ISSN 2278-3091
Volume 10, No.2, March - April 2021
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse661022021.pdf
https://doi.org/10.30534/ijatcse/2021/661022021

Knowledge Representation of Highly Dynamic Ontologies Using Defeasible Logic

Faiz Al-Shrouf
Isra University, Faculty of Information Technology, Department of Computer Science, Amman-Jordan

ABSTRACT

Description logic gives us the ability of reasoning with acceptable computational complexity with retaining the power of expressiveness. The power of description logic can be accompanied by the defeasible logic to manage non-monotonic reasoning. In some domains, we need flexible reasoning and knowledge representation to deal the dynamicity of such domains. In this paper, we present a DL representation for a small domain that describes the connections between different entities in a university publication system to show how could we deal with changeability in domain rules. An automated support can be provided on the basis of defeasible logical rules to represent the typicality in the knowledge base and to solve the conflicts that might happen.

Key words: Knowledge Representation, Ontologies, Defeasible Logic, Description Logic, Formal Methods.

1. INTRODUCTION

This paper presents defeasible description logics representation of ontology of a small domain of publications and research issues in a university.

The term ontology is used in different domains with different senses. Information systems borrowed the word ontology from philosophy reinterpreted it to be more suitable for their concerns [1]. We might have athing of the term as computational ontology which early defined as "explicit specifications of conceptualizations" [2]. The term Ontology used by Aristotle was defined as the science of "being qua being," i.e., the study of attributes that belong to things because of their very nature.

Computer science cares about computational ontology that is a special type of information object or computational artifact. Computational ontologies are used to model and represent the structure of the system or some domain of knowledge in a formal way [4]. The description of the domain deals with the things, entities and the relations existed between them.

The ontology engineer defines the entities in the system in addition to the relations among them. The base of the ontology consists of the generalization/specialization relationship of the concepts. An example is a university that has people and the inter-relationships between them, if we are interested in the educational part then person, lecturer and student might be relevant entities where the person is a neat concept of the Lecturer and Student. Cooperates-with can be considered as a relation between people. A specific student in the University is an illustration of the student concept. [2].

Student identifies ontology as "An ontology is a formal, explicit specification of a shared conceptualization". In this clarity, he merged the definition of Gruber who defined ontology as "explicit specification of a conceptualization" and the characterization of Borstdescription of ontology as a "formal specification of a shared conceptualization".[3]
The definition of Studer focuses on three terms

- The conceptualization.
- The formal and explicit specification.
- Sharing.

2. ONTOLGIES IN AI

The main purpose of ontologies in the field of AI is to facilitate data sharing and reuse them. Ontologies had a research interest in many AI fields such as knowledge engineering, natural-language processing and knowledge representation. [4] [12].

The main reason for the popularity of ontology is that it is used as a way to share common understanding of a domain and can be communicated between users and agents.

Ontologies are used with great success in education because they allow to formulate the representation of a learning domain by specifying all concepts involved, relations between concepts and all properties and conditions that exist. [20].

The formalism of ontologies has become more attractive recently by the existing of the means of languages such as OWL (Web Ontology Language) and OWL2, that are used for representing Semantic Web Ontologies. Another approach to represent ontologies that we will use in this paper is description logic. [4].

Description logics (DLs) are a family of knowledge representation languages that are widely used in ontological modeling. DL is a solid foundation for the W3C OWL.

Inferring new knowledge increases the power of the modeling language. The computation of inferences is called "Reasoning". The goal of DL is to supply reasoning algorithm with a good performance. [5].

The standard DL–based reasoning is monotonic which specifies that the knowledge base (KB) is incremental. Monotonic behaviour can be exhibited using the following example: if we have $\alpha \rightarrow \beta$ it follows that $\alpha \ and \ \gamma \ \rightarrow \ \beta$ for any γ. Assume α is "I am a bird" and γ is "I can fly" then we can say "if I am bird then I can fly". Now if γ is "I am an emu" we can according to the classical logic express the following statements "If I am bird and I am an emu then I can fly". The previous conclusion is not reasonable in all domains because an emu is actually an exception and it does not fly. This is the problem of monotonicity, so sometimes non-monotonic reasoning is required. This form of information usually represents typical situation. Birds typically fly but, in some instances, we have exceptions such in the case of penguin and emu. These exceptional cases symbolize defeasible information [6]. Defeasible Logics is the approach that we use to deal with KBs that contain defeasible information. Defeasible logic gives the flexibility to use non-monotonic reasoning. [7].

The main objective of the paper is to signify ontology of a small domain related to the publications and research issues in a university. In this document (*i*) We will correspond to the domain using description logic. (*ii*) We will use defeasible logic to deal with exceptions that we might face through our representation. (*iii*) We will show how we can extract information from this ontology.

The rest of this paper is organized as follows: Section 3 introduces the background of logic. Section 4, presents the description logic and defeasible logic, section 5 presents our methodology, the domain and the knowledge representation and lists the benefits of the formalism applied .and finally, Section 6 concludes the results and displays suggested future work.

3. BACKGROUND

First-order logic is the dominant logic formalism technique used for knowledge representation. In first order logic, the domain of the interest can be portrayed as a set of objects each of which has an individual identity, and allows the construct of logical formulas around these objects formed by predicates, functions, variables and logical connectives.

First-order logic is the wide spread,important representation formalism. Entailment is a notion that describes the reasoning that can be captured by first order logic. Also, to entailment of the first order logic, it introduces the notion of *universal truth – tautology* that is true regardless of preconditions. The process of deriving logical consequences from a theory is called "Deduction". The deduction allows us to extract knowledge that is not explicitly given but implicitly existed in the knowledge.

At the heart of this is what has become known as a *proof theory*. The proof theory describes syntactic rules that act on theories which allow it to be deriving logical consequences without explicit recurrence to models.
In the below sections, we will list two types of logics that are built on first order logics, description and defeasible logics.

4. DESCRIPTION LOGICS

Ontologies can be represented using different techniques. One of the most used demonstrated languages that are used in ontological representations is the family description logics.

Description logic has been used as a representation tool before the progress of ontologies and ontology modelling. The grand W3C OWL Web Ontology Language is based on the description logic.

DLs are logics, and they are equipped with formal semantics that are precise meaning of DL ontologies. The power of this semantics allows us to exchange ontologies built on DL without ambiguity as to their meaning and enable us to infer additional knowledge from the facts embedded in an ontology using the logical deduction.

Inferring new knowledge increases the power of the modeling language. The computation of inferences is called reasoning, and the goal of DL is to supply reasoning algorithm with good performance.

The main property of DL is that it gives a proper balance between expressiveness and computational complexity of reasoning. [8] [13].

Description logic has the following properties:
- Decidable.
- Expressive enough.
- Major knowledge representation paradigm.

4.1 Basic Building Blocks of DL Ontologies

In DL there are three types of entities:
- Concepts: represents sets of individuals, (unary predicates in FOL)
- Roles: represents binary relations between individuals. (Binary predicates)
- Individual names. (constants)

In DLs, Figure 1, we usually separate the axioms into three different groups:

- Assertional (**ABox**) axioms.
- Terminological (**TBox**) axioms.
- Relational (**RBox**) axioms

TBox: consists of statements in the form $C \sqsubseteq D$ or $C \equiv D$,….From the business domain example we can give the following example:

Man≡ *Human* ⊓*Male* (Man is a male human).

ABox: consists of statements in the form C(a) or R(a,b).

Examples for ABox statements are *Flight(FL4711)* and *booked By(FL4711, UbiqBiz),*

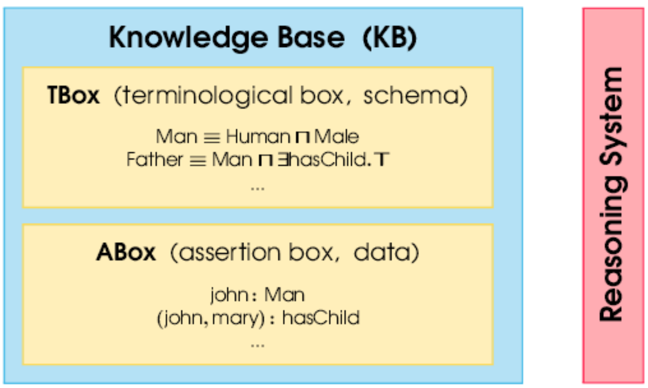

Figure 1: Description Logic (DL) Structure

DL Boolean constructors:

- Conjunction
- Disjunction
- Negation.
-

DL statements relate named or anonymous concepts are constructed by:

- Inclusion
- inverse inclusion
- equivalence.

DL also contains:

- $\top \equiv C \sqcup \neg C$
- $\bot \equiv C \sqcap \neg C$

As an example, we can write $C \sqsubseteq D \sqcup \neg E$ which is translated to

$$\forall x: (C(x) \rightarrow (D(x) \vee \neg E(x))).$$

The semantics of DL is expressed in the form of interpretation $I = (\Delta^I, \cdot^I)$. in which

$\Delta^I$is the domain, $\cdot^I$ is an interpretation function that maps every concept name N to subset A^I of Δ^I and every role r to a binary relation r^I over Δ^I.

In the rest of this paper, we will use type of DL called SROIQ that is widely used DL language these days [dl primer]. SROIQ basic concept C can be expressed as follows:

$C ::= N\_C \mid C \sqcap C \mid C \sqcup C \mid \neg C \mid \top \mid \bot \mid \exists R.C \mid \forall R.C \mid \geq n\ R.C \mid \leq n\ R.C \mid \exists R.$

Where n is a non-negative integer [5]

4.2 Defeasible Logic

The main aim of developing defeasible logic by [9] was the importance of computational efficiency. In classical FOL when systems have tens of thousands of propositions, the complexity of the algorithm becomes a critical issue. Defeasible logic is a good solution for such a problem and these days we have several examples of implementations exist [10], and some of them can deal with theories consisting of over 100,000 propositional rules [11]. Defeasible logic was successful in representing non-monotonic reasoning and has the required flexibility to deal with applications such as contracts, business rules, automated negotiation and multi-agent systems.[6]

In defeasible logic, we have the following types of knowledge: facts, strict rules, defeasible rules, defeaters, and a superiority relation. [7].

Facts represent unarguable statements such as “Mohammed is a student” that can be formed as Student (Mohammed). Strict rules are the classical rules in FOL in the form of -if then- which means whenever the premises are true then the conclusions are true too. An example is “Journal papers are publications” written in logic

$$Journal(x) \rightarrow Publication(x)$$

Defeasible (non-strict) rules are rules that hold in the typical situation. A good example is when we say “Birds fly” this is a precise statement, but the defeasible version will be something like “most birds fly”.

Defeaters are rules that are not used to conclude anything but to prevent a conclusion.

The superiority relation in a binary relation defined over the set of rules. The superiority relation determines the relative strength of two (conflicting) rules.

We will use the defeasible logic in this paper to solve situations where we have some exceptional cases in out domain.

5. METHODOLOGY

In this section, we will present the ontology and then we will use description logic and defeasible logic as a modeling technique for knowledge representation of the underlying ontology.

5.1 Domain

We present an illustration of a domain that describe the relations between members of an educational institute and the publications that they have produced. In addition,,we will describe some properties for each member type such as giving lectures. Members of our institute could be a faculty member or a student. Students might be Ph.D. students. Members of our institute can publish contents. The publications could be of type book, an article in a journal, or an article in a conference.

Each member in our domain has an interest in some research areas. Faculty members can supervise students except that faculty members with MSc degree cannot supervise Ph.D. students.

Figure 2 depicts the concepts of our domain and the relations between them.

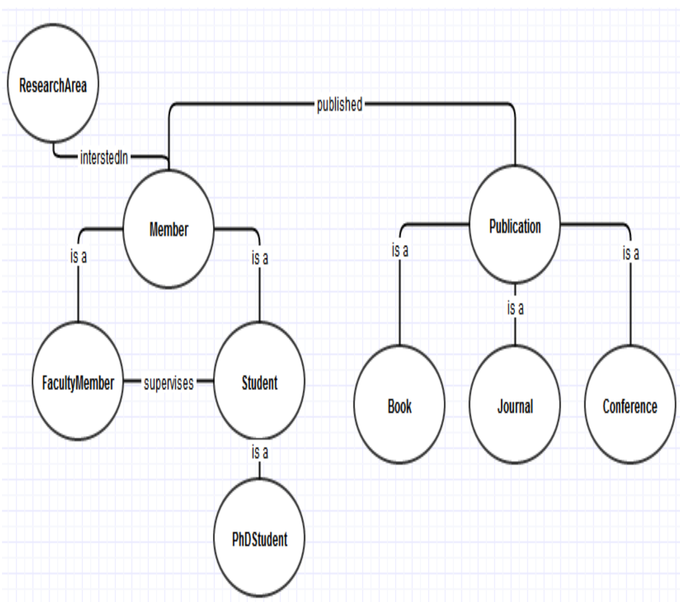

Figure 2: Members in Educational Institute

5.2 Description Logic

The description logic consists of concepts, roles and individual names. The concepts in our Domain are:
Research Area, Member, Faculty Member, Student,PhDStudent, Publication, Conference, Book, Journal and giving Lectures.

TBOX
The following rule says that Faculty Member is a member

$$FacultyMember \sqsubseteq Member \text{ (FOL: } FacultyMemeber \rightarrow Member)$$

The same is applied on Student. Also, we say that Conference, Journal and Book are Publication.

$$Student \sqsubseteq Member$$
$$PhDStudent \sqsubseteq Student$$
$$Conference \sqsubseteq Publication$$
$$Journal \sqsubseteq Publication$$
$$Book \sqsubseteq Publication$$

In addition to concepts, description logic contains roles. For example, some roles in the domain are interested in supervising, and publishment. Interested in describes the relation between members and their research areas Supervises is a relation between faculty members and students that describe the supervision relation. Published is a relation between the member and the publication.

Supervises(NADEEM,MOHAMMED) is an axiom says that Nadeem supervises Mohammed, Published(AHMAD,DL) tells us that DL is published by Ahmad, interested in (Mohammed, IT) states that Mohammed is interested in the IT research area.

The Author concept says that Author is a Member and has published some books.

$$Author \equiv Member \sqcap published(Publication.Book)$$

$$FOL(\forall x \exists y\ Member(x) \wedge Book(y) \wedge published(x,y) \rightarrow Author(x))$$

IT Lecturer is a faculty member with interest in IT research area

$$ITLecturer \equiv FacultyMember \sqcap interestedIn.(ResearchArea.\{IT\})$$
$$FOL(\forall x\ FacultyMember(x) \wedge interestedIn(y, IT) \rightarrow ITLecturer(x))$$

Paper is a publication in a conference or in a journal:

$$Paper \equiv Conference \sqcup Journal$$

Researcher is a professor who published, at least, one paper

$$Researcher \equiv FacultyMember \sqcap published.(Publication.Paper)$$

The associate professor is a professor with at least five published paper three of them are published in journals.

$$AssociateProf \equiv Professor \sqcap \exists{\geq}5\ Published.Paper \sqcap \exists{\geq}3\ Published.Journal$$

$$FullProf \equiv AssociateProf \sqcap \exists{\geq}10\ Published.Paper \sqcap \exists{\geq}7\ Published.Journal$$

In the following rules, we say that anyone gives lectures is a professor. Students are not professors, but Ph.D. students may give lectures.

$$GiveLectures \sqsubseteq FacultyMember$$
$$Student \sqsubseteq \neg FacultyMember$$
$$PhDStudent \sqsubseteq GiveLectures$$

Another example to describe the supervisee relationship is as follows: faculty members can supervise students, but faculty members with MSc degrees cannot supervise Ph.D. students.

$$\forall supervises.Student \sqsubseteq FacultyMember$$
$$\forall supervises.PhDStudnet \sqsubseteq FacultyMember \sqcap PhDHolder$$

The first rule says that any faculty member can supervise students while the other one says that only faculty members with a Ph.D. degree can supervise PhD students.

In the above rules, we may say some conflicts such as in the case of giving lectures. We presume that anyone gives lectures is a faculty member. On the other hand, students do not give lectures, but sometimes Ph.D. students give lectures. In such a situation, the knowledge is called defeasible because faculty members typically give lectures. To deal with such a situation, we will use the defeasible login in the next section.

5.3 Defeasible Logic

Defeasible logic is a way to work with non-monotonic reasoning and has the required flexibility to deal with applications such as regulations contracts, business rules, automated negotiation and multi-agent systems.

As we mentioned before defeasible logic knowledge consists of facts, strict rules, defeasible rules, defeaters, and a superiority relation.

In our domain facts could be something like Student (Ahmad) which states that “Ahmad is a student”. Strict rules such as “Journal papers are publications” written in logic

$$Journal(x) \rightarrow Publication(x)$$

Defeasible Rules are rules that hold the typical situation. An example of a defeasible rule from our domain will be “Most lectures are given by professors”.

$$GiveLectures(x) \Rightarrow Professor(x)$$

An example of defeaters could be:

$$PhDStudent(x) \rightsquigarrow GiveLectures(x)$$
$$Student \longrightarrow \neg Professor$$

Which says that “Ph.D. students may give lectures”.

When we have conflicting rules, we need to use the superiority relation. In our domain, the following rules will produce a contradiction

$$r1: GiveLectures(x) \Rightarrow Professor(x)$$
$$r2: Student \longrightarrow \neg Professor$$
$$PhDStudent(x) \rightsquigarrow GiveLectures(x)$$

If we introduced a superiority relation > then we can say r2>r1. In such a case we can conclude that Ph.D. students who give some lectures are not faculty members.

Another example is the supervise example

$$r3: \forall supervises.Student \sqsubseteq FacultyMember$$
$$r4: \forall supervises.PhDStudnet \sqsubseteq FacultyMember \sqcap PhDHolder$$

The rules say that faculty members can supervise students. In the case of Ph.D.students,faculty members with a Ph.D. degree are only allowed to supervise Ph.D Students, we will use superiority relation to giving more strength to rule 4 so we can say r4>r3.

$$r5: AssociateProf \equiv Professor \sqcap \exists \geq 5\ Published.Peper \sqcap \exists \geq 3\ Published.Journal$$
$$r6: FullProf \equiv AssociateProf \sqcap \exists \geq 10\ Published.Peper \sqcap \exists \geq 7\ Published.Journal$$

In the above rules, we have sort of overlapping and conflict between them because Associate Professor requirements are asubset of Full Professor requirement so we need use superiority relation. In this case, we have r5>r6.

The usage of description logic gives us the ability to express the knowledge in the domain and by the combination of defeasible logic, we able to express the typicality and handle the exceptions in our domain knowledge base.

5.4 Formal Methods Support the Domain

The usage of logic to model our ontology is beneficial regarding the reasoning in many ways. The advantage of this formal representation is as follows:

1. Explanation: when answers are given the area possible explanation. For example, we can use this formalism to select supervisors and explain why.

2. Decision making: it will help in taking decisions. For example, we will be able to decide who is available to supervise student X or whether we have enough full professors or not.

3. Detecting anomalies: inconsistencies, incompleteness and circularity can be detected. This can be done easily in defeasible logic using static analysis for instance. [6]

4. Debugging: We can perform debugging because in many cases we know what the answer to a particular query should be.

The reasoning in our ontology should be used to answer questions such as:

Can X give lectures?
Who is available to supervise student X?
How many publications have been published by X?
Who is the author?

6. CONCLUSIONS AND FUTURE DIRECTIONS

In this paper, we showed how can we deal with dynamic domains and we presented an ontology that describes the relations between researchers and publications. We used one of the widely accepted knowledge representation technique to represent the domain that is the defeasible logic to overcome the situations in which exceptions arise. Of course, there is much work to be done such as modelling the regulations of promotions where we can see the suitability of defeasible logic.

Our ontology will be the heart of our tool that to be modeled in our project for future considerations study, which used to classify and identify the experts in the field of universities. In many cases governments, research institutes, and universities themselves are seeking for individuals with specific sets of expertise. The underlying ontology is the core of the knowledge representation of the knowledge that the tool is using.

Our ontology consists of vocabulary to represent the academic members and ranks, the publications and the research interests of the employees in the institute.

This representation gives us the ability to identify the research areas, the deep of the knowledge and the value of the publications of the members of any institute.

Institutes searching for people to fill some required positions can use the tool to find the right individuals according the relations between the individuals and areas of interest that the tool can find by exploring the publications and the journals where these publications are published.

For future directions, we will contribute that we have a research project. In this research project, we need to build an expert system. One of the key persons for this project to succeed is to find a person with deep knowledge in AI expert systems development. The tool is going to start narrowing the search by investigating the people with research areas in AI, Expert Systems and knowledge representation for example, this can be decided by exploring the ontology and finding the connections between persons and research areas. Research areas can be specified by evaluating the value of the publications that are connected to set of journals in the ontology. The tool will provide a list contains the persons that have the highest matching value for the required position.

ACKNOWLEDGEMENT

The author owes thanks and appreciations to Scientific Research Deanship at Isra University for facilitating procedures of conducting this research and the financial support to the underlying project.

REFERENCES

1. Z. Gloria, **Ontology: its transformation from philosophy to information systems.** *Proceedings of the International Conference on Formal Ontology in Information Systems*-Volume 2001.ACM, 2001.

2. G. Nicola, D. Obrele, S. Staab**. What is an Ontology?** *Handbook on ontologies.* Springer Berlin Hiedelberg, 2009, pp. 1-17.

3. S. Rudi, V.R. Benjamins. **Knowledge engineering: principles and methods.** *Data & knowledge engineering*. Vol 25. No. 1 pp. 161-197, 1998.

4. M. Kody, T. Meyer, and I. J. Varzinczak**. A defeasible reasoning approach for description logic ontologie**s. *Proceedings of the South African Institute for Computer Scientists and Information Technologists Conference*. ACM, 2012.

5. K. Markus, F. Simancik, and I. Horrocks. **A description logic primer**. *arXiv preprint arXiv*: pp. 1201-4089, 2012.

6. A. Grigoris, D. Billington, M. J. Maher. .**On the analysis of regulations using defeasible rules**. *Systems Sciences, 1999.HICSS-32. Proceedings of the 32nd Annual Hawaii International Conference on* .IEEE, 1999.

7. G. Guido. **Defeasible description logics. Rules and rule markup languages for the semantic web**. Springer Berlin Heidelberg, pp. 98-112. 2004.

8. S. Grimmest, H. Pascal, A. Andreas. **Knowledge representation and ontologie**s. *Semantic Web Services: Concepts, Technologies, and Applications.* pp. 51-105. 2007.

9. N. Donald. **Defeasible logic**. *Handbook of logic in artificial intelligence and logic programming 3*, pp. 353-395. 1994.

10. E. Thomas**, Combining answer set programming with description logics for the semantic web**. *Artificial Intelligence* 172.12 :pp. 1495-1539. 2008.

11. M. Michael. **Efficient defeasible reasoning systems.** *International Journal on Artificial Intelligence Tools* 10.04, pp. 483-501. 2001.

12. K. Munir, M. Shiraz. **The use of Ontologies for Effective Knowledge Modeling and Information Retrieval.** *Applied Computing and Informatics Journal.* Volume 14,Issue 2, pp. 116-126. 2018.

13. I. Varzinczak. **A note on a Description Logic of Concept and Role Typicality for Defeasible Reasoning Over Ontologies.** *Springer Verlag.* Vol. 12, No. 2, pp. 297-325, 2018.

14. K. Stancin, P. Poscic, D. Jaksic. **Ontologies on Education-State of the art.** *Springer Verlag.* Vol 25, No 14, pp. 5301-5320, 2020.