DESCRIPTION LOGICS REASONING IN WEB-BASED EDUCATION ENVIRONMENTS

NENAD KRDŽAVAC

Department of Computer Engineering and Informatics, School of Electrical Engineering, University of Belgrade
Bulevar Kralja Aleksandra 73, 11000 Belgrade, Serbia and Montenegro
nenadkr@galeb.etf.bg.ac.yu

DRAGAN GAŠEVIĆ, VLADAN DEVEDŽIĆ

FON – School of Business Administration, University of Belgrade
Jove Ilića 154, POB 52, 11000 Belgrade, Serbia and Montenegro
gasevic@yahoo.com, devedzic@galeb.etf.bg.ac.yu

Abstract. The paper proposes some applications of description logic reasoning techniques in web–based education environments. Current solutions are web oriented but without applying ontology paradigm and do not use full power of semantic web and description logics reasoning techniques which are more useful than case base reasoning. We propose which description logic reasoning technique can be applied in some segments of web–based education systems. How to implement these techniques is not topic of the paper.

1. Introduction

Artificial intelligence plays important role in Intelligent Tutoring Systems (ITS). The architecture of World Wide Web is a good environment for ITS. Ontologies have a big influence on the development of the future ITS. Description Logics (DLs) are good candidates as ontology languages [5], but also the actual recommendation for the Web Ontology Language (OWL) is based on DL. This paper analyzes the role of description logics reasoning techniques in web-based educational environments, and how one can use reasoning techniques in Web-based Education Systems (WBESs) especially in cases when tutors want to reuse parts of the knowledge built in other tutors, which have already been developed [3]. Reasoning is important to ensure the quality of an ontology [6]. Some web education systems use case based reasoning [9]. Furthermore, rule base reasoning described in [4] where the developers use an XML format to represent course materials. The XML format will be only one part of learning material (for example students' answers), and second part should be represented, for example, in OWL DL (learning materials). This subdivision helps use full DLs' reasoning services in a WBES, since it is a natural way to represent course material as description logics knowledge base i.e. OWL DL as TBox and XML format as ABox.

In section 2, we briefly look at some basic concepts of description logics, describe basic reasoning techniques in some description logics, and give a formal definition of knowledge representation system based on description logics. In section 3 we present some DLs reasoning techniques in the web-based education environment and explain how we can use description logics reasoners in WBES for an intelligent analysis of student solutions. In section 4 we show how we can use a DL reasoner to improve communication between a WBES and a user of the system. Especially we stress how we can apply some advantages of the DL reasoner in collaborative WBES.

2. Description logics properties

Description Logics are the most recent name for a family of knowledge representation (KR) formalisms that represent the knowledge of an application domain (the “world”) firstly by defining the relevant concepts of the domain (its terminology), and then using these concepts to specify properties of objects and individuals occurring in the domain (the world description). Description logics are descended from so called “structured inheritance networks”, which were introduced to overcome ambiguities of early semantic networks and frames [1]. The basic entities for representing knowledge using DLs are so called concepts, which correspond to formulas with one free variable in mathematical logic. Complex concepts are built from concept names (unary predicates), role names (binary predicates) and concept constructors [2]. The smallest propositionally closed description logic is ALC (Attributive Language with Complements). Definition of syntax and semantic and some extensions are given in [1] [2]. For example, by using ALC concepts, we can describe fathers having at least one daughter using the concept

$$\text{Male} \sqcap \forall\, \text{hasChild}.\, \text{Human} \sqcap \exists\, \text{hasChild}.\, \text{Male} \tag{1}$$

Where Male, Human and Female are concept names and hasChild is a role name [2]. Expressive power of ALC is too weak for many applications and extensions can be divided into (at least) three groups: Restriction of interpretations, additional concept constructors, and role constructors. For example, extension of ALC description logic is SHIF (D) description logic.

A description logic knowledge base is naturally separated into two parts: TBox is a set of axioms describing a structure of a domain. (i. e. conceptual schema) and ABox is a set of axioms describing a concrete situation (data) [2] [1]. Basic description logics reasoning techniques are: concept consistency [2], concept subsumption, instance checking, and concept satisfiability [7]. According to [8], there is a correspondence between DLs knowledge base and OWL. The previous ALC formula (1) can be represented in the RDF syntax as follows:

```xml
<owl:Class>
    <owl:intersectionOf rdf:parseType=" collection">
        <owl:Class rdf:about="#Male"/>
        <owl:Restriction>
            <owl:onProperty rdf:resource="#hasChild"/>
            <owl:toClass>
                <owl:intersectionOf rdf:parseType=" collection">
                    <owl:Class rdf:about="#Human"/>
                    <owl:Restriction>
                        <owl:onProperty rdf:resource="#hasChild"/>
                        <owl:hasClass rdf:resource="#Male"/>
                    </owl:Restriction>
                </owl:intersectionOf>
            </owl:toClass>
        </owl:Restriction>
    </owl:intersectionOf>
</owl:Class>
```

OWL has many features in common with description logics, but also has some significant differences. The first difference between OWL and description logics is that the OWL syntax is an RDF-based syntax. OWL information is thus encoded in RDF/XML documents and parsed into RDF graphs composed of triples. Because RDF graphs are graphs, however, it is possible to create circular syntactic structures in OWL, which are not possible in DLs [8]. Some constructs in ontology languages go beyond the standard description logic constructs.

3. Applications of DLs reasoning techniques in the web-based education environment

Many web-based education environments and adaptive hypermedia systems have an experts' knowledge embedded in their structure. They use different reasoning techniques to help the authors to make improvements in the course design (e.g. case-based reasoning techniques explained in [9], or rule base reasoning [4]). For example, Simic and Devedzic [4] used an XML format to represent the domain knowledge and generate a CLIPS file (*.clp) before using the reasoning mechanism. The Jess Expert System shell is used as an inference engine [4]. However, there are some problems that are difficult for Jess to solve:

1. Can not avouch if course materials subsume another one.

Learning material is organized in chapters, and chapters are organized in lessons. DL reasoner has these classification mechanisms to check it. Figure 1 represents the hierarchy of chapters in a lecture of a logics course. Learning material in Logics can divide in to two chapters: Classical Logics and Non classical Logics. These two chapters are concepts in a DLs knowledge base. The Classical Logics chapter consists of two subchapters: first order logic and propositional logic, while the non classical logics chapter consists of two chapters: modal logics and temporal logics.

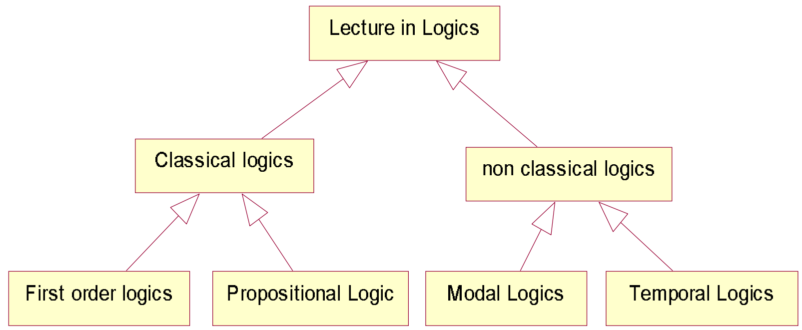

Figure 1. Hierarchy of teaching material in logics

If one wants to add a new chapter, for example, the chapter about description logics, a DL reasoner, using classification, can find "right place" for this new concept (in this case chapter) in the taxonomy tree. The Jess reasoner is unable to do it.

2. If a student's answering (in an XML format) is a model of a domain knowledge (i.e. an OWL ontology).

Students' answers can be submitted to a tutoring system in an XML format. A few students can give different answers to the same question. The reasoner can check if answers are model of the learning material.

3. Can not help an intelligent analysis of student solutions.

We can use a DLs based reasoner instead of Jess, but we have to divide the domain knowledge in two parts: TBox and ABox. The syntax of OWL is the syntax of RDF and the semantics of OWL are an extension of the RDF semantics [8]. TBox can be represented as an OWL DL and Abox can be represented in an XML syntax. Relationships between OWL DL and description logics are described in [13]. For preparing a course material we can use some ontology tools like Protégé (http://protege.stanford.edu/). Reasoning with the ontology can be reduced to knowledge base satisfiability SHOIN (D) and SHIF (D) description logics respectively [8]. The OWL's relationship to expressive description logics provides a source of algorithms for solving key inference problems, in particular satisfiability.

As mentioned in [3], two or more Web-based ITSs can refer to a common, shared part of their knowledge as in Figure 2. For instance, two tutors (App1 and App2) can have their own private knowledge and reasoning mechanisms. The Web pages corresponding to either tutor in this scenario must contain pointers to ontologies the tutor uses as its meta-knowledge [3].

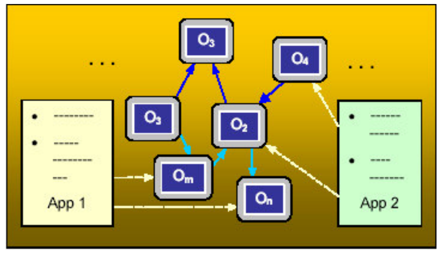

Figure 2. Two web-based ITS applications sharing ontologies

The ontologies on Figure 2 can be interpreted as Tbox in some description logics. Having understand those ontologies as only learning material and the application's knowledge, then we may use a description logic reasoner to check if some local knowledge is a model for any ontology O_i i.e. consistency Abox w. r. t Tbox. Figure 2 shows that O_i ontology is a part of the other ontology or ontologies. The reasoner can check it before using the ontologies in the knowledge teacher component. Furthermore, if we want to upgrade some of the ontologies we can use the reasoner to check consistency i.e. check modeling errors.

Currently, all adaptation technologies applied in Web-based Adaptive Education Systems (AES) are adopted from either the ITS area or the adaptive hypermedia area [11]:

1. Curriculum sequencing
2. Intelligent analysis of student' s solutions
3. Interactive problem solving support
4. Example-based problem solving support
5. Collaboration support
6. Adaptive presentation
7. Adaptive navigation support
8. Adaptive collaboration support

We explain only intelligent analysis of student solutions that deals with students' final answers to educational problems (which can range from a simple question to a complex programming problem) no matter how these answers were obtained. Intelligent analyzers can tell what is exactly wrong or incomplete and which missing or incorrect knowledge may be responsible for an error [11]. If we spouse that a WBES has its domain knowledge organized as an ontology we can use a DLs reasoner to check errors using the consistency reasoning technique. If a student's answer is submitted as an OWL document, we can use the technique to find all inconsistent classes mentioned in that OWL document and these classes are the result of student's modeling errors. Intelligent analyzes can provide the student with an extensive error feedback and update student model [11].

In spite of differences between DLs and OWL, DLs reasoning mechanisms (classification, subsumption, etc.), are one possible choice in WBES, where teaching materials is based on ontologies. For reasoning service we may also use other formalisms such as First Order Logic (FOL) or conceptual graphs, but in spite of the high worst case complexity of reasoning in such DLs, highly optimized implementations of these algorithms are available and have been shown to work well with realistic problems [8]. For example, according to [14], knowledge base satisfiability in SHIQ (D) is EXP-TIME complete. This is not case in reasoning technique such as resolution in FOL.

4. DL Reasoner as a core of explanation system

Every WBES needs at least two kinds of knowledge which can be clearly separated. One kind is a domain knowledge, which a user should learn. The other kind is the tutoring knowledge, which teaching material [10]. The major task of an ITS is comparison of a student answer with the domain knowledge in order to assess the student. This comparison is based on the student model maintained by the system [10]. The learner's knowledge can be a subset of the domain knowledge and the reasoner can use subsumption algorithm to check it. If we suppose that the knowledge domain and knowledge of the learner are based on description logics then they have two part of the knowledge, Tbox and Abox. We can use the reasoner to check:

1. Is Abox in the learner's knowledge subsumed by Abox part knowledge of the domain?
2. Is Abox in the learner's knowledge model of Tbox part knowledge of the domain? This reasoning technique called consistency Abox w. r. t. Tbox.
3. Is Tbox in a learner's knowledge subsumed by Tbox in knowledge of the domain?

For example if the domain knowledge is a concept of a mother who has only sons and at least one of them is a doctor or lawyer

$$\text{Mother} \doteq \text{Female} \sqcap \forall \text{has\_child}. \neg \text{Female} \sqcap \exists \text{ has\_child}. (\text{Doctor} \sqcup \text{Lawyer}) \tag{5}$$

If the student's knowledge is (i.e. student answer):

$$\text{Mother} \doteq \text{Female} \sqcap \forall \text{has\_child}. \neg \text{Female} \tag{6}$$

The reasoner can conclude that the student's answer is true but not complete because the concept in the student's knowledge is subsumed by a concept from the domain knowledge.

According to [10], the main task of explanation WBES is to communicate its domain knowledge to the user. A user is novice in the domain and the user's answer can be syntactically wrong but semantically true. The DL reasoner can conclude that the user does not input a wrong answer. The reasoner uses the subsumption technique and find a concept which subsumes the concept of the student's answer and help the student to understand the difference between his/her answer and the answer in the domain knowledge.

5. Practical experience

We have been implementing a description logics reasoner for education systems. We try to extend the existing multi-tutor system [4] with description logics reasoner. The reasoner should use all the reasoning techniques we have previously mentioned. We implement this reasoner as a Java API with support for OWL ontologies. In order to parse and query (RDQL) XML and RDF documents, as well as OWL ontologies we use the JENA framework

6. Conclusion and future work

In the paper we analyzed applications of description logics reasoning techniques in WBESs. We discussed the existing systems and identified their disadvantages. We believe that this paper may help researchers to use the full power of a description logics reasoner during implementation of a WBES. If we understand DLs reasoners as web-agents we may use them in the collaborative web-based education environment. The practical contribution of our approach is a reasoner that uses well-defined description logic reasoning techniques. Since this reasoner is being developed as an API it can be used by variety of WBESs. That way, we can use this reasoner instead of popular inference engines and APIs (e.g. Jess, Algernon, etc.). In the future we will finish the implementation of the reasoner and investigate its capacities in real-world WBESs.

7. References

1. F. Baader, D. Calvanese, D. McGuiness, D. Nardi, P. Patel - Schneider, Description Logic Handbook - Theory, Implementation and Applications, Cambridge university press, (2003)
2. 2. C. Lutz. The Complexity of Description Logic with Concrete Domains. PhD Thesis, LuFG Theoretical Computer Science, RWTH Aachen, Germany, (2002)
3. A. Mitrovic and V. Devedzic, A Model of Multitutor Ontology -Based Learning Environments, In Proceedings of ICCE Workshop on Concepts and Ontologies in Web-based Educational Systems, ISSN: 0926- 4515, New Zealand, (2002)

4. G. Simic, The Multi -cources Tutoring System Design, vol. 1, no.1, ComSIS, (2004)
5. D. Tsarkov and I. Horrocks. DL reasoner vs. first-order prover. In Proc. of the 2003 Description Logic Workshop (DL2003), volume 81 of CEUR (http://ceur-ws.org), pp.152-159, (2003)
6. F. Baader, I. Horrocks, and U. Sattler. Description logics as ontology languages for the semantic web. In Dieter Hutter and Werner Stephan, editors, Festschrift in honor of Jörg Siekmann, Lecture Notes in Artificial Intelligence. Springer, (2003)
7. F.M. Donini, M. Lenzerini, D. Nardi, A. Schaerf, Reasoning in description logics, In Gerhard Brewka, editor, Foundation of Knowledge Representation, pp.191-236. CSLI-Publications, (1996)
8. I. Horrocks and P. F. Patel-Schneider. Reducing OWL entailment to description logic satisfiability. In Dieter Fensel, Katia Sycara, and John Mylopoulos, editors, Proc. of the 2003 International Semantic Web Conference (ISWC 2003), number 2870 in Lecture Notes in Computer Science, pp.17-19. Springer, (2003)
9. M. Ferrario and B. Smyth, Collaborative Knowledge Management & Maintenance, In Proceedings of German Workshop of Case Based Reasoning, pp. 14-15, Germany, (2001)
10. G. Teege, Using Description Logics in Intelligent Tutoring Systems, In Proc. Int'l Workshop on Description Logics (DL94), (1994)
11. M. Ahdon, A. Zakaria F. Siraj, AI in Education: Towards Adaptive and Intelligent Web-Based Learning, (2002) (http://www.aisig.uum.edu.my/azizi/TN6073/Adaptive Education.pdf).
12. V. Haarslev and R. Möller, Racer: A Core Inference Engine for the Semantic Web, Proceedings of the 2nd International Workshop on Evaluation of Ontology-based Tools (EON2003), located at the 2nd International Semantic Web Conference ISWC, (2003), Sanibel Island, Florida, USA, pp.27-36, October, (2003)
13. Franz Baader, Ian Horrocks, and Ulrike Sattler, Description Logics as Ontology Languages the Semantic Web, In Dieter Hutter and Werner Stephan, editors, Festschrift in honor of Jörg Siekmann, Lecture Notes in Artificial Intelligence, Springer, (2003)
14. Stephan Tobies, Complexity results and Practical Algorithms for Logics in Knowledge Representation, PhD Thesis, LuFG, Theoretical Computer Science, RWTH-Aachen, Germany, (2001).