SEMANTIC KNOWLEDGE MODELS OF NON-CRISP KNOWLEDGE

*Tatyana Ivanova *, Petya Petkova*

Technical University of Sofia, Technical College of Sofia, Botevgrad
Bulgaria

* Corresponding Author, e-mail: t.ivanova@tu-sofia.bg

Abstract: Many practical applications, such as medical diagnosis, business decision-making, information searching and retrieval, etc., require usage of uncertain or ambiguous knowledge. Classical ontology-based technologies can represent and reasoning only with cri Several fuzzy or probabilistic extensions of classical description logics and languages for semantic knowledge representation have been proposed recently, but high reasoning complexity of its decision procedures make difficult its usage in real applications. It is of great importance to select the knowledge representation technology, ensuring both the needed expressiveness and effective reasoning. In this paper we make short analysis of knowledge representation and reasoning capabilities of description logics and ontology representation languages allowing representation of uncertain knowledge and propose a methodology for selecting the best description logic and ontology representation variant for every practical application.

Key words: probabilistic description logic, fuzzy description logic, ontology, methodology.

1.INTRODUCTION

Description logic (DL) - based ontologies represent crisp knowledge, but in many real-world applications, some concepts are fuzzy. For example, in a healthcare domain, “high temperature”, "sick" , and many others can be subject to degrees of uncertainty or gradual change. And positive or negative results of medical tests usually have some probability degree. Fuzzy Knowledge can be modelled in ontologies by adding elements of fuzzy logic in description logic. Fuzzy Ontologies are used in applications that can handle vagueness and imprecision in data and reasoning. Fuzzy ontologies use fuzzy sets to allow for concepts to have degrees of membership, improving representation.

Probabilistic extensions of description logic are used to handle uncertainty. As uncertainty and vagueness are fundamental components of many real domains, various probabilistic or fuzzy extensions of DLs have been proposed during the last few years. The main drawback of probabilistic and fuzzy DLs is high computational complexity of reasoning. In this research we make a brief analysis of crisp and extended DLs to find main features, causing higher reasoning complexity and propose a methodology for choosing appropriate logical formalisms for modelling knowledge in real domains.

More clearly, goals of this research are:

- To find which description logic axioms are main sources of high computational complexity of reasoning tacks.
- To analyse and systematize reasoning capabilities of description logics capable for representing non-crisp knowledge, including fuzzy and probabilistic extensions of description logics.
- To discuss how exactly adding of non-crisp knowledge representation elements affect complexity of reasoning ways to reduce the logical complexity of reasoning.
- To propose a methodology for choosing appropriate logical formalisms and tools for development of ontological models in practical domains that use vague knowledge.

2. SURVEY OF RESEARCH IN THE FIELD

Description logics (DLs) are the main logical ground of development of knowledge models, needed for most artificial intelligence applications. DLs are a family of formal knowledge representation languages that allow for reasoning about concepts and relationships in domains. For finding which description logic axioms are main sources of high computational complexity of reasoning tacks we first analysed and systematized axioms and reasoning capabilities of valuable variants of description logics family (see table 1). Then we analysed DL extensions capable for representing non-crisp knowledge, including fuzzy and probabilistic extensions of description logics. $\mathcal{EL}$ (A Lightweight Fragment), $\mathcal{ALC}$ (Attributive Language with Complements), $\mathcal{SHIQ}$ (Role Hierarchies, Inverse Roles, and Qualified Existential Quantification), and $\mathcal{SRIQ}$ (for more complex ontologies) are more frequently used sublanguages in DL family. Choosing appropriate description logic for knowledge modeling depends on the specifics of modeling domain and requirements of applications that will use these knowledge models. Main requirements are related to the level of expressiveness and needs to use uncertain knowledge.

Table 1. Features of popular DL logics and OWL languages

Feature \ logic	$\mathcal{EL}$	$\mathcal{ALC}$	$\mathcal{SHIQN}$	$\mathcal{SHIQ}$	***DL-Lite***	***OWL2 RL***	***OWL2 QL***	***OWL-DL***
(A ⊓ B)	+	+	+	+	+	+	+	+
(A ⊔ B)	-	+	+	+	-	-	-	+
Complex ¬A	-	+	+	+	-	-	-	+
A ⊑ B	+	+	+	+	+	+	+	+
Equivalence (≡)	+	+	+	+	+	+	+	+
Disjointness (⊥)	*Restr.*	+	+	+	+	*Restr.*	-	+
(∀R.A)	*Only in TBox*	+	+	+	*Restr.*	*Only in TBox*	*Only in TBox*	+
(∃R.A)	+	+	+	+	*Only in TBox*	*Only in TBox*	+	+
(R⊑ S)	+	-	+	+	+	+	*Simple*	+
Inverse Roles (R^{-1})	-	-	+	+	+	*Restr.*	*Restr.*	+
Functional Roles			+	+		-	-	+
Inverse-Functional Roles	-	-	+	+	+	-	+	+

Transitive Roles R^+	**-**	**-**	+	+	+	+	-	+
Symmetric Roles	**-**	**-**	+	+	-	+	-	+
Complex Role Inclusion Axioms	**-**	**-**	+	+	-	*Restr.*	-	+
Nominals ({a, b})	**-**	**-**	+	+	-	*Restr.*	-	+
Cardinality Constraints (nR.An R.AnR.A)	**-**	**-**	+	*Qualified*	-	-	-	+
Some Role Restrictions	*Role Disjointness*	*No Role chains*			*No Role chains*		*No Role chains*	
Individuals in TBox	**-**	**-**	+	+	-	-	-	+

DL-Lite, a lightweight Description Logics (DLs) language, is designed for efficient reasoning and query answering, particularly in relational database settings. OWL 2 sublanguages are shortly described in table 2. OWL 2 QL is based on DL-Lite.

Table 2. OWL 2 languages - Features and complexity of reasoning tacks

OWL 2 Profile	***Description Logic***	***Reasoning Complexity***
OWL 2 EL	$\mathcal{EL}$ ++	*Polynomial time (PTime) for standard reasoning tasks. Suitable for large ontologies with simple class hierarchies.*
OWL 2 QL	*DL-Lite_R*	*Logarithmic space for query answering. Optimized for applications requiring efficient query processing.*
OWL 2 RL	*Description Logic Programs (DLP)*	*Polynomial time (PTime) for rule-based reasoning. Designed for applications leveraging rule-based inference engines.*

As description logics having possibilities to represent non-crisp knowledge subsume some properties from corresponding crisp logics, we first made complexity analysis of reasoning on crisp ontologies, and then we discuss possibilities for modelling fuzzy or probabilistic knowledge and complexity of reasoning in non-crisp models.

2.1. Complexity of reasoning in crisp ontologies (see table 3)

Table 3. Complexity of Satisfiability, Subsumption reasoning tacks

Description Logic	***Expressiveness***	***Time Complexity***	***Space Complexity***
$\mathcal{ALC}$	*Medium*	*ExpTime-complete*	*PSpace-complete*
$\mathcal{ALCH}$	*Medium*	*ExpTime-complete*	*PSpace-complete*
$\mathcal{SRI}\vartheta\mathcal{Q}$ *(OWL 2 DL)*	*High*	*ExpTime-complete*	*PSpace-complete*
$\mathcal{SHIQ}$	*High*	*ExpTime-complete*	*PSpace-complete*
$\mathcal{EL}$	*Low*	*PTime-complete*	*PTime-complete*
DL-lite	*Low*	*PTime-complete*	*PTime-complete*
$\mathcal{SHI}\vartheta\mathcal{Q}$	*Very High*	*NExpTime-complete*	*Exponential*

Practical usability of description logic variants is tightly related to presence of corresponding ontology development languages and effective reasoning engine.

Table 4. Features and complexity frequently used DLs

Description Logic (DL) variant	*Features*	*Data Complexity*	*Combined Complexity*
DL-Lite	*Lightweight, optimized for databases, no full negation*	*LOGSPACE*	*PTIME*
$\mathcal{RL}$, $\mathcal{EL}+$	*Efficient for large ontologies, supports conjunction and existential restrictions*	*PTIME*	*PTIME*
$\mathcal{ALC}$	*Allows negation, disjunction, and full existential quantification*	*NP-complete*	*PSPACE-complete*
$\mathcal{SHIQ}$	*Adds transitive roles, inverse roles, number restrictions*	*NP-hard*	*EXPTIME-complete*
$\mathcal{SRIQ}$ *(OWL 2 DL)*	*Highly expressive (role hierarchies, qualified number restrictions)*	*NP-hard*	*2EXPTIME-complete*

2.2. Probabilistic ontologies

There are two main types of non-crisp knowledge: probabilistic and fuzzy knowledge, representing two different approaches to handling uncertainty. Fuzzy knowledge deals with imprecision and vagueness. Imprecise information is represented in degrees rather than binary true/false values. Fuzzy knowledge allows for partial truths. Probabilistic knowledge represents uncertainty in terms of the likelihood of events occurring.

2.2.1. Probabilistic description logics

Probabilistic Description Logics (PDLs) extend classical DLs by integrating probability theory, allowing for reasoning under uncertainty. Probabilistic semantics often is modelled by distributions over possible worlds or by probabilistic annotations on axioms. For example, a concept may have a probability distribution over its instances or relations might have certain probabilities attached. In probabilistic DL, the entailment task seeks to find the probability that one statement follows from others under certain probabilistic constraints. **Consistency Checking** in PDLs involves verifying whether there exists a model where the probabilistic constraints hold true. This can include checking if a certain class have a non-zero probability or have probability greater than some value.

Model checking involves verifying whether a given model (e.g., a probabilistic interpretation of a knowledge base) satisfies a set of constraints. This may include checking whether a particular class membership has a sufficiently high probability.

Classification tack in PDL includes determining the most likely class for an individual or a set of individuals based on their probabilistic relationships. **Instance checking with probabilities** include determining whether an individual satisfies a probabilistic concept. **Query Answering** involves determining the probability of certain facts or relations in probabilistic knowledge base. Probabilistic query example: "What is the probability that a certain individual is an instance of a class?". Probabilistic extensions of standard DL reasoning tacks add complexity of calculating probabilities associated with them and complicates reasoning, making some tasks more computationally intensive than in classical description logic. Inference in probabilistic

DL is more complex than tableau-based reasoning in crisp DLs due to the need of specialized probabilistic inference algorithms, such as belief propagation, Markov Chain Monte Carlo (MCMC), or approximate inference techniques.

Practically valuable Probabilistic Extensions of DLs include: Probabilistic $\mathcal{ALC}$ ($\mathcal{P}$-$\mathcal{ALC}$) and Probabilistic DL-Lite. In probabilistic DL-Lite [1] DL-Lite axioms are associated with probabilities.This extension allows for querying knowledge bases with probabilistic information and maintain tractable reasoning incorporating uncertainty.

Probabilistic $\mathcal{ALC}$ ($\mathcal{P}$-$\mathcal{ALC}$) is a probabilistic extension of the DL sublanguage $\mathcal{ALC}$. $\mathcal{P}$-$\mathcal{ALC}$ handles probabilistic reasoning while still maintaining a balance between expressiveness and complexity. It can be used in situations requiring detailed knowledge about the relations between concepts, such as in medical diagnosis.

Other probabilistic DL, $\mathcal{BEL}$ [2] is designed to represent classical knowledge that depends on an uncertain context. The probability distribution of different contexts is expressed by a Bayesian network (BN). The probabilistic DL extension $\mathcal{BALC}$ combines $\mathcal{ALC}$ and BNs[3]. $\mathcal{BALC}$ axioms are required to hold only in some (possibly uncertain) contexts, which are expressed through annotations.

Probabilistic Description Logics model uncertainty using probability distributions over a set of variables, representing concepts and relationships as nodes, and the probabilistic dependencies between them as edges. Bayesian Description Logics are useful in decision support systems, medical diagnosis, and risk management, where uncertain knowledge and dependencies need to be modeled probabilistically.

2.2.2. Probabilistic ontology languages

Various probabilistic extensions of DLs have been proposed recently but most of them are only presented as mathematical theories. Only a few extensions are implemented in reasoning engines. PR-OWL (Probabilistic OWL) is a ontology representation language for reasoning in the presence of uncertainty. PR-OWL is an extension of OWL that integrates Bayesian networks and description logics [4]. It is useful in applications requiring both probabilistic inference and structured knowledge representation.

The BayesOWL [5] framework extends OWL capacities for modelling and reasoning with uncertainty. It applies a set of rules to transform the class hierarchy defined in an OWL ontology into a Bayesian network.

2.2.3. Probabilistic reasoning complexity and probabilistic reasoning engines

The main problem, related to the practical usage of probabilistic description logics is its high reasoning complexity. The complexity of reasoning tacks in particular probabilistic DL extension depends on the complexity of reasoning in the corresponding crisp DL, used probabilistic model and effectiveness of reasoning algorithms, built in reasoning engines. In almost all cases, adding probabilistic elements in description logic variants increase the complexity of reasoning (see table 4 for standard DL). This is because of the needs of additional calculations of the probability of certain conclusions during reasoning.

Table 4. Comparison of reasoning complexity in standard and probabilistic DL

Task	Complexity (Classical DL)	Complexity (Probabilistic DL)
Consistency checking	PSPACE	PSPACE (extended to probabilistic cases
Entailment	EXPTIME	EXPTIME (extends in probabilistic versions)
Probabilistic Satisfiability	NP (for standard DL)	PSPACE or EXPTIME (depends on the model)
Probabilistic Inference	-	#P-complete or worse
Axiom Subsumption	ExpTime (for DL)	PSPACE or EXPTIME
Classification	EXPTIME	EXPTIME or harder
Query Answering	PSPACE or EXPTIME	NP-complete or higher (probabilistic queries)

Annotation properties, or specific language extensions can be used, and probabilities can be assigned to different axioms. The practical implementations of reasoning algorithms, used for probabilistic inference in probabilistic reasoners also are of great importance.

TRILL [6] is a free probabilistic reasoning engine for DISPONTE KBs, implemented in Prolog. It uses backtrackings for handling the non-determinism of the tableau algorithm. An reasoning engine BUNDLE [7] is a probabilistic extension of well-known Pellet. It first uses Pellet to perform inference over crisp logic, discarding probability values, and then compute the probability of the query using probabilistic annotations of axioms. In fact, BUNDLE is a modular realization of an algorithm, working with probabilistic annotations of DL axioms. It can be also embedded in Hermit, Fact++ and JFact OWL reasoners. PR-OWL Decision [8] is a framework that supports description logic reasoning and probabilistic inference. It includes both a graphical user interface (GUI) and a reasoning engine developed to work with PR-OWL. It is useful for supporting decision-making based on probabilistic ontologies.

In large Markov Networks, exact inference can be computationally expensive, and approximate methods like Loopy Belief Propagation (LBP) or MCMC are often used.

2.3. Fuzzy knowledge inclusion in ontologies

2.3.1. Fuzzy description logics

Classical DLs can't be used for modelling incomplete, unclear or vague knowledge. Fuzzy Description Logics (FDL) extends classical Description Logics (DLs) by adding capabilities to represent unclear, vague (fuzzy) knowledge.

Fuzzy Description Logic introduces the idea of graded membership of instances in concepts, or degrees of truth of statements to handle vagueness more effectively. In FDL, the membership functions can have a range of values, typically between 0 and 1, indicating the degree to which an object belongs to a particular concept. This is useful for more nuanced reasoning, specific in many cases for real-world applications where categories are not sharply defined or can overlap each other. Fuzzy Description Logics are extremely useful in areas as Medical Diagnosis, Image and Speech Processing, Natural Language Processing. Fuzzy description logics can use one or more of the following fuzzy components: Fuzzy Sets and Membership Functions, Fuzzy Roles, Fuzzy Operators, fuzzy properties.

In FDL, concepts are described by fuzzy sets, where each individual has a membership degree in the set, less or equal to 1. Fuzzy Role Assertions can also have membership functions, which quantify the degree to which a relationship holds between two individuals. In this model, fuzzy quantifiers like "most," "some," "few," or "almost all" are applied. For example, a rule might assert that "most tall people are good basketball players," with the degree of truth (membership) being fuzzy rather than binary. Fuzzy Operators AND, OR, NOT: In the context of Fuzzy DL, T-norms and T-co-norms are generalizations of the standard conjunction (AND) and disjunction (OR) operators in fuzzy logic, used to handle the fuzzy intersection and union of concepts

Fuzzy logic includes new reasoning tacks: Fuzzy Classification, determining to what degree an individual belongs to a fuzzy class; Fuzzy querying, aiming to find individuals that satisfy fuzzy constraints. The choice of T-norm significantly influences both the decidability and complexity of reasoning tasks in FDLs. While Minimum and Product T-norms tend to maintain EXPTIME-complete complexity, other T-norms, such as Lukasiewicz, may introduce increased complexity. Discrete norms lead to less computational complexity than continuous norms.

2.3.2. Fuzzy ontology languages

Fuzzy OWL is an extension of the widely used OWL ontology language. The fuzzy extension of OWL 2 is Fuzzy OWL 2 [9], which provides support for representing fuzzy data types, fuzzy modifiers, fuzzy classes, axioms, and fuzzy Abox.

Fuzzy OWL is based on fuzzy DLs $f\text{-}\mathcal{SHIF}(\mathcal{D})$ and $f\text{-}\mathcal{SHOIN}(\mathcal{D})$. Fuzzy OWL also has several interchangeable syntactic forms: RDF/XML, abstract syntax, etc.

2.3.3. Fuzzy reasoning engines

Approximate reasoning techniques are often used in practical reasoning under fuzzy logics. For example, a limited number of discrete fuzzy values or crisp approximations to fuzzy sets are used. This can lead to tractable reasoning while still maintaining the useful properties of fuzziness. Some OWL reasoners like Pellet or HermiT (traditional OWL reasoners) have been extended to handle fuzzy ontologies. These reasoners can infer fuzzy memberships for concepts, classes, and properties, although they are not always as fully featured as dedicated fuzzy DL reasoners.

The fuzzyDL reasoner [10] provides reasoning services over fuzzy ontologies. As crisp ontologies are a special case of fuzzy ontologies, it also can be used for reasoning over crisp ontologies. It supports an extension of the fuzzy Description Logic $\mathcal{SHIF}$ (D) (including GCIs, role inclusions, transitive roles, functional roles, inverse roles, inverse functional roles, reflexive roles, and data property range axioms). FuzzyDL reasoner performs pre-processing, tableaux reasoning and applies optimisation techniques to maximize its computational efficiency. DELOREAN reasoner supports fussy OWL2 [11]. It translates fuzzy OWL2 ontologies to crisp OWL2 ontologies. Then translated versions can be used by classical Description Logic inference engines.

2.4. Description logic-based technologies, modelling both fuzzy and probabilistic knowledge

There are only a few research on combining both fuzzy and probabilistic knowledge in ontologies. Fuzzy Bayesian networks (FBN) enhance the classical BN by adding vague and imprecise knowledge that may be attached to the random variables. So, FBNs are hybrid models, combining the capabilities of Bayesian networks and fuzzy logic [12]. The proposed solution is described as a process, taking as an input a fuzzy ontology and outputs a probabilistic fuzzy ontology. On the author's knowledge, reasoning tool, using the resulting ontology was not proposed. Combination of Fuzzy and Probabilistic logics using fuzzy Bayesian networks were also discussed in [13, 14]. In other cases, researchers combine different inference mechanisms, such as fuzzy rule-based systems (for ambiguity) and Bayesian networks [15]. The idea is to ensure use of different approaches to represent different types of knowledge. Complexity and decidability problems of these approaches were not discussed.

3. METHODOLOGY FOR SELECTING THE APPROPRIATE TECHNOLOGY FOR MODELING OF UNCERTAIN KNOWLEDGE

The complexity of reasoning in expressive DLs is the main problem in the practical usage of ontology-based knowledge models. And inclusion of probabilistic or fuzzy knowledge usually leads to decreasing of reasoning effectiveness. We propose a methodology for selecting the suitable for specific domain and tack knowledge representation formalism (having minimal possible logical complexity) that can ensure maximal reasoning effectiveness.

Our methodology includes the following steps and main sub steps (see fig.1):

1. **Identify the Domain and its Complexity:**

Number of classes and relations; needs from general concept inclusions; needs from complex roles (inverse, transitive, etc.); needs from storing and reasoning with grand number of individuals.

2. **Define the needed Reasoning Tasks**

Importance and frequency of usage of subsumption tacks; importance and frequency of usage of query answering tacks; importance and frequency of usage of satisfiability tacks; importance and frequency of usage of A-box reasoning tacks.

3. **Consider Ontology Requirements and Select a basic description logic** based on the expressiveness and reasoning needs (e.g., $\mathcal{ALC}$, $\mathcal{SHIQ}$, $\mathcal{EL}$, etc.); based on the important functionality for the domain and application and complexity of reasoning tacks.

4. **Consider existing standards** (e.g., OWL 2, effective reasoning engine) if you're working with existing ontologies or web-based applications

5. **Choosing of appropriate extensions of crisp knowledge (fuzzy or probabilistic, or both uncertain type knowledges should combined**

Is it important to include probabilistic knowledge? What is the best probabilistic model; Is it important to include fuzzy knowledge? What is the best fuzzy model?

6. **Adding needed non-crisp knowledge** based on the importance of needed non-crisp knowledge and computational complexity. Identify the types of uncertainty related to data and applications.

6.1. Adding probabilistic knowledge in the knowledge bases:
Preliminary examination of the needs, analysing specifics of the needed probabilistic knowledge and selection of appropriate probabilistic elements, Choosing the probabilistic model (e.g., pDL, MLNs, P-OWL, etc.).

6.2. Adding fuzzy knowledge in the knowledge base:
Preliminary examination of the needs, analysing specifics of the needed fuzzy knowledge and selection of appropriate fuzzy elements, choosing the fuzzy model

7. **Find appropriate reasoning tools**
8. **Evaluation of possibilities of existing reasoners for selected extended DL variant**

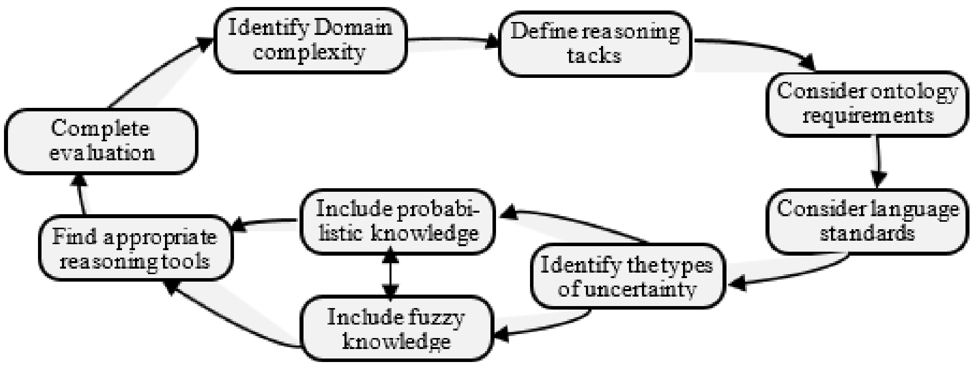

Figure 1. The methodology for selecting the appropriate technology

In many cases, results from evaluation can be unsatisfiable, or developers may wish to achieve higher expressiveness, or faster reasoning. To do so, all the main steps should be repeated in the context of stated new goals.

4. DISCUSSION

During the first three steps selection of the underlined crisp description logic system is performed. The selection is based on domain and tack requirements on the size of ontologies, richness of knowledge representation, important reasoning tacks and theoretical evaluation of expected reasoning complexity. If there is a need to work with large ontology, decreasing of logical complexity is very important. This can be achieved by careful exploration of needed restrictions related to reasoning with GCIs, roles and individuals and avoid usage of transitive, reflexive, inverse or functional roles, if possible. Try to use lightweight OWL sublanguages for effective reasoning if its limitations are acceptable for the application. For example, in Medical domain a highly expressive DL like **SROIQ** might be needed for complex hierarchical relationships and constraints, but ontology size should be restricted to ensure reasonable computational efficiency. On the other hand, lightweight **EL** is used in large ontologies to ensure reasonable efficiency of reasoning tacks like query answering.

Consideration of the possibilities of partitioning for large ontologies and the available inference algorithms (in the used reasoners or specific software) and optimizations in available reasoners also is important. Standard reasoners are optimized for reasoning tacks over standard OWL sublanguages and underlined logical systems, so consider selection of the logical system, related to some standard OWL sublanguage or implemented in well working or highly optimizer reasoner (step 4). If it is impossible, think about development of specific reasoning and optimization algorithms (for your specific ontological system).

Choosing the needed description logic extension (fuzzy or probabilistic) is a result of analysis of the needs from usage of non-crisp knowledge and analysis of probabilistic and fuzzy extensions capabilities, proposed in this work. The most complex case is when domain and tack needs require working with both probabilistic and fuzzy knowledge. This is usual case in some practical domains, as medical or e-learning domains. Identifying conditional independence relationships within the ontology will reduce the number of variables that need to be modelled and allows probabilistic reasoning to focus only on the essential dependencies. Choosing only a subset of features that are important for the reasoning task is also important.

To minimize complexity, use simple fuzzy operations, such as Zadeh's T-norms and T-conorms (for intersection and union, respectively) or Lukasiewicz logic (which is more computationally friendly). Avoidance of nested or higher-order fuzzy operations (e.g., fuzzy intersections of fuzzy intersections) and usage of discrete fuzzy models and reasoning tools, implementing approximate reasoning will ensure reasonable complexity. Selection between conditional probability models and Markov probabilistic models according to the modelled domain specifics and available well optimized reasoning tools is important for development of effective systems. Probabilistic models, based on Bayesian networks are better for combining both probabilistic and fuzzy knowledge in the same ontology. Another approach for combining the two types of non-crisp knowledge is to add probabilistic knowledge in ontology by usage of annotation properties**.** Most of non-crisp knowledge modelling studies mainly focus on how to represent probabilistic or fuzzy information in ontologies and perform reasoning through them. It is important to explore carefully reasoner's properties and optimizations. Selection or development of highly optimized probabilistic or fuzzy DL reasoning tools that use approximation techniques for inference also is important. Most of the upper-discussed approaches for development of fuzzy or probabilistic ontologies are new and are not automatically applicable for systems that already have and use developed ontologies. And in some cases it will be better for experts to have possibilities to store crisp, fuzzy or probabilistic knowledge in one and the same knowledge base and have possibilities to reason using independently crisp, probabilistic or fuzzy knowledge using simpler and faster reasoners in parallel processes and integrate results. ByNowLife framework [16] is a step towards ensuring integration of crisp ontologies and BNs. It presents an approach for integrating BN with OWL by providing an interface for retrieving probabilistic information through SPARQL queries. ByNowLife can transform logical information contained in an ontology into a BN and probabilistic information contained in a BN into an ontology. So, ByNowLife, can integrate separate ontologies and BN knowledge bases into a single knowledge base to ensure both logical and probabilistic reasoning through it, and also can separate crisp and probabilistic knowledge from probabilistic ontologies. This can be useful for building reasoning engines, performing parallel crisp and probabilistic reasoning to decrease effective reasoning complexity.

5. CONCLUSION

Non-crisp knowledge representation and reasoning are crucial for many practical domains and tacks, but reasoning complexity in corresponding systems is very high. To reduce the computational complexity in non-crisp extensions it is better to avoid use of expressive ontologies (e.g. using fewer hierarchical layers or fewer disjointed concepts, simpler roles and instance restrictions). Non-crisp extensions also should have minimal possible expressiveness (e.g. use most adequate models, minimal possible numbers of probabilistic variables, or discrete norms in fuzzy extensions, if possible). The main strength of the proposed methodology is its orientation to maximal possible decreasing of logical complexity of non-crisp ontologies. The proposed analysis of various DL extensions shows how the simplification can be done. Some of standardized OWL reasoners have optimized probabilistic or fuzzy extensions (and should be used in practical reasoning systems), but they are very slow for reasoning with large and complex ontologies. Large, complex knowledge bases should be partitioned to ensure faster reasoning, if it is possible. This makes the question of development of effective reasoners for particular logics very important.

ACKNOWLEDGEMENT

The authors would like to thank the Research and Development Sector at the Technical University of Sofia for the financial support.

REFERENCES

[1] Chekol, M.W., Stuckenschmidt, H. Tractable reasoning in probabilistic OWL profiles. In *Proceedings of the 33rd ACM Symposium on Applied Computing*, 2018, pp. 888-895.

[2] Ceylan, I.I., Penaloza, R. The Bayesian ontology language BEL. *Journal of Automated Reasoning*, vol.58, no.1, 2017, pp. 67-95.

[3] Botha, L., Meyer, T., Penaloza, R. The probabilistic description logic. *Theory and Practice of Logic Programming*, vol. 21, vol.4, 202, pp. 404-427.

[4] Carvalho, R.N., Laskey, K.B., Costa, P.C. PR-OWL–a language for defining probabilistic ontologies. *International Journal of Approximate Reasoning*, no. 91, 2017, pp. 56-79.

[5] Zhang, S., Sun, Y., Peng, Y., Wang, X. BayesOWL: A prototype system for uncertainty in semantic web. In *Proceedings of the International Conference on Artificial Intelligence*, 2009, July, pp.1-8.

[6] Bellodi, E., Lamma, E., Riguzzi, F., Zese, R., Cota, G. A web system for reasoning with probabilistic OWL. *Software: Practice and Experience*, vol. 47, no.1, 2017, pp.125-142.

[7] Riguzzi, F., Bellodi, E., Lamma, E., Zese, R. Probabilistic description logics under the distribution semantics. *Semantic Web*, vol. 6, no.5, 2015, pp. 477-501.

[8] Matsumoto, S. *PR-OWL decision: A framework for decision making with probabilistic ontologies.* George Mason University, 2019, (200 p.).

[9] Straccia, U. Fuzzy logic, annotation domains and semantic web languages. In *Proceedings of International Conference on Scalable Uncertainty Management*, SUM, 2011, pp.2–21.

[10] Huitzil, I., Straccia, U., Bobed, C., Mena, E., Bobillo, F. The serializable and incremental semantic reasoner fuzzyDL. In *2020 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 2020, July, pp. 1-8, IEEE.

[11] Bobillo, F., Delgado M., G´omez-Romero J. Reasoning in fuzzy OWL 2 with DeLorean. In *International Workshop on Uncertainty Reasoning for the Semantic Web*, 2013, pp. 119-138.

[12]Riali, I., Fareh, M., Bouarfa, H. A Semantic approach for handling probabilistic knowledge of fuzzy ontologies. In *Proceedings of the 21st International Conference on Enterprise Information Systems* (*ICEIS*), 2019, pp. 407-414.

[13] Riali, I., Fareh, M., & Bobillo, F. ProbFuzzOnto: A fuzzy ontology-driven uncertainty approach using fuzzy Bayesian networks. *International Journal of Fuzzy Systems*, vol.26, no.4 , 2024, pp. 1-21.

[14] Riali, I., Fareh, M., Bouarfa, H. Fuzzy probabilistic ontology approach: a hybrid model for handling uncertain knowledge in ontologies. *International Journal on Semantic Web and Information Systems*, vol. 15, no.4, 2019, pp. 1–20.

[15] Lukasiewicz, T., Straccia, U. Description logic programs under probabilistic uncertainty and fuzzy vagueness. *International journal of approximate reasoning*, vol. 50, no.6, 2009, pp. 837-853.

[16] Setiawan, F.A., Budiardjo, E. K., Wibowo, W.C. ByNowLife: a novel framework for OWL and Bayesian network integration. *Information*, vol.10, no.3, 2019, pp. 95.

Information about the authors:

Tatyana Ivanova received her M.S. in Mathematics and Informatics from Sofia University and Ph.D. degree in Artificial Intelligence Systems from Technical University of Sofia, in 1987 and 2009, respectively. She is currently an Associate Professor of Computer Systems and Technologies at Technical University, Sofia, Bulgaria. Her research interests are Semantic Web, E-learning, Ontological Engineering, Databases. Email: tiv72@abv.bg

Petya Petkova – Graduated in 2018 with M.S. Electronic Management in English Language Faculty of Engineering in Technical University of Sofia and a year later completed a PhD in Communication and Computer Technologies, specialty Automated systems for information processing and control in the same university. Now she is Assistant Professor in Technical College of Sofia, part of Technical University of Sofia. The research field areas she is interested in are big data, Internet of Things ecosystem, machine learning models for data analysis, blockchain and distributed ledger technologies, natural language processing. E-mail: petya.petkova@tu-sofia.

Manuscript received on 13 March 2025