Yinglong MA[1], Yongming SHANG[1], Ke LU[2]

North China Electric Power University (1), University of Chinese Academy of Sciences (2)

Design of Stable Semantic Measurement for Ontology-Based Systems

Abstract. *Ontology selection is an important challenge in the field of ontology engineering. A feasible solution of ontology selection is to measure and evaluate the candidate ontologies, and further select and reuse the high-quality ontologies from them according to some measurement criteria. In this paper, we designed an ontology measurement system (ONTOM) for measuring the quality of candidate ontologies by integrating our semantic ontology metrics into ONTOM. The experiments show that ONTOM can effective measure the semantic quality of ontologies for ontology reuse.*

Streszczenie. *W artykule przedstawiono system pomiaru ontologii ONTOM, realizujący dwie główne funkcje: pomiar semantyk ontologii oraz określenie parametru jakościowego ontologii. W pierwszym punkcie dokonany jest dobór czynników, które należy uwzględnić w pomiarze. W drugim kroku działań chodzi o określenie parametru odzwierciedlającego ontologię danego kandydata. Przedstawiono wyniki badań eksperymentalnych. (**Opracowanie metody pomiaru semantycznego w systemie ontologicznym**).*

Keywords: ontology engineering, ontology measurement, ontology semantic metrics
Słowa kluczowe: inżynieria ontologii, pomiar ontologii, metryka semantyki ontologii.

1 Introduction

Ontology engineering is the next generation knowledge engineering [1]. Ontologies provide shared semantic vocabulary that agrees on domains of interests. They are crucial to develop the semantic-driven application systems, and have widely been applied in many fields such as knowledge management and information integration [2]. Different domain experts have manually created plenty of domain ontologies for their ontology based systems. Some domain ontologies are created in a (semi-)automatic manner. Currently, there are many domain ontologies are available on the Web. For example, you can obtain more than 100,000 ontologies by some search engines such as Swoogle [3]. However, the construction of ontologies is time-consuming and costly, so ontology engineers often achieve their ontology based applications by reusing the existing ontologies. To reuse ontologies is to measure the quality of ontologies, and further select the most suitable ontologies from the existing ontologies because you can only control what you can measure [4]. Unfortunately, there are at least two important challenges in the field of ontology reuse. First, we need to consider what should be measure. The existing approaches are just to measure syntaxes of ontologies such as the literature [5], but they can not measure the semantics of ontologies. The problem is that the same ontology semantic knowledge possibly can be represented in different syntaxes, so it seems that some of the current approaches can not satisfy the requirements of measuring ontology semantics. The second problem is how to measure the quality of a candidate ontology. We need a feasible measurement tool to help ontology engineers obtain the measuring results automatically. However, to our knowledge, there are no tools available to focus on stability of ontology measurement and automatically measure the semantic quality of ontologies indeed.

In this paper, we design and implement an ontology measurement tool called ONTOM. We mainly consider the two problems mentioned above. First, we fully consider measuring semantics of ontologies. To measure ontology semantics, we will perform a pre-processing algorithm that treats any candidate ontology to be measured as a unique semantic representation before measuring it. In the case, on one hand, we can measure the semantic of the ontologies. On the other hand, we also can ensure the stability of ontology measurement, i.e., the same semantic knowledge will have the same measurement results. Second, we design and implement some ontology metrics proposed in the previous work [6]. An ontology metric is an indicator that can reflect a certain characteristics of ontologies. Generally, ontology metrics are represented as formulae, which show how the entities in the ontologies to be measured are calculated. In the literature [6], we proposed a set of ontology metrics to measure ontology cohesion, which were selected as the criteria of ontology measurement for ontology based systems. They are: number of ontology partitions (NOP), number of minimally inconsistent subsets (NMIS), average impact of intramodule relationships (AIM-IR) and average depth of maximum concept subsumption of leaf concept (ADMCS-LC). Meanwhile, we will integrate and implement these ontology metrics into our ONTOM system for measuring the cohesion of candidate ontologies to be measured. We argue that more ontology metrics can be integrated in to the ONTOM system because we fully consider the scalability of ONTOM system.

This paper is organized as follows. Section 2 is the preliminaries about ontology and ontology measurement. In Section 3, we discuss the algorithm for ontology pre-processing. Section 4 introduces the criteria of ontology measurement, i.e., the ontology cohesion metrics. In Section 5, we discuss the system design overview of ONTOM, and further give the algorithms of implementing these ontology metrics in details. Section 6 is to show the effectiveness of our ONTOM system and the related experiments for semantic based ontology measurement are also made. Sections 7 and 8 are the related work and the conclusion, respectively.

2 Ontology Language and Ontology Measurement

Ontologies are a kind of knowledge representation tools. Ontology knowledge can be represented in some ontology description languages such as Resource Description Framework (RDF), Web Ontology Language (OWL) and Description logic (DL) [6]. OWL is an extension of RDF and a machine-readable language for sharing and reasoning information on the Web. Currently, OWL language has been recommended by W3C as the standard web ontology language. Generally, an OWL ontology representation consists of axioms and facts. Axioms are the semantic knowledge defined by building relationships between classes and properties. Facts represent the individual assertions. Description Logic language provides the formal ontology representation with a well-founded theoretical foundation.

To certain extents, ontologies are closely related to modern object oriented software design in representation.

The concept is the foundation of ontological knowledge, and the class is the core of object-oriented design. Hence, it is natural to use object-oriented software measurement methodology for the task of ontology measurement. In object-oriented software, cohesion refers to the degree of the relatedness or consistency in functionality of the members in a class. Cohesion measures separation of responsibilities, independence of components and control of complexity [8]. Similarly, ontology cohesion refers to the degree of the relatedness of OWL classes conceptually related by the properties. An ontology has a high cohesion value if its entities are strongly interrelated. Ontologies with strong cohesion are often desirable because a strong cohesion of an ontology means that the ontology can achieve common goals.

3 Algorithms of Pre-processing to Ontology

An algorithm of pre-processing for an ontology to be measured is adopted to treat the ontology as its unique semantic presentation. The reason why we do like this is to ensure that we can measure the semantic quality of ontologies rather than syntaxal quality. This will also ensure the measurement results should be stable and comparable [9]. Before introducing the algorithm, we should first introduce the concept of intramodule-relationship (IR).

Definition 1. An ontology can be regarded as a set of intramodules relationships. An intramodule-relationship (IR) refers to a binary relation between ontology elements, which is one of the following forms as follows.

1) Subsumption relation between concepts, e.g, owl:subClassOf(X, Y), which denotes the subsumption between Concept X and Concept Y.

2) Domain-range relation, e.g., R(X, Y)), which represents the domain of a binary relation R is Concept X, and the range of R is Concept Y.

3) Subsumption relation between properties, e.g, owl:subPropertyOf(R, S), which denotes the property subsumption between Property R and Property S.

4) Binary relation between individuals, e.g., R(a, b), which represents that Individual a is associated with Individual b by the binary relation R. And

5) Membership relation, e.g., X(a), which denotes that Individual a is an object of Class X.

The specific pre-processing algorithm can be divided into four steps as follows.

3.1 Pre-processing for Classes and Individuals

In ontology terms, a concept is just a class. For an atomic concept or individual, there is no need to make any pre-processing. The key problem occurs in anonymous classes and individuals. Anonymous classes are defined by some class constructors such as owl:Restriction, owl:unionOf and owl:intersectionOf. They have no names, which possibly cause the errors of the calculatrion of number of classes. Similar is to anonymous individuals. So these elements should be named. The existing reasoners such as KAON2 [10], can detect this labels and further rename them with new labels. Based on KAON2, we propose an algorithm to automatically detect the relevant labels and name anonymous classes, and tag it by using a unique concept name. Note that some anonymous classes may be nestedly defined by other anonymous classes. For example, Figure 1 is an anonymous class. The anonymous class includes the two nested anonymous classes:, i.e., Class 1 and Class 2. The nested anonymous classes also need to be further named. Here, Class 1 is named as ”All_hasDegree_Ph.D”, and Class 2 is named as ”All_hasTitle_Professor”. Anonymous individuals can be detected and named by class membership.

```
<owl:Class>
  <owl:intersectionOf rdf:parseType="Collection">
    <owl:Restriction>                                          } All_hasDegree_Ph.D
      <owl:onProperty rdf:resource="#hasDegree" />
      <owl:allValuesFrom rdf:resource="#Ph.D"  />
    </owl:Restriction>
    <owl:Restriction>                                          } All_hasTitle_Professor
      <owl:onProperty rdf:resource="#hasTitle" />
      <owl:allValuesFrom rdf:resource="#Professor" />
    </owl:Restriction>
  </owl:intersectionOf>
</owl:Class>
```

Fig.1. An example of nestedly defined concept

3.2 Mining of Implicit IRs

An ontology includes not only explicit syntax-based knowledge, but also the implicit knowledge that is derived from the explicit knowledge. In the case, it is obvious that those implicit IRs should be excavated and are used to enrich the semantics of the original ontologies. For example, we assume that the concepts represented by the above ontology segment is denoted ”Faculty”. Then, we can find that concept ”Faculty” is not only a subclass of concept ”All_hasDegree_Ph.D”, but also a subclass of ”All_hasTitle_Professor”. All these subsumptions are implicit. They should be excavated and explicitly expressed in the ontology. Beside some instances explicitly asserted by class membership, some classes or properties should be enriched with other instances by reasoning the ontology. Then, individual a is an instance (object) of concept (class) A, and concept A is a subclass of concept B, then, a should be also an instance of B.

3.3 Pre-processing for Cycles of IRs

Cycles of concept subsumptions will make semantic reasoners difficult to perform correct semantic reasoning. In this case, semantic reasoning process will not be terminatable. A cycle of concept subsumptions is of the form owl:subClassOf(A, A_1), owl:subClassOf(A_1, A_2), . . . , owl:subClassOf(A_n, A), where A, A_1, A_2,, ..., A_n,are concepts. It is obvious that cycles of concept subsumptions will not only make the program goes to closed loop, but also impede the ontology measurement process. So the cycles of concept subsumptions in an ontology to be measured must be eliminated. The specific treatment process is to detect all cycles of Irs based on the traditional search algorithm in Graph theory. Then, for each cyclic of IRs, we replace each A_i ($1 \le i \le n$) with A. Note that the relation owl:equivalentClass is a special case of owl:subClassOf, i.e., owl:equivalentClass(A, B) $\leftrightarrow$ owl:subClassOf(A, B) and owl:subClassOf(B, A). So the approach that we treat cycles of class equivalence is similar to the approach of treating cycles of class subsumption.

3.4 Pre-processing for Transitivity Derived IRs

For our pre-processing algorithm, there is dilemma to construct semantic representation of ontology. On one hand, we should excavate implicit semantic information as mor eas possible by semantic reasoning. On the other hand, the addition of implicit semantic information will cause variable semantic representation models. It also cause double counting of ontological IRs. For obtaining a unique semantic representation of an ontology, we compromise the two aspects. To avoid the problem of double counting of ontological IRs, we need to eliminate transitivity relationships of an ontology to be measured. For instance, an IR owl:subClassOf(A, C) is indirectly obtained by the related IRs owl:subClassOf(A, B) and owl:subClassOf(B, C). In the case, if the IRs owl:subClassOf(A, B) and owl:subClassOf(B, C) are counted, owl:subClassOf(A, C) will no longer be counted because counting owl:subClassOf(A, C) means the double counting to other

two IRs. This is necessary to make ontology measurement stable.

The following example illustrates us how to make pre-processing for an ontology. The ontology represented in Figure 2, and the pre-processing process is shown in Figure 3. The syntax based representataion of the ontology is shown in Figure 3.a. Figure 3.b is the ontology representation by semantic reasoning, and the Figure 3.c is the ontology representation after treating the transitivity derived relations.

```
<owl:Class rdf:ID="A">
  <owl:subClassOf rdf:resource="#B">
  <owl:equivalentClass>
    <owl:unionOf rdf:parseType="Collection">
      <owl:Class rdf:about="#C">
      <owl:Class rdf:about="#E">
    </owl:unionOf>
  </owl:equivalentClass>
</owl:Class>
<owl:Class rdf:ID="D">
  <owl:equivalentClass>
    <owl:intersectionOf rdf:parseType="Collection">
      <owl:Class rdf:about="#C">
      <owl:Class rdf:about="#E">
    </owl:intersectionOf>
  </owl:equivalentClass>
</owl:Class>
<owl:Class rdf:ID="B"/>
<owl:Class rdf:ID="C"/>
<owl:Class rdf:ID="E"/>
```

Fig.2. A representation example of ontology

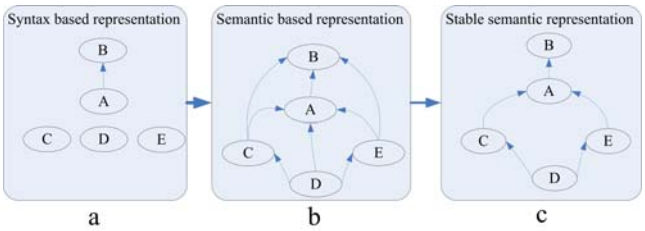

Fig.3. The procedure of pre-processing for an ontology

4. Criteria for Ontology Measurement

In the previous work[6], we proposed four semantic ontology metrics that are used to measure and evaluate cohesion of ontologies. These ontology metrics are first considered in our measurement as criteria of ontology measurement. Note that more ontology metrics can be be integrated our measurement system because we fully consider the scalability of our ontology measurement system.

4.1 Number of Ontology Partitions (NOP)

A partition of an ontology is defined as a subontology of this ontology. In an ontology partition, all concepts, properties and individuals are not semantically associated with other ontology partitions. The conception NOP represents the number of partitions in an ontology. The NOP metric can be used to measure whether the contents of an ontology have a common topic.

(1) $$NOP = |\{P_i\}|$$

where $\{P_i\}$ refers to the set of all ontology partitions.

4.2 Number of Minimally Inconsistent Subsets (NMIS)

In an ontology, an inconsistency is caused by some IRs in conflicts. For example, there exists an inconsistency in the three IRs, owl:subClassOf(A, B), owl:subClassOf(A, C), owl:complementOf(B, C) and A(a), where A, B, C are concepts, and a is an individual. By semantic reasoning, we can obtain the facts B(a) and C(a), but C and B are disjointed, which are in conflict. Some existing reasoners such as KAON2 can provide the function to check this inconsistency. MIS refers to Minimally Inconsistent Subset here. The set of all MISs in an ontology is denoted SMIS, and the cardinality of the set is denoted NMIS.

(2) $$NMIS = |SMIS|$$

An ontology with MIS is not in good design, and difficult to reuse because inconsistent IRs will impede the understanding and sharing of the whole ontology. MISs in the ontology should be detected by measuring MISs in the ontology, and therefore we can adopt some methods to eliminate them. Within an ontology with a large number of MISs, some ontological modules cannot be effectively congregated together and achieve a close and unambiguous goal.

4.3 Average Inconsistency Impact of Intramodule Relationship (AIM-IR)

In an inconsistent ontology, the inconsistency impact of an IR refers to the number of MISs containing IR. A metric AIM-IR is the ratio of the sum of inconsistency impact of all IRs to the number of all minimal inconsistent subsets in the ontology to be measured. Specifically, the metric AIM-IR of an ontology *O* can be formulated as follows:

(3) $$AIM\text{-}IR(O) = \frac{\sum_{ir \in (T \cup A)} impv(ir)}{|SMIS_o|}$$

A higher AIM-IR value of an ontology means that the ontology is more difficult to understand. AIM-IR can also be used to measure the inconsistent degree of an ontology.

4.4 Average Depth of Maximum Concept Subsumption of all Leaf Concepts (ADMCS-LC)

For an ontology *O*, a root concept is the one which is subsumed by no concept except itself. A leaf concept subsumes no concept except itself. A path of concept C, denoted by pC, is a sequence starting from a root concept to C by concept subsumption. The length of path pC, denoted by |pC|, is the total number of all concepts in pC. The depth of concept subsumption of C, denoted by depth(C), can be defined as $depth(C) = \max_{p \in S_{pC}} \{|pC|\}$, where S_{pc} is the set of all possible paths of C. The set of all leaf concepts in ontology O is denoted by SLC_O. The ADMCS-LC of *O* can be formulated as follows:

(4) $$ADMCS\text{-}LC(O) = \frac{\sum_{C \in SLC_O} depth(C)}{|SLC_o|}$$

A higher ADMCS-LC value of an ontology means that the ontology has a higher depth of concept subsumption, and the knowledge of the ontology is more cohesive. It can be used to measure the degree to which the semantic knowledge of an ontology to be measured is organized.

5. Design of ONTOM

5.1 System Overview

System ONTOM includes four main components, which is shown in Figure 4. Component *Ontology* is the ontology data. It can be implemented in ontology database or ontology files. Component *Ontology Pre-processing* is used to make pre-processing for the ontology to measured. It mainly includes the preprocessing for classes, individuals, mining of IRs, cycles of IRs, and transitivity derived IRs. The reason why we do like this is to treat an ontology to be measured for stable ontology metric, and ensure that we can measure the semantic quality of ontologies rather than syntaxal quality. Thus we can simplify the measurment process.

Once the ontology has been treated by the pre-processing, the ontology measurement is carried out as follows. The ontology processed is divided into partitions which do not have IRs with each other, then we can calculate the number of the partitions is the value of NOP.

For each partition, all the MISs are found and moved to the set of MISs. The cardinality of this set is NMIS. And then the value of AIM-IR is computed to measure the average inconsistency impact of IR. Once the value of ADMCS-LC has been computed, the measurement work of this ontology has been achieved. All these ontology metrics are designed and implemented according yo their definitions. Their values can be automaically obtained. At last, the related measurement results are shown to Component *User Interface*, through which ontology engineers can further perform the related ontology quality evaluation.

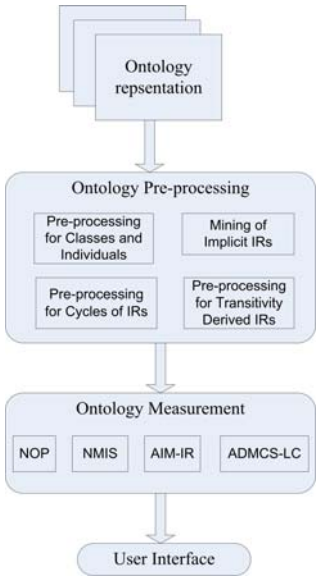

Fig.4. The process of ontology measurement in ONTOM

5.2 Ontology Measurement Based on Ontology Metrics

When the pre-processing for the ontology to be measured is finished, we can measure it based on the four ontology metrics.

5.2.1 Measurement of NOP

Acording to the definition of NOP, concepts or individuals connected directly or indirectly by IR will be put into the same partition. Moreover, the structure of a partition is a tree, so the problem computing NOP value is same to count the number of trees in a forest. As mentioned above, an ontology can be regarded as a set of IRs, denoted SIR. The algorithm for NOP isas follows.

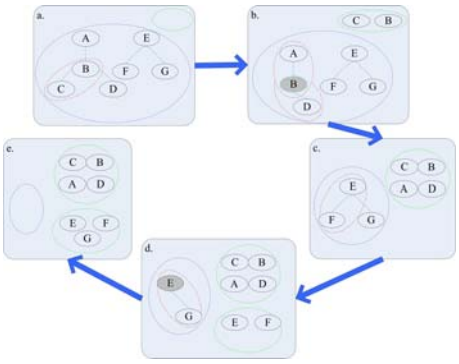

Fig.5. The procedure of finding the partitions

First, initial SIR as the set of all the IRs, select an IR, put its concepts or individuals into a partition *P*, and remove it from the set of IRs. Then find out all the IRs which include the concepts or individuals in P, and remove these IRs from SIR and put them into *P*. Repeat these two steps until there is no IR which includes any concept or individual in *P*. Therefore, a partition will be found. We repeat the procedure of finding a partition to find out other partitions. Finally, we will get the number of partitions. Figure 3 illustrates the procedure of finding the partitions.

For enhancing ontology cohesion, either the ontology should be decomposed into multiple ontologies with a single topic, or new semantic relationships should be exploited and added between these partitions. As illustrated in green ellipses in Figure 5, our system tool can display all the partitions, so as to assist the ontology engineer to refine an ontology.

5.2.2 Measurement of NMIS

For each partition, we can further find all the MISs. At the beginning, SMIS is empty and SIR includes all the IRs in this partition. All nonempty subsets of SIR will be ranked and traversed according to their cardinalities from the small to larger. For each subset of SIR being traversed, we will check whether it have IR conflicts. Using KAON2, we can check if a set of axioms is inconsistent. So the set of IRs is saved into a temperary ontology, and get its inconsistency by check the inconsistency of the temperary ontology S. If *S* is inconsistent, it will be moved into *SIMS*, and all the subset of SIR which contains *S* will be skiped. As a result, we will obtain all the MISs, which are stored in SMIS.

The more the MISs in an ontology, the more the ontology is difficult to share. Meanwhile, a large number of MISs in an ontology means that ontology engineers have to take more time and efforts to modify these inconsistencies. System ONTOM can evaluate the degree of inconsistency, and pinpiont all the MISs, which will help ontology engineers to modify the ontology.

5.2.3 Measurement of AIM-IR

After the NMIS is found, we notice different IRs may appear different times in the SMIS. IRs that appear more times in the SMIS can be considered as having more contribution to ontology inconsistencies. We use the term inconsistency impact values to denote the occurrence number that the same IR occurs in different MISs of SMIS. It is obvious that an IR with a high value in a MIS has a higher inconsistency impact values than those with lower values. Ontology developers and users have to take more time and efforts to understand and eliminate them.

Given the cardinality of the SIR and the sum of the cardinalities of all the set MISs, we can get the values of AIM-IR easily, according to the formulations described above. The value of AIM-IR given by System ONTOM is used to describe the extent to which an ontology is inconsistent. Meanwhile, the IRs occuring most times in the SMIS will be pinpointed, as this axiom may be the key piont to reduce the ontology inconsistency.

5.2.4 Measuring ADMCS-LC

At last, the metric ADMCS-LC will be used to measure the degree to which the semantic knowledge of an ontology to be measured is organized. First, the leaf concepts and the root cocepts will be found. Concepts that do not subsume any concept except for themselves, will be taged as leaf concepts. Concepts that are not subsumed by any concept are taged as root concepts. The depth of every concept is set initial value 1. Second, the depth of each leaf concept is counted, and the details of obtaining the depth of a leaf concept are illustrated in Figure 6. Based on these, the ADMCS-LC value of an ontology can be computed easily according to the formulation described above.

```
Input: SRS (the set of RS)
Output: DLC (the depths of leaf-concepts)
1   Boolean flag = true;
2   While flag is true do
3       flag = false;
4       for all IR ∈ SIR do
5           if IR is equivelent axiom then
6               if IR.leftside.depth < IR.rightside.depth + 1 then
7                   IR.leftside.depth = IR.rightside.depth + 1;
8                   flag = true;
9               end
10          else then
11              if IR.leftside.depth < IR.rightside.depth then
12                  IR.leftside.depth = IR.rightside.depth;
13                  flag = true;
14              else if IR.leftside.depth > IR.rightside.depth then
15                  IR.rightside.depth = IR.leftside.depth;
16                  flag =true;
17              end
18          end
19      end
20  end
21  mvoe the depths of the leaf-concepts to DLC;
22  return DLC
```

Fig.6. The algorithm to obtain the depth of the leaf concept

Note that, the metric ADMCS-LC only concerns on the concepts of an ontology, so the IRs in Figure 6 refer to the IRs which only have concepts in their two sides. Moreover, as the equivelent axioms also only have concepts in their two sides, while its concepts in its two sides represent the same thing, they can hardly be taken as having different depths, so they were made to have the same depth.

The ADMCS-LC value given by this framwork can well indicate the distribution of information across the different levels of ontology's concept subsumption or the fan-outs of parent concepts. It can also indicate how well knowledge is grouped into different categories and subcategories in an ontology.

6 Experiments and Results

By using System ONTOM, we will perform two preliminary experiments, including the measurement result analysis of our system, and the comparison of the stability differences between the syntax-based measurement and semantic-based measuring. Syntax based measurement is sometimes called structure based measurement.

6.1 Experiments and Results

In the experiment, several testing ontologies are used to obtain the values of the ontology cohesion metrics: NOP, NMIS and AIM-IR, they include: Object ontology[1], University ontology[2], Koala ontology[3], mini-Tambis ontology[4]. All these ontologies will be treated with pre-processing steps described in Section 3. Then, we can count the number of concept names (including atomic and complex concepts), properties, individuals, and IRs. The experimental results by using NOP, NMIS and AIM-IR to measure the four testing ontologies are shown in Figure 7.

	Object	Koala	University	Mini-Tambis
NOP	5	1	3	19
NMIS	0	3	15	30
AIM-IR	0	3.35	4.21	6.91

Fig.7. Measuring results of NOP, NMIS and AIM-IR

First, it is clear that the ontology with a higher NOP value has more topics than other, and the concept organization and aggregation within the ontology are loose relatively. Ontology engineers can either decompose it into multiple ontologies with a single topic, or exploit new semantic relationships, add them among these partitions and form a relatively complete topic.

Second, ontology with a higher NMIS value has a bigger scope of contents in conflict, and more modules of the ontology possibly cannot be effectively congregated together and achieve a close and unambiguous topic. Hence, it is rather difficult to understand and reuse.

Third, a higher AIM-IR value of an ontology means that the intramodule relationships in it have a higher inconsistency degree to which these intramodule relationships belong to together. Though a higher AIM-IR value may show the close interconnects between the intramodule relationships of the ontology, it also reflects that the ontology developers do not fully understand the domain knowledge very well; such an ontology will be difficult to understand and revise.

Finally, it seems ontology with a higher NOP value may have a lower NMIS value or a lower AIM-IR value, and the relationship between NMIS value and the AIM-IR value is nonlinear. According to the definition of ADMCS-LC, the ADMCS-LC value will not have strong relationship with the other three metric values. Hence, all of these metrics is necessary for the ontology measuring tool.

6.2 Comparison with Structure-Based Measurement

In this paper, we propose a semantic based ontology measurement. It is necessary to compare our approach with traditional structure based ontology measurement. Their main differences are: the former is stable and the latter's measurement results rely on the specific representation of ontology. To compare the stability differences between structure based ontology measurement and semantic based ontology measurement, we need to select two ontology metrics that focus on the same measurement property. The two ontology metrics should be designed structurally, semantically and respectively. We just select the ADMCS-LC metric from our four ontology cohesion metrics as the representative metric. We argue that the other three ontology cohesion metrics can be also selected for stability comparison.

Meanwhile, we designed a structural ontology metric ADMIT-LN (average depth of maximum inheritance tree of all leaf nodes) that completely corresponds to the ADMCS-LC metric. The definition and computation process are same to ADMCS-LC, except following points: 1) ADMIT-LN only deals with those classes that are explicitly defined. 2) ADMIT-LN only deals with concept subsumption (i.e., class inheritance) that is explicitly defined. 3) Before using ADMIT-LN to measure an ontology, we do not treat the ontology by the pro-processing approach described in section 3.

We obtained eight testing ontologies by using the Swoogle search engine. The testing ontologies are: Wine, Person, mini-Tambis, swrc v0.3, Terrorism, publication, univ-bench and GlycO_0_1, and then measured all these eight ontologies by ADMCS-LC and ADMIT-LN respectively. The experimental results of the two ontology metrics are shown in Figure 8. Each value in the figure is of the form X/Y. X is a measure value that corresponds to an ADMIT-LN value or an ADMCS-LC value for the same ontology of the eight testing ontologies. Y is the number of explicitly defined classes if X corresponds to ADMIT-LN. Otherwise, Y is the number of both all concepts obtained after pre-processing if X corresponds to ADMCS-LC.

	Wine	Person	miniTambis	Swrc	Terrorism	Publication	Univ-bench	GlycO
ADMIT-LN	1.27 / 76	2.71 / 21	1.61 / 102	2.56 / 105	2.5 / 21	2.82 / 13	2.53 / 43	8.35 / 370
ADMCS-LC	3.44 / 272	3.71 / 26	3.47 / 270	3.34 / 105	3.23 / 28	3.82 / 24	3.79 / 51	8.86 / 496

Fig.8. Comparison of ADMIT-LN and ADMCS-LC

[1] www.flacp.fujitsulabs.com/tce/ontologies/2004/03/object.owl.
[2] protege.stanford.edu/plugins/owl/owl-library/koala.owl.
[3] www.mindswap.org/2005/debugging/ontologies/miniTambis.owl.
[4] www.mindswap.org/2005/debugging/ontologies/tambis.owl.

After analyzing and comparing the experimental results, we can obtain that, 1) the ontology metrics semantically defined, e.g., ADMCS-LC, can find more classes as much as possible in the testing ontologies. That's because some implicit classes or nested anonymous classes can be found. Those ontology metrics structurally defined, e.g., ADMIT-LN, only find those classes that are explicitly defined by some labels such as owl:Class. 2) considering that classes are one of the key components of an ontology, we believe that an ontology cannot be accurately measured and evaluated if the number of classes of the ontology is estimated too low because we cannot evaluate the overall quality of ontology just by using the partial components of ontology.

7 Related Work

In the last yeas, many researches have been made in the field of ontology measurement. OntoQA is a tool that implements a number of metrics [5], the authors define metrics like richness, population, or cohesion, but they fail to define if the metrics are structurally or semantically defined. A quality-oriented ontology description framework (QOOD) [11] was also proposed. The authors describe how measures should be built in order to actually assess quality by these models, and the relation between the ontology description, the ontology graph, and the conceptualization that is expressed within the graph, and they also define measures for the structural, functional, and usability dimension. A framework for metrics in the wider area of ontology engineering was provided by OntoMetric [12]. OntoMetric proposed a complex framework consisting of 160 characteristics spread across five dimensions: content, language, methodology, tools and costs.

Most of existing ontology measurement frameworks [5,12,13] mainly consider ontology structure and seldom measure ontological semantics, and they may fail in measuring the ontologies in the context of the changing and dynamic Web, in which ontologies may evolving and become inconsistent [14]. Our previous work [6] proposed some semantic based ontology metrics, through which ontology engineers can fully considers the semantics of ontologies to be measured, and the stability of ontology measurement. Because semantic based ontology measurement needs semantic reasoning, a pre-processing algorithm is also needed to guarantee that all information in the ontology will be explicitly and non-redundantly expressed. Our reasoning service mainly is based on the KAON2, which implemented some novel algorithms [10,15].

To our knowledge, there is no tool for stable ontology semantic measurement that fully considers semantics of ontologies and stability of ontology measurement.

8 Conclusion

This paper designed and implemented an ontology measurement tool called ONTOM for stable semantic ontology measurement. The ontology preprocessing and calculation algorithms for ontology metrics were used in this tool. The system design implementation technology were introduced. The experiment results show that that this system can enable stable semantic measurement for ontologies. However, the system still needs to be improved, and the future work include: 1) integrating more ontology metrics such as ontology metrics related to ontology coupling and complexity because the ontology metrics adopted in this paper just belong to ontology cohesion metrics. 2) integrating some popular ontology evaluation approaches into our system. The aim of ontology measurement is to perform ontology evaluation, which will decide whether an candidate ontology can be reused.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (No.61001197), the State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences (No.SYSKF1010), Beijing Municipal Natural Science Foundation (No.4102058), the National Basic Research Program of China (973 Program) (No.2009CB320704), and the Fundamental Research Funds for the Central Universities.

REFERENCES

[1] Mizoguchi, R., Tutorial on ontological engineering: part 3: Advanced course of ontological engineering, *New Generation Computing*, 22 (2004), no.2, 198-220.

[2] Zhou, H., Jing, X., Pan R. and Zhang X. Ontology Theroy Based Process Planning and Supporting System under E-manufacturing, Przeglad Elektrotechniczny, 88(2012), no.1B, 107-110.

[3] Li, D., Finin, Y., Joshi, A., Pan, P., Cost, S., Peng, Y., Reddivari, P., Doshi, R., and Sachs, J., Swoogle: A Search and Metadata Engine for the Semantic Web, *Proceedings of CIKM2004*, (2004) 652–659.

[4] DeMarco T. Controlling Software Projects. Management, Measurement and Estimation, *Yourdon Press*, 1982.

[5] Tartir S, Arpinar IB, Moore M, Sheth AP, Aleman-Meza B, OntoQA: Metric-based ontology quality analysis, *Proceedings of IEEEWorkshop on Knowledge Acquisition from Distributed, Autonomous, Semantically Heterogeneous Data and Knowledge Sources*, 2005.

[6] Ma, Y., Wu, H., Ma, X., Jin, B., Huang, T. and Wei, J., Stable cohesion metrics for evolving ontologies, J*ournal of Software Maintenance and Evolution: Research and Practice*, 23(2011), No.5, 343-359

[7] Baader, F., et al., The description logic handbook: theory Implementation and Applications, *Cambridge University Press*, 2003.

[8] Chae, H.S., Kwon, Y.R., Bae, D.H., A cohesion measure for object-oriented classes, *Software Practice and Experience*, 30(2000), no.12, 1405-1431.

[9] Ma Y., Towards Stable Semantic Ontology Measurement. *Proceedings of International Semantic Web Conference (ISWC2010)*, (2010) 21-24.

[10] Motik, B., Reasoning in Description Logics using Resolution and Deductive Databases. *PhD Thesis, University of Karlsruhe, Karlsruhe*, Germany, 2006

[11] Gangemi A, et al., Qood grid: A metaontology based framework for ontology evaluation and selection, *Proceedings of Workshop EON2006*, May 2006.

[12] Lozano-Tello A, Gomez-Perez A, OntoMetric: A method to choose the appropriate ontology, *Journal of Database Management*, 15(2004), no.2,1-18.

[13] Oh S., Yeom H.Y., Ahn J.,Cohesion and coupling metrics for ontology modules. *Information Technology and Management*, 12(2011), no.2, 81-96.

[14] Deng X, Volker H, Nematollaah S, Measuring inconsistencies in ontologies, Proceedings of ESWC2007, (2007) 326-340.

[15] Grau B., et al. Completeness Guarantees for Incomplete Ontology Reasoners: Theory and Practice. Journal of Artificial Intelligence Research, 43(2012), 419-476.

***Authors**: Associate Professor Dr. Yinglong Ma, School of Control and Computer Engineering, Electric Power University, Beijing 102206, China, E-mail:gtmikema@gmail.com; Master Student, Yongming SHANG, School of Control and Computer Engineering, Electric Power University, Beijing, China; Professor Dr. Ke Lu, University of Chinese Academy of Sciences, Beijing 100049, China. E-mail:luk@ucas.ac.cn.*

The correspondence address is:
e-mail: gtmikema@gmail.com