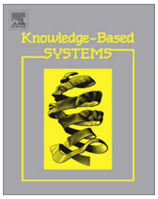
Knowledge-Based Systems 50 (2013) 234–245

Contents lists available at SciVerse ScienceDirect

Knowledge-Based Systems

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



Measuring ontology information by rules based transformation

Yinglong Ma [a,b,*], Ke Lu [c], Ying Zhang [a], Beihong Jin [b]

[a] *School of Control and Computer Engineering, North China Electric Power University, Beijing 102206, China*
[b] *State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, Beijing 100190, China*
[c] *University of Chinese Academy of Sciences, Beijing 100049, China*

ARTICLE INFO

Article history:
Received 28 October 2012
Received in revised form 14 June 2013
Accepted 19 June 2013
Available online 28 June 2013

Keywords:
Ontology
Ontology measurement
Stable measurement
Semantic measurement
Ontology Engineering

ABSTRACT

Ontologies have currently attracted much attention of researchers and engineers in many fields such as knowledge management, etc. It is attractive for ontology engineers to select and reuse the existing ontologies by measuring and evaluating them because ontology construction is rather tedious and costly. In this paper, a general framework for stable semantic ontology measurement is proposed. We first clarify the concepts of syntactic, semantic and stable semantic ontology measurement. Then we present the semantic derived model (SDM) to represent the semantic model of an ontology. By rule based transformation, an ontology can be automatically transformed into its final semantic derived model (FSDM) which is unique. Furthermore, we can measure ontologies based on FSDM by analyzing the types of entities of the existing ontology metrics. The related experiments are made to illustrate that our framework can effectively excavate and stably measure the semantics of ontologies.

1. Introduction

In the last decades, ontologies have attracted much attention of researchers and engineers in many fields such as knowledge management and acquisition [1,2], business decision supports [3], semantic search [4], etc. Ontologies formalize knowledge meaning of information and contents, and facilitate communication between human and software agents [6]. As a new successor of knowledge engineering, ontology engineering [5] aims at knowledge sharing and reuse by designing, implementing and deploying ontologies. However, ontology construction is rather tedious and costly [7,8]. It is attractive for ontology engineers to select and reuse the candidate ontologies that most satisfy their requirements by measuring and evaluating them [9].

In the literature [10–14,30,16] of measuring ontologies for ontology reuse, most of the existing approaches for ontology measurement neglect implicit kinship of entities and structural semantics hidden in ontologies. So we cannot correctly calculate entities in ontologies. For example, we only calculate those classes defined by `owl:Class`, and fail to consider some complex concepts which are very helpful to more precise semantic similarity computing between entities across interrelated ontologies. Furthermore, we also neglect the problem caused by flexibility in ontology representation when ontology languages with more expressivity are used to describe ontologies. In other words, the same semantic knowledge can be represented by different syntaxes in more expressive ontologies. As a result, what is surprised is that variable measurement results will be obtained based on the same semantic knowledge. This makes unreliable ontology measurement. Unless ontology measurement results are effective and reliable, the evaluation based on such measurement results should be doubtful.

In this paper, we confine our work to ontology measurement rather than ontology evaluation because we argue that effective and reliable ontology measurement is the precondition on which the meaningful and useful ontology evaluation can be made. We will discuss what problems need to be addressed in ontology measurement before one can achieve meaningful and useful ontology evaluation. We propose a framework to measure ontology information by rule based transformation. In this framework, we establish a unique semantic derived model for each of measured ontologies, and measurement based on ontologies will be reduced to that based on their semantic derived models. This will make us measure ontological entities without caring their specific syntactic representations.

The rest of this paper is organized as follows. Section 2 gives a motivating example to illustrate the problem we will address in this paper. In Section 3, we formally define four kinds of ontology measurement, and give the overview of this measurement framework. Section 4 presents semantic derived models (SDM). A rule based approach is proposed to transform an ontology to its FSDM. We also discuss the related properties of SDMs. Section 5 discusses how to collect measurement entities based on FSDM. Section 6 is the related ontology measurement experiments. Sections 7 and 8 are the related work and conclusion, respectively.

---

* Corresponding author at: School of Control and Computer Engineering, North China Electric Power University, Beijing 102206, China. Tel.: +86 10 61772643.

E-mail address: yinglongma@gmail.com (Y. Ma).

0950-7051/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.knosys.2013.06.016

2. Motivating example

In the last decades, considerable approaches have been adopted to study the principles and criteria of ontology measurement such as the literature [17–21]. Despite the fact that these approaches provide the useful guides about ontology measurement, several challenges keep unchanged, which make the ontology measurement process ambiguous and unreliable. In this section, we have identified the challenges that we will address by a motivating example.

Although ontologies can be represented by some languages such as RDF [22] and OWL[23], different ontology languages for representing ontological knowledge have different levels of expressiveness provided with different language constructs. For example, OWL has three increasingly expressive sublanguages: OWL Lite, OWL DL and OWL Full (ordered by increasing expressiveness). An expressive ontology language is more flexible to represent ontological knowledge. In other words, the same ontological knowledge can be represented in different syntaxes of ontology languages. Figs. 1 and 2 respectively illustrate the two examples of ontologies. The two ontologies represent the same ontological knowledge, but they are presented in the different syntaxes of the OWL language. The flexibility in ontology representation also brings about the problem of syntactic variability in ontology representation even if the same ontological knowledge is represented.

Furthermore, considering that most of the existing approaches of measuring ontologies are made based on ontology graphs which are the syntactic descriptions of ontologies, we establish the ontology graphs for the two ontologies in Figs. 1 and 2. They are shown in Figs. 3 and 4, respectively. An ontology graph is a graph-based representation for ontology information, where nodes are the elements including classes or instances, and edges are relations between these elements. Ellipses represent class nodes, and arrows are `subClassOf` relations.

What is surprised is that the two ontology graphs obviously are different although the semantic knowledge in them is the same. In the case, if we measure the different forms of ontology representa-

```
<owl:Class rdf:ID=″Researcher″>
   <owl:subClassOf rdf:resource=″#People″/>
   <owl:unionOf rdf:parseType=″Collection″>
      <owl:Class rdf:about=″#Professor″/>
      <owl:Class rdf:about=″#PhD″/>
   </owl:unionOf>
</owl:Class>
<owl:Class rdf:ID=″Prof_with_PhD″>
   <owl:intersectionOf rdf:parseType=″Collection″>
      <owl:Class rdf:about=″#Professor″/>
      <owl:Class rdf:about=″#PhD″/>
   </owl:intersectionOf>
</owl:Class>
<owl:Class rdf:ID=″Student″>
   <owl:intersectionOf rdf:parseType=″Collection″>
       <owl:Restriction>
           <owl:onProperty rdf:resource=″#register″/>
           <owl:someValuesFrom rdf:resource="#Dept"/>
       </owl:Restriction>
       <owl:Restriction>
           <owl:onProperty rdf:resource=″#take″/>
           <owl:someValuesFrom rdf:resource="#Course"/>
       </owl:Restriction>
    </owl:intersectionOf>
</owl:Class>
<owl:Class rdf:ID=″PhDStudent″>
    <owl:subClassOf rdf:resource=″#Student″/>
   <owl:subClassOf>
       <owl:Restriction>
           <owl:onProperty rdf:resource=″advisedBy″/>
           <owl:allValuesFrom rdf:resource=″#Professor″>
       </owl:Restriction>
    </owl:subClassOf>
</owl:Class>
<owl:Class rdf:ID=″People″/>
<owl:Class rdf:ID=″Professor″/>
<owl:Class rdf:ID=″PhD″/>
<owl:Class rdf:ID=″Dept″/>
<owl:Class rdf:ID=″Course″/>
```

Fig. 1. An example of ontology representation.