed to represent the meaning of concepts. Nowadays, these models exist parallel and are used for different purposes.

Description Logics (DL) are considered the most important knowledge representation formalism for ontology unifying and giving a logical basis to the well known traditions of frame-based systems, semantic networks and KL-ONE-like languages, object-oriented representations, semantic data models, and type systems. The main advantages of using DL are the automatic classification, logical inference and consistency checking. It provides the most general approach to knowledge representation. This representation form provides a precise inference framework for reasoning purposes. Hayes [2] has proven first the equivalence between the framework representation and the first order logic formalism.

Based on the researches of Brachman and Levesque [3] the reasoning in the frame and network representations can be accomplished without the full power of the first order logic. The DL provides a formal language for defining concepts and individuals [4]. A concept is an intentional description of a class of individuals. The binary relationships between the concepts are called roles. In DL, the basic logic operators are beside the general disjunction, conjunction and complement operators the inclusion (subsumption), classification and recognition. Concept C1 subsumes concept C2 when every instance of C2 is also an instance of C1. The classification integrates the concept into a taxonomy. The recognition determines the set of most specific concepts which an individual instantiates. There are some additional operators used in the extension languages like the union, role value restriction, role existence restriction, role number restriction, role transitivity, role hierarchy and inverse role. The value restriction constraints the range of role relationship. The standard approach in DL is the open-terminology assumption which presumes an incomplete terminology.

## 2. Representation forms of uncertainty

The measured values are usually single strict values, but in some domains the values are more complex. The complexity may have many reasons, like multivalued nature or uncertainty in the measurement. The typical categorizations of fuzziness use the following reasons:

- unknown values (not defined value, missing value,..)
- many-valued concept
- aggregation for a container concept.

Regarding the unknown value, there are two main approaches for the representation. The simplest way is the select a value from the domain to denote the unknown values. This solution does not need an extension of the usual value management module, but it is just an ad-hoc solution and this method is not portable to any different applications. The second way is to extend the internal value management system with a new state variable. This variable is set if the corresponding values is unknown. The standard SQL language uses this kind of approach, the NULL value corresponds to the unknown state.

Regarding the management of fuzziness in general, we can define some dimensions of fuzziness that should be considered in selection of the appropriate value representation form. The main dimensions are

- the value is known/unknown
- there is a single valued/many-valued measure
- there is a single measure/multi measures
- the frequency weights of the values.

For example, a crisp value 10 belongs to the coordinate (known value, single valued, single measure, 1.0).

In the case of (known and unknown values, many-valued, multi measures) variables, the exact description uses a list of function formalism. For discrete variables, the

$$(U/f_0, v_1/f_1, v_2/f_2, ..., v_n/f_n)$$

vector is used where U denotes the unknown state; v: value from the domain; f: frequency weight of the values. Here

$$\sum_{i=0} f_i = 1$$

If the variable has a continues value domain than we use a frequency function on the domain D:

$$\int_D f(v)dv \leq 1$$

The weight for the unknown state is equal to

$$f(U) = 1 - \int_D f(v)dv.$$

In many approaches, the long vector or function description is replaced with a compact, aggregated value. The usual aggregation operator is the average value:

$$\bar{f} = \int_D vf(v)dv$$

Also the fuzzy theory uses this approach implicitly. This operation can be defined only for numeric variables. In the case of categorical variables, the selection of median is a usual aggregation operator. The main drawback of the approach is that is cannot precisely express the value distribution and the weight of the unknown value.

## 3. Logic models in ontology

The classic approach in knowledge representation is the application of binary valued logic. The True (1) and False (0) values give an efficient representation of the truth values. The crisp binary truth value is also a fundamental part of membership grade representation in the traditional conceptual modelling. Considering the classic ontology models [5] , the relationship between concept *a* and an another concept or value (as instance concept) *b* is a strict relationship: *aRb* is met or not. The methodology of classic FCA (formal concept analysis) uses the same approach [6]. The context of the FCA is based also on a strict relationship between objects and attributes. This context can be represented also with a binary values attribute-set: the attribute *a* at object *o* is 1, if *oRa* is met, otherwise the attribute *a* has a value 0. Thus every concept *c* can be uniquely represented with an attribute set, the intention part of the concept, where this set is a crisp set. The main benefit of FCA model is that it provides an efficient framework for concept generalization. The generalization *g* of a concept *c* has an intention set which is subset of the intention set for *c*. Thus only the existence of some common properties are the key factors in the concept generalization process. Considering, some others classic software engineering design models, like ER or UML, although the attributes have a multi-valued domain, the existence of an attribute at an entity is a strict binary valued parameter.

The binary valued logic in modelling of the open objective world is accurate tool, as the law of excluded third is accepted as a general rule of our objective world. Thus *p* or $\neg p$ is true for every proposition *p*. The inaccuracy of this binary valued logic was detected when the modelling was extended to subjective knowledge bases. The term 'subjective ontology' or 'epistemology' [8] refers to ontology models which were created within a subject (agent). The subjective ontology is the conceptual model generated by a cognitive process. During the learning process, the input from the environment is used to update the conceptual model.

Considering a training set and a relation *R*, then if every transaction in *O* contains *oRa*, then the subjective ontology can contain also *oRa*. On the other hand if in all of the training transactions *aRb* is met, then the model should contain $\neg$(*oRa*). An interesting case is when some transaction contains *oRa* while others contain *oRb*. A very different case is when the transaction does not contain information on *aR*. For these models, the simple binary valued attribute is not accurate enough and the logic model was extended to different directions.

The main approach is to use multi-valued logic. There are many different approaches to implement the multi-valued logic. The simplest case is three-valued case, the three truth values are true, false and unknown. This model is used for example in SQL standard too. The Lukasiewicz, Tarski, Post, Gödel [7] investigated the different extension of two-valued logic where some internal structure is assumed among the different truth values. A simplified model is among other the information system model [11] where the truth values are given with a simple enumeration.

## 4. Negative property

The presented logic models can be considered as scalar logics as the truth value can be represented with a real number. The truth value denotes only the existence of some phenomena, i.e. usually the existence of some property. The not existence is usually considered as a derived property, a function of the existence property, similar to the other compound property expressions. On the other hand, some authors like Meixner [9] argues for the existence of negative property. Another pioneer work on considering negative properties relates to Russell [10], who introduced these concepts to solve the paradox of Demo. A negative property $p^-$ has always a positive counterpart property $p^+$ and it can be defined here as

$$p^-(x) \leftrightarrow \neg p^+(x).$$

One of the applications of negative property in knowledge engineering technologies relates to the work of Kourie[8]. The knowledge database is constructed from a context similar to the context in Formal Concept Analysis. The context contains only those properties which are relevant in the problem domain. Kourie showed that if *p* is a relevant positive property than its negative counterpart is also a relevant property. To manage uncertainty, a three-valued logic is proposed in logic representation. The knowledge model consists of two base relations among the concepts: abstraction and its inverse, the refinement (<) relation. The set of properties that are true for concept *C* is denoted by *true(C)*. The Kourie assumption states the following for any two concepts $C_1, C_2$:

$$C_1 < C_2 \rightarrow$$
$$true(C_1) \supseteq true(C_2), \forall p \in true(C_1) \setminus true(C_2) : \exists p' \in true(C_2) : p < p'$$

This means, that every sub-concept must have at least one such characteristic property which does not met at the super-concept, but there exists a generalized property that is true for the super-concept

## 5. Introduction of a vector-valued logic representation

In the proposed logic representation model, the lossless representation formalism was used for the extension of the base tree-valued logic. Thus the domain of the truth values are {T,F,U}. The truth values for the compound measures are given by

$$(f_T, f_F) : \quad f_T + f_F \leq 1, \; f_T, f_F \geq 0$$

The domain of possible truth values is a rectangle shown in Figure 1.

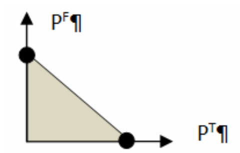


*Figure 1. Domain of vector truth values*

The meaning of some cornel values are shown here to see the correspondence with the usual representation models:

(1,0) : T in fuzzy and Boolean model

(0,1) : T in fuzzy and Boolean model

(0,0) : unknown value.

The line segment between (0,1) and (1,0) corresponds to the Fuzzy truth domain.

The main benefit of the proposed model is that it preserves more information on the component values and it provides an explicit representation of the unknown values too. Considering the implementation of the model in ontology, the current OWL language does not support the direct implementation. The OWL works with the DL logic language which is based the classic Boolean truth representation. In the OWL, the construction of an intermediate class should be used to model an n-ary relation. The basic elements to implement an additional probability descriptor are the followings.

```
<owl:ObjectProperty rdf:ID="has_age">
  <rdfs:range rdf:resource="#Age_Relation"/>
</owl:ObjectProperty>
<owl:Class rdf:ID="Age_Relation">
    <rdfs:subClassOf> <owl:Restriction>
    <owl:someValuesFrom rdf:resource="#Probability_value "/>
    <owl:onProperty>
      <owl:FunctionalProperty rdf:about=" #diagnosis_probability"/>
    </owl:onProperty>
   </owl:Restriction> </rdfs:subClassOf>
</owl:Class>
```

With the help of the presented structures, the vector-valued logic can be implemented in the knowledge management systems on a straightforward way.

## References

[1] Gruber, T.: *A Translation Approach to Portable Ontology Specifications*, Knowledge Acquisition, 5(2):199-220, 1993.
[2] P. Hayes. *The logic of frames*. In Metzing, editor, Frame Conceptions and Text Understanding. de Gruyter, Berlin, 1979.
[3] Brachman R. J. , Levesque. H.J. :*The tractability of subsumption in frame-based description languages*. In AAAI-84, pages 34--37, 1984
[4] W. Woods and J. Schmolze. *The KL-ONE Family*. Computer and Mathematics with Applications, 23(2 5), 1992.
[5] Krötzsch M, Simancik F, Horrocks I*: A Description Logic,*. IEEE Intelligent Systems 29(1): 12-19 (2014)
[6] Ganter, B., Wille, R.: *Formal Concept Analysis: Mathematical Foundations*, Springer-Verlag, Berlin, ISBN 3-540-62771-5, 1998
[7] Bergmann, M.: *An introduction to many-valued and fuzzy logic*, Cambridge University Press, 2008
[8] Gruner, S.: *Abstraction, Refinement, Enrichment*, Formal Aspects of Computing, Shaker Verlag, 2013
[9] Meixner., U.: *On negative and disjunctive properties*, Language, Truth and Ontology, Kluwer Publisher, 1992
[10] Oaklander, N., Miracchi, S.: Russell, *Negative Facts and Ontology*, Philosophy of Science, pp. 434-455, 1980
[11] Orlowska, E., Pawlak, Z.: *Representation of nondeterministic information*, Theoretical Computer Science, 29, pp.27-39 (1984)