Expert Systems with Applications 38 (2011) 13975–13986

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Reasoning and change management in modular fuzzy ontologies

Yuncheng Jiang [a,b,*], Yong Tang [a], Qimai Chen [a], Ju Wang [c]

[a] School of Computer Science, South China Normal University, Guangzhou 510631, PR China
[b] State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, Beijing 100190, PR China
[c] School of Computer Science and Information Technology, Guangxi Normal University, Guilin 541004, PR China

ARTICLE INFO

Keywords:
Ontologies
Fuzzy ontologies
Modular ontologies
Description logics
Fuzzy description logics
Distributed description logics

ABSTRACT

The growing emphasis on complexity concerns for ontologies has attracted significant interest from both the researcher's and the practitioner's communities in modularization techniques as a way to decrease the complexity of managing huge ontologies. On the other hand, it has been widely pointed out that classical ontologies are not appropriate to deal with imprecise and vague knowledge, which is inherent to several real world domains. In order to handle these types of knowledge, some fuzzy extensions of classical ontologies are presented, yielding fuzzy ontologies. In this paper, we integrate modular ontologies with fuzzy ontologies, i.e., the notion of modular fuzzy ontologies is presented. Furthermore, we present an infrastructure for the representation of and reasoning with modular fuzzy ontologies based on distributed fuzzy description logics.

1. Introduction

Currently, research in the area of the Semantic Web (Berners-Lee, Hendler, & Lassila, 2001; Huang & Lin, 2010) is in a state where ontologies (Guzmán-Arenas & Cuevas, 2010; Gruber, 1993; Kang, Lee, Choi, & Kim, 2010) are ready to be applied in real applications such as Semantic Web portals, information retrieval or information integration (Stuckenschmidt & Klein, 2007). In order to lower the effort of building ontology-based applications, there is a clear need for a representational and computational infrastructure in terms of general purpose tools for building, storing and accessing ontologies. A number of such tools have been developed, i.e., ontology editors (Abu-Hanna, Cornet, Keizer, Crubézy, & Tu, 2005; Bechhofer, Horrocks, Goble, & Stevens, 2001), reasoning systems (Sirin, Parsia, Grau, Kalyanpur, & Katz, 2007; Tsarkov & Horrocks, 2006) and more recently storage and query system (Alkhateeb, Baget, & Euzenat, 2009). Most of these tools, however, treat ontologies as monolithic entities and provide little support for specifying, storing and accessing ontologies in a modular manner (Stuckenschmidt & Klein, 2007).

It is well-known that there are many reasons for thinking about ontology modularization (Stuckenschmidt & Klein, 2007). For example, in distributed environments such as Semantic Web, ontologies in different places are built independent of each other and can be assumed to be highly heterogeneous (Lee, Park, Park, Chung, & Min, 2010; Stuckenschmidt, Parent, & Spaccapietra, 2009). Unrestricted referencing of concepts in a remote ontology can therefore lead to serious semantic problems as the domain of interpretation may differ even if concepts appear to be the same on a conceptual level. The introduction of modules with local semantics can help to overcome this problem (Bouquet, Giunchiglia, Harmelen, Serafini, & Stuckenschmidt, 2004; Stuckenschmidt & Klein, 2007). Modularization also helps to manage very large ontologies which sometimes contain more than a hundred thousand concepts. These ontologies are hard to maintain as changes are not contained locally but can affect large parts of the model (Stuckenschmidt & Klein, 2007). Another argument for modularization in the presence of large ontologies is reuse (Grau, Horrocks, Kazakov, & Sattler, 2008): in most cases, we are not interested in the complete ontology when building a new system, but only in a specific part. Experiences from software engineering show that modules provide a good level of abstraction to support maintenance and reuse (Stuckenschmidt & Klein, 2007). On the other hand, a specific problem with distributed ontologies as well as with very large models is the efficiency of reasoning. While the pure size of the ontologies causes problems in the latter case, hidden dependencies and cyclic references can cause serious problems in a distributed setting. The introduction of modules with local semantics and clear interfaces will help to analyze distributed systems and provides a basis for the development of methods for localizing inference (Serafini, Borgida, & Tamilin, 2005; Stuckenschmidt & Klein, 2007).

Stuckenschmidt and Klein (2007) provided a representational framework for modular ontologies that builds on top of existing work on distributed description logics (Borgida & Serafini, 2003) as a framework for reasoning about distributed ontologies in order

* Corresponding author at: School of Computer Science, South China Normal University, Guangzhou 510631, PR China.

E-mail addresses: ycjiang@scnu.edu.cn, jiangyc@ics.ict.ac.cn (Y. Jiang).

0957-4174/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2011.04.205

to improve ontology maintenance and reasoning. More concretely, Stuckenschmidt and Klein (2007) concentrated on the benefits of modularization in the context of ontologies, explicit representations of the terminology used in a domain, defined a formal representation for modular ontologies based on the notion of distributed description logics, and introduced an architecture that supports local reasoning by compiling implied axioms. Furthermore, they addressed the problem of guaranteeing the correctness and completeness of compiled knowledge in the presence of changes in different modules and proposed a heuristic for analyzing changes and their impact on compiled knowledge and guiding the process of updating compiled information that can often reduce the effort of maintaining a modular ontology by avoiding unnecessary recompilation.

Obviously, the framework presented by Stuckenschmidt and Klein (2007) can only deal with classical (or crisp) ontologies. Nevertheless, it has been widely pointed out that classical ontologies are not appropriate to deal with imprecise and vague knowledge, which is inherent to several real world domains (Bobillo, Delgado, & Gomez-Romero, 2009; Sanchez, 2006). Since fuzzy set theory and fuzzy logic are suitable formalisms to handle these types of knowledge, some fuzzy extensions of classical ontologies are presented, yielding fuzzy ontologies (Bobillo et al., 2009, Bobillo, Delgado, Gomez-Romero, & Straccia, 2009; Jiang, Tang, Wang, Deng, & Tang, 2010; Lukasiewicz & Straccia, 2008). Fuzzy ontologies have proved to be useful in several applications, such as Chinese news summarization (Lee, Jian, & Huang, 2005) and semantic help-desk support (Quan, Hui, & Fong, 2006). There are also a lot of applications in the Semantic Web field (Quan et al., 2006; Straccia, 2006) and, more generally, in the Internet (Sanchez, 2006).

Consequently, fuzzy extension to the framework presented by Stuckenschmidt and Klein (2007) should be considered, as it would allow to turn the framework more intelligent, that is, be able to deal with more knowledge. This paper will investigate reasoning and change management in modular fuzzy ontologies. In other words, we will extend the framework presented by Stuckenschmidt and Klein (2007) to the fuzzy case (i.e., modular fuzzy ontologies). Concretely, we will define a representational framework for modular fuzzy ontologies that builds on top of Distributed Fuzzy Description Logics (DFDLs) as a framework for reasoning about distributed fuzzy ontologies and introduce an architecture that supports local reasoning by compiling implied fuzzy axioms. We further address the problem of guaranteeing the correctness and completeness of compiled fuzzy knowledge in the presence of changes in different fuzzy modules. Lastly, we will propose a heuristic for analyzing changes and their impact on compiled fuzzy knowledge and guiding the process of updating compiled fuzzy information that can often reduce the effort of maintaining a modular fuzzy ontology by avoiding unnecessary recompilation.

The rest of the paper is organized as follows: Section 2 introduces a representational framework for modular fuzzy ontologies that builds on top of DFDLs as a framework for reasoning about distributed fuzzy ontologies. In Section 3 we define reasoning mechanisms for modular fuzzy ontologies as a special case of general inference in DFDLs. Section 4 discusses the problem of handling changes in external fuzzy ontologies. At last, in Section 5 we review some related work on fuzzy ontologies and modular ontologies while Section 6 concludes the paper and presents some perspectives for future research.

2. Modular fuzzy ontologies

In this section, we present a formal model for modular fuzzy ontologies that will be used throughout the paper. Our starting point is the use of Fuzzy Description Logics (FDLs) as the basis for representing fuzzy ontologies. Especially, we formally introduce the syntax and semantics of the FDL $\mathcal{FSHIQ}$ which is the basis for our work. We then proceed with the definition of our model for modular fuzzy ontologies by defining a distributed extension of $\mathcal{FSHIQ}$ with fuzzy mappings between different models known as Distributed Fuzzy Description Logics (DFDLs). As our model turns out to be a subset of DFDLs, we conclude this section by explaining the restrictions to the general framework of DFDLs that apply to the modular fuzzy ontologies. Obviously, our model of modular fuzzy ontologies is a fuzzy extension of modular ontologies presented by Stuckenschmidt and Klein (2007).

2.1. Fuzzy ontologies and fuzzy description logics

Ontology is a conceptualization of a domain into a human understandable, machine-readable format consisting of entities, attributes, relationships, and axioms (Guarino & Giaretta, 1995). Informally, an ontology consists of a hierarchical description of important concepts in a particular domain, along with the description of the properties (of the instance) of each concept. The current standard language for ontology creation is the Web Ontology Language (OWL (Horrocks, Patel-Schneider, & Harmelen, 2003; Kang, Lee, Kim, & Lee, 2009)), which comprises three sublanguages of increasing expressive power: OWL Lite, OWL DL and OWL Full. However, since its first development, several limitations on expressiveness of OWL have been identified, and consequently several extensions to the language have been proposed (Bobillo et al., 2009). Now, the up-to-date Web Ontology Language is OWL 2 (Grau, Horrocks, Motik et al., 2008).

Description Logics (DLs for short) (Baader, Calvanese, McGuinness, Nardi, & Patel-Schneider, 2007) are a family of knowledge representation languages which can be used to represent the terminological knowledge of an application domain in a structured and formally well-understood way. DLs have proved to be very useful as ontology languages (Baader, Horrocks, & Sattler, 2005). For instance, OWL Lite, OWL DL and OWL 2 are close equivalents to the DLs $\mathcal{SHIF}(\mathbf{D})$, $\mathcal{SHOIN}(\mathbf{D})$ and $\mathcal{SROIQ}(\mathbf{D})$, respectively (Bobillo et al., 2009, 2009; Grau, Horrocks, Motik et al., 2008; Horrocks, Kutz, & Sattler, 2006; Horrocks et al., 2003).

Nevertheless, the conceptual formalism supported by classical ontologies and DLs may not be sufficient to represent imprecise and vague information commonly found in many application domains due to the lack of clear-cut boundaries between concepts of the domains (Quan et al., 2006). Since fuzzy set theory and fuzzy logic are suitable formalisms to handle these types of knowledge, some fuzzy extensions of classical ontologies are presented, yielding fuzzy ontologies (Bobillo et al., 2009, 2009; Lukasiewicz & Straccia, 2008).

In this paper, we consider fuzzy ontologies represented in the fuzzy description logic $\mathcal{FSHIQ}$, which is a sublanguage of the fuzzy description logics $\mathcal{FSROIQ}$ (Bobillo et al., 2009) and $\mathcal{FSROIQ}(D)$ (Bobillo et al., 2009). This choice is motivated by the facts: (i) $\mathcal{FSHIQ}$ covers a large part of the expressive power of the fuzzy Web Ontology Language (fuzzy OWL, f-OWL for short (Bobillo & Straccia, 2009b; Calegari & Ciucci, 2007; Stoilos, Simou, Stamou, & Kollias, 2006; Stoilos, Stamou, Tzouvaras, Pan, & Horrocks, 2005; Straccia, 2006)); (ii) we can base our framework on the framework presented by Stuckenschmidt and Klein (2007) that provide us with basic mechanisms for specifying links between fuzzy concepts in different fuzzy ontologies in a loose way. In the following, we briefly introduce the basic notions of the fuzzy description logic $\mathcal{FSHIQ}$ which is a fuzzy extension of the $\mathcal{SHIQ}$ (Glimm, Lutz, Horrocks, & Sattler, 2008; Stuckenschmidt & Klein, 2007) DL.

Fuzzy Description Logics (FDLs) (Bobillo et al., 2009; Bobillo & Straccia, 2009a; Hajek, 2005; Jiang et al., 2010; Lukasiewicz &