# Survey of Measures for the Structural Dimension of Ontologies

Luis Reynoso, Marcelo Amaolo, Claudio Vaucheret
Faculty of Computer Science
*University of Comahue, Argentina*
*Buenos Aires 1400 (8300) Neuquén*
{luis.reynoso, marcelo.amaolo,claudio.vaucheret}@fai.uncoma.edu.ar

Mabel Álvarez
*Patagonia San Juan Bosco University, Argentina*
*Belgrano y Rawson (9100) Trelew, Chubut*
mablop@speedy.com.ar

## Abstract

*Different authors from literature had argued that measures for ontologies can help: to select a suitable ontology for the user needs, to improve dynamic web service composition and to predict the completed system's overall quality. However, the majority of the ontologies' measures go no further than their definitions. We have compared a set of 51 measures according to minimal criteria that a measure must fulfill. In order to do a coherent comparison of their definitions and their intents we have formalized the measures using Object Constraint Language (OCL) upon the Ontology Definition Model (ODM). The formalization of the measures help to avoid the misunderstanding and misinterpretation introduced when measures are informally defined using natural language. The formal definitions upon a OMD metamodel assure that measures capture the concepts they intend for and could facilitate the implementation of measures extraction tools.*

**Keywords:** Ontology Quality. Ontology measures. Structural Measures, Semantic Web.

## 1. INTRODUCTION

Ontologies are playing an increasingly important role in knowledge management and the Semantic Web and to achieve several purposes of cognitive informatics [1], [2], [3], [4]. They represent a shared conceptualization of a community, due to the fact they provide a shared knowledge model to semantic-driven application in the Internet. Many authors had argued about the importance of assessing the ontology quality: (1) To allow the ontology developer to automatically recognize areas that might need more work [5]. (2) To anticipate and reduce future maintenance requirements [6]. (3) To allow ontology users to select among different ontologies an ontology that fits their needs, or to detect parts of the ontology which might cause problems [5]. Quality assessment of ontologies can be measured in different dimensions [5], [7]. When ontologies are built from scratch: Noy et al. [8] details a guide for developing the ontology, Sugiura et al. [9] provides an environment for its development, Parsia et al. [10] defines a logical model that can be used to evaluate the ontology quality and to detect any problem. When ontology users need to select an ontology that fits their needs, they can use a the Noy et al framework [11] for comparing ontology schemas, they can use the Lozano-Tello framework [12] which provides five dimensions of ontology characteristics. Different ontology representation languages have been used to store domain knowledge allowing reuse, share and interchange of that knowledge. The Web Ontology Language OWL, an open standard developed by the W3C Web Ontology Working Group is the more known modeling language to define and instantiate web ontologies. OWL is based on RDF /RDFS (Resource Description Framework and RDF schema).

In this paper we analize 51 measures defined by different authors: Tartir et al. [13], Orme et al. [14], Yao et al. [6] and Gangemi et al. [15],[7]. We describe a comparison of the ontology measures related to structural aspects. We compared the ontology measures according to the following criteria:

- Is the intent of the measure described in its definition?, It is described how the value of measure can

**Proc. 12th IEEE Int. Conf. on Cognitive Informatics & Cognitive Computing (ICCI*CC'13)**
**D.F. Hsu, Y. Wang, A.R. Rao, D. Zhang, W. Kinsner, W. Pedrycz, R.C. Berwick & L.A. Zadeh (Eds.)**
**978-1-4799-0783-0/13/$31.00 ©2013 IEEE**

be used or interpreted? Is any example provided?

- Is the measurement concept that the measure captures, validated? (theoretical validation)
- The formalism used to define the measures: Are the measures defined using a mathematical or formal underpinnings?
- Are the measures empirically validated? Which findings were obtained?
- Is any software tool used? Which findings were obtained?
- The measures can be applied to ontologies defined in RDFS and/or OWL?
- An evaluation of the measure definition against the measurement ontology is provided.

In order to compare the measures and to understand clearly their measure definition we specify the whole set of measures using OCL upon the ODM metamodel [16]. After the formalization was performed we evaluated which ODM metaclasses need to be used in each definition and we can compare the concepts captured by the measures.

This paper is organized as follows: Next section describes the measures of ontologies we selected to perform the study. Section 3 shows the formal definition of the measure using OCL upon ODM. Section 4 includes a comparison of the measures proposal of Tartir et al. [13], Orme et al. [14], Yao et al. [6] and Gangemi et al. [15],[7]. Finally, the last section presents some concluding remarks arising from these preliminary findings.

## 2. MEASURES FOR ONTOLOGIES

In this section we outline specific details of different set of ontology measures. The measures of Tartir et al. [13], the coupling measures of Orme et al. [14], the cohesion measures of Yao et al. [6] and the measures of Gangemi et al. [15],[7]. A comparison of the main aspect is provided at the end of this section.

### 2.1 Measures of Tartir et al.

The authors proposed in [13] a set of metrics to analyze ontology schemas and their populations. The set of metrics are included in a model called OntoQA. The intent of the authors is to define measures to serve as a means to evaluate the quality of a single ontology or to compare ontologies. The metrics can be used for ontologies using RDFS or OWL document and instances defined in RDF file. Regarding the metrics they proposed, they are defined using a mathematical formalism defined in [17]. Neither the theoretical validation nor the measurement concept captured by each metric (except Coh measure) are provided. Each measure describes its purpose and a clear interpretation of how the measure is used is detailed, and the set is divided into two related categories: schema metrics and instance metrics (which are detailed in the following subsections). There are no examples of how the value is obtained in [5]. The measures are implemented in a Java-based prototype. The tool was used in [5] to exemplify the use of measures in a comparison among three ontologies. There is no empirical validation of the measures.

#### 2.1.1 Schema metrics

Their goal is to evaluate the ontology design and its potential for rich knowledge representation. They provide three metric (see Table 2) to indicate the richness of an ontology schema, assuming to the following arguments: An ontology that contains many relations other than class-subclass relations is richer than a taxonomy with only class-subclass relationships (RR measure). The higher number of attributes (slots) an ontology defines, the more knowledge the ontology conveys (AR measure). The distribution of information across different levels of the ontology inheritance tree or the fanout of parent classes is a good indicator of how well knowledge is grouped into different categories and subcategories in the ontology (IR measure). The measure distinguishes an ontology with a very detailed type of knowledge (vertical shape) from another representing a wide range of general knowledge (horizontal shape).

#### 2.1.2 Instance Metrics

They assess the effective use of the ontology to represent the knowledge modeled through the evaluation of the way data is placed within the ontology.

- A measure obtains how many instances are related to classes defined in the schema. It is expected that the data represents most of the knowledge in the schema (CR measure).
- A way of measuring whether the instances are insufficient to represent all the knowledge in the schema is to obtain the average number of instances per class (P measure).
- It is expected that the instances are connected between each other forming a connected graph. Otherwise, an unconnected graph would indicate that

the certain areas need more instances (Coh measure).

- A measure for ontology developers is provided in order to help them to identify which areas of the schema are in focus when instances are extracted. The distribution of instances of a class is defined as the importance of a class (Imp measure). It takes into account the number of instances that a class or the subtree rooted at that class has compared to the total number of instances.

- When an ontology developer knows an expected number of instances of a class, the F measure provides an indicator of how well the data extraction was to the expected values.

- The distribution of a particular inheritance tree can be evaluated. Very specific domains are depicted in a vertical inheritance shape whereas classes in an ontology that represents a wide domain are depicted in a horizontal inheritance shape (IRc measure).

- An indicator of how much of the properties of a class in the schema is actually being used at the instance level is provided by RRc measure. This measure indicates how many relationships of the class relationship in the schema are used by the instances.

- An indicator of how focal some classes are is provided by Cn measure. This measure is similar to Imp measure and valuates the popularity of instances of a class and it helps to identify whether a class plays a central role in the ontology.

A Java-based prototype was used to implement the metrics. The tool was used to evaluate three existent ontologies showing the utility of three metrics. The results only show how the class importance and the class connectivity metrics manifest the most important and connected classes respectively. Beside readability classes are detected.

### 2.2 Measures of Orme et al.

They define in [14] three coupling measures for ontologies. They argue that measuring the level of coupling earlier in the software life cycle can help to predict the ontology-based system quality. The measures are defined but the intent of each measure is not clearly specified, all of them captures the coupling according to the introduction of the work. Examples were provided.

The measures NEC, REC and RI (see Table 2) represent the number of distinct external classes that are used to define of classes and properties in the ontology, the number of reference to that classes, and the number of includes at the top of the ontology respectively. The first two classes can be applied to ontologies defined in RDFS or OWL. However the last one is defined exclusively in terms of XML DTDs, theres is no characteristics of RDFs nor OWL included in the RI definition. The measures were theoretically validated according to the Kitchenham validation framework.

They developed a simple XML parser to calculate the metrics for 33 ontologies. The coupling of these ontologies was also determined according a scale of five linguistic labels by 18 evaluators. The average software development experience of evaluators was 3.5 years. The interrater reliability of evaluators was studied, i.e. how well they agreed with each other. As their reliability results consistent, the authors averaged the evaluator ratings and performed a correlation analysis between the averaged evaluator data with the computed NEC, REC and RI values. They found correlations showing that the metrics were statistically valid.

The metrics were also used in a case study about integration of ontologies used by bioinformatics and genomics researchers, which are characterized by the molecular structure of living things. Their goal was to reduce coupling that can cause failures in the runtime use of ontologies. The integration study was using: layered approaches (this approach creates new ontologies simply by referencing resources in other ontologies), non-layered approaches (the simplest approach in this category takes all or part of the references to the distributed ontologies and integrate or copies them into a single, integrated ontology), and hybrid integration (where the strategy applied is a balance of the last two). The metrics help to automate ontology integration and to indicate the coupling levels of different combination, allowing selection of a less-coupled integrated ontology. The case study evaluates the combinations of Human Development Anatomy and the Edinburgh Mouse Atlas Project Development Anatomy. The results show that RI should be more useful when the number of ontologies being integrated increase, and REC seems particular useful for measuring coupling and can help developers to make these wise choices.

### 2.3 Measures of Yao et al.

They define in [6] a set of three cohesion metrics for ontologies to measure the modular relatedness of ontologies (recall that in object-oriented software cohesion refers to the degree of the relatedness or consistency in functionality of the members in a class). The measures are: the number of root classes (NoR measure), the number of leaf classes (NoL measure) and the average depth of inheritance tree of Leaf Nodes (ADIT-LN measure). The relatedness of ontology classes is semantically related by the properties: if their properties are strongly related, a high cohesion is expected. The measures were defined to measure ontologies defined with OWL language. The idea behind the measures is using the measures as an indicator of separation of responsibilities and independence of components of ontologies.

The metrics are defined using the Maedche mathematical model [17]. In [6] a definition and an example for each measure are provided, however it is not explained how the value of the measure is used in practice. The metrics are theoretically validated using the Kitchenham et al. [18]. and Briand et al. framework [19]. Regarding the last framework applied there are many properties that the authors define as not applicable (null value and monotonocity properties for NoR, NoL and ADIT-LN measures). We suggest that the theoretical measure should be revised due to the fact NoR and NoL measures seem to be size measures and ADIT-LN measure seems to be a depth measure, according to Briand et al. framework [19].

The metrics are empirically validated using the same empiricial validation as Orme et al. [14] and the measures results are compared against the assessment performed by a human team of 18 evaluators in the same way as Orme et al do [14]. They compute interrater reliability of the assessment of 33 ontologies. Results reveal consistent agreement between the evaluators. Then, the correlation between the averaged evaluation ratings of evaluator for ontologies and cohesion metrics was determined. A correlation was statistically found using Person test.

### 2.4 Measures of Gangemi et al.

Gangemi et al. describes 32 structural measures in [15]. A summary of them is shown in [20]. As the authors argue in [7] the technical report does not include a complete set of examples of good/bad quality for each measure. The examples were deferred to future versions of their work. There is no explanation about the intent of defining each measure. There is no theoretical validation of the measures.

The list of measures are arranged into the following groups:

- The graph structure of an ontology in which only is-a arcs are considered can be measured by the following three groups:
  - Measures for depth and breadth measures: The depth takes into account the quantity of nodes from the root to the leaf in the graph whereas the breadth measure the cardinality of levels or generations in a graph. A generation is the set of all sibling nodes sets that share the same distance from the root node, i.e. they have the same level in the graph.
  - Measures for tangledness: Classes within an ontology can inherit from different classes. This measures which is computed using a graph of is-a arcs is related to the multihierarchical nodes in a graph.
  - Measures for fan-outness: Fan-outness is measured in graph of is-a arcs, and is related to the dispersion of graph nodes. The dispersion is evaluated for leaf as well as internal nodes in the graph (sibling nodes).
- Measures for differentia specifica: They are concerned with the *rationale* behind sibling node sets, and it is measured looking for common relationships between sibling nodes through *non is-a* relationships.
- Measures for modularity: A module is considered as any subgraph of the ontology graph (is-a and non is-a arcs are included). Modularity is related to the asserted modules of a graph.
- Measures for logical and meta-logical adequacy: The measures of this group are computed against the graph which includes is-a and non is-a arcs. Several logical adequacy (consistent classes, anonymous classes, inverse relationships, etc) and meta-logical adequacy (meta-consistent classes e.g. the defined by OntoClean [21] measures) are defined.

## 3. FORMAL DEFINITION OF MEASURES FOR ONTOLOGIES

In this section we show the formal specification of measures for ontologies upon the ODM Metamodel [ODM, 2009]. Section 3.1, 3.2 and 3.3 include the formal definition of measures of Tartir et al., Yao et al., and Orme et al. respectively. The formal definition of Gangemi et al. measure are not included by problems of space constraints.

### 3.1 Formal Def. of Tartir's Measures

#### 3.1.1 Relationship Richness (CR) Measure

The relationship richness of a schema is defined as the ratio of the number of (non-inheritance) relationship, divided by the total number of relationships defined in the schema (the sum of the number of inheritance relationships and non-inheritance relationships).

```
context OWLOntology::relationship_schema(): Integer
body: self.owluniverse()->select(p | p.istypeOf(RDFProperty) and
(p.RDFSDomain->notEmpty() or p.RDFSRange->notEmpty()) )->size()

context: context OWLOntology::inheritance_relationship(): Integer
body: self.classes_of_the_ontology()->collect(p |
    p.RDFSubclassof()->size())->sum()

context: context OWLOntology::relationship_richness(): Integer
body: self.relationship_schema.div(self.relationship_schema() +
self.inheritance_relationship() )
```

#### 3.1.2 Attribute Richness (CR) Measure

'The attribute richness is defined as the average number of attributes per class. It is computed as the number of attributes for all classes divided by the number of class'. According to ODM an 'object property with no inverse and is not inverse functional' is mapped to as an attribute of a class in UML to OWL. This argument is used to calculate the attribute richness.

```
context OWLOntology::attribute_richness(): Integer
body: self.owluniverse()->select(c | c.istypeOf(OWLObjectProperty) and
    c.inverseProperty->isEmpty() and// no inverse
    c.OWLinverseOf->isEmpty() and
    not c.oclIsTypeOf(SymmetricProperty) and
    //not its own inverse
    not c.oclIsTypeOf(InverseFunctional)
    // not inverse functional
    )->size().div(
    self.classes_of_the_ontology()->size())
```

#### 3.1.3 Inheritance Richness (CR) Measure

'The inheritance richness of the schema (IR) is defined as the average number of subclasses per class'.

```
context: context OWLOntology::inheritance_richness(): Integer
body: self.classes_of_the_ontology()->collect(c |
    c.superclassof()->size())->sum().div(
    self.classes_of_the_ontology()->size())
```

#### 3.1.4 Class Richness (CR) Measure

'This measure is defined as the percentage of the number of non-empty classes (classes with instance) divided by the total number of classes defined in the ontology schema'. From the set of classes defined in an ontology, we select those classes that have an instance (typedResource links a class to a resource that is an instance of the class).

```
context OWLOntology::class_richness():Integer
def: self.classes_of_the_ontology()->collect(i |
    i.typedresource()->notEmpty())->size().div(self.qof_classes_ontology())
```

#### 3.1.5 Cohesion (Coh) measure

'The cohesion is defined as the number of connected components of the graph'. In this metric is not clear which are the possible connections that should be considered in order to determine that an individual is connected with another individual.

```
// instance_of_classes obtains the set of individual of classes
context OWLOntology::instance_of_classes(): Set(Individual)
body: self.owluniverse()->select(i | i.istypeOf(Individual) and
    c.classes_of_the_ontology()-> intersect(i.RDFType)->notEmpty() )

// connect verifies whether an Individual belongs to an existent component
context OWLOntology::connect(c:Individual): Set(Individual)
post: not resultatpre->includes(i | i.connected(c))
    implies result->append(c)

// obtain_components calls connect'operation for each individual
context OWLOntology::obtain_components(): Set(Individual)
body: self.instance_of_classes()->iterate(i | self.connect(i))

// cohesion is defined as the size of connected components
context OWLOntology::cohesion():Integer
body: self.obtain_components()->size()
```

#### 3.1.6 Class Importance (Imp) measure

'The importance of a class is defined as the percentage of the number of instances that belong to the inheritance subtree rooted by a class, compared to the total number of class instances'.

```
context OWLOntology::class_importance(c: OWLClass):Integer
pre: self.classes_of_the_ontology()->includes(c)
body:
self.owluniverse()->select(i | i.istypeOf(Individual) and
c.descendant()->intersect(i.RDFType)->notEmpty()) ->size().div(
self.owluniverse()->select(i | i.istypeOf(Individual) and
c.classes_of_the_ontology()->
```

intersect(i.RDFType)->notEmpty() )->size())

### 3.1.7 Fullness (F) measure of Tartir

'The importance of a class is defined as the actual number of instances that belong to the subtree rooted by a class, compared to the expected number of instances that belong to the subtree rooted at Ci'.

```
context OWLOntology::class_importance(c: OWLClass, en: Integer):Integer
pre: self.classes_of_the_ontology()->includes(c)
body:
self.owluniverse()->select(i | i.istypeOf(Individual) and
c.descendant()->intersect(i.RDFType)->notEmpty()) ->size().div(en)
```

### 3.1.8 Inheritance Richness of a class (IRC)

'The inheritance richness of a class is defined as the average number of subclasses per class in the subtree.'.

```
context: context OWLOntology::inheritance_richness(c:RDFSClass): Integer
: body: self.descendant(c)->collect(s | s.superclassof()->size())->
: sum().div( self.descendant(c)->size())
```

### 3.1.9 Relationship Richness of a class (Imp) measure

'The relationship richness of a class is defined as the percentage of the number relationships that are being used by instances that belong to the class compared to the number of relationships that are defined for the that class at the schema level'.

```
context OWLOntology::relationship_schema(c: OWLClass): Integer
: body: self. owluniverse()->select(p |
: p.istypeOf(RDFProperty) and p.RDFSDomain = c or
: p.RDFSRange = c)->size()

context OWLOntology::relationship_richness_of_class(c: OWLClass):Integer
body: c.classes_connectivity(c).div(
    self.relationship_schema(c))
```

### 3.1.10 Class Connectivity (Conn) measure

'The connectivity of a class is defined as the total number of relationships instances of the class have with instances of other classes'. We obtain the set of properties between instances of classes, where the property connects an instance of the class to another instance.

```
context OWLOntology::class_connectivity(c: OWLClass):Integer
pre: self.classes_of_the_ontology()->includes(c)
body: self. owluniverse()->select(p | p.istypeOf(OWLObjectProperty) and
p.statementWithPredicate()->notEmpty())->collect(x|
x.RDFSsubject.RDFType = c xor x.RDFSobject.RDFType = c )->size()
```

### 3.1.11 Readability of a Class (Conn) measure

'The readability of a class is defined as the sum of the number attributes that are comments and the number of attributes that are labels the class has'. If an annotation is associated with a class, the OWLAnnotationProperty is associated with an RDFStatement where its RDFObject is of Type RDFSClass.

```
context OWLOntology::(c: RDFSClass):Integer
body: self. owluniverse()->select(a |
p.istypeOf(OWLAnnotationProperty) and a.uriref()->notEmpty() and
a.statementwithpredicate()->notEmpty()
and a.statementwithpredicate()->size() = 1
and a.statementwithpredicate.RDFObject->oclisTypeof(RDFSClass))-
>size()

context OWLOntology::(c: RDFSClass):Integer
body: self. owluniverse()->select(a |
p.istypeOf(RDFSClass) and a.RDFScomment()->notEmpty())->
select(c.RDFScomment()->size())->sum() + self. owluniverse()->select(a
|
p.istypeOf(RDFSClass) and
a.RDFSlabel()->notEmpty())->select(c.RDFSlabel()->size())->sum()
```

## 3.2 Yao measure definition

### 3.2.1 Number of Root Class (NoR) measure

The number of root classes explicitly defined in the ontology.

```
context: OWLOntology::classes_in_hierarchy(): Set(OWLClass)
body: self.classes_of_the_ontology()->collect(c|
c.RDFSubclassof()->isEmpty() or c.superclassof())->isEmpty())

context: OWLOntology::root_classes(): Set(OWLClass)
body: self.classes_in_hierarchy()->collect(c|
c.RDFSubclassof()->isEmpty() and not c.superclassof())->isEmpty())

context: OWLOntology::number_of_root_classes(): Integer
body: self.root_classes()->size()
```

### 3.2.2 Number of Leaf Class (NoL) measure

'The number of leaf classes explicitly defined in the ontology'.

```
context: OWLOntology::leaf_classes(): Set(OWLClass)
body: self.classes_in_hierarchy()->collect(c| not
c.RDFSubclassof()->isEmpty() and c.superclassof())->isEmpty())
```

```
context: OWLOntology::number_of_leaf_classes(): Integer
body: self.leaf_classes()->size()
```

#### 3.2.3 Average Depth of Inheritance Tree of Leaf Nodes (ADIT-LN) measure

'The sum of depths of all paths divided by the total number of paths.

```
context: OWLOntology::average_depth_of_inheritance_tree(): Integer
body: self.leaf_classes()->collect(c|
self.level(self.root_classes(),c,0))->sum().div(self.leaf_classes()->size())
```

### 3.3 Orme's Measure definition

#### 3.3.1 NEC measure

Classes_of_the_Ontology operation obtains the classes defined in the ontology. A non-empty fragmentIdentifier associated with an empty uri implies that the uri is the xml:base (default namespace) of the document'. Classes_of_the_ontology counts the quantity of classes that belong to the ontology. An attribute class is no considered as a class.

```
context RDFSClass::is_attribute(c):Boolean
body: self.owluniverse()->select(c | c.istypeOf(OWLObjectProperty)
and c.inverseProperty->isEmpty() and// no inverse
c.OWLinverseOf->isEmpty() and
not c.oclIsTypeOf(SymmetricProperty) and
//not its own inverse
not c.oclIsTypeOf(InverseFunctional)
// not inverse functional
and c.RDFSDomain = c)

context OWLOntology::classes_of_the_ontology():Set(OWLClass)
def: self.owluniverse()->select(c | c.istypeOf(OWLClass)
c.uriref.uri->isEmpty() and
and not c.uriref.fragmentIdentifier->isEmpty() and
not c.is_attribute() )

context OWLOntology::qof_classes_ontology():Set(OWLClass)
def: self.classes_of_the_ontology()->count()
```

#### 3.3.2 REC measure

External_classes operation obtains the external classes of the ontology.

```
context OWLOntology::external_classes:Set(OWLClass)
def: self. owluniverse()->select(c |
c.istypeOf(OWLClass)-> self.classes_of_the_ontology()
```

The number of references to external classes in the ontology (REC measure of Orme et al.) is obtained through the reference_to_external_classes operation.

```
context OWLOntology::reference_to_external_classes
def: external_classes->collect(c| c.triplewithsubject)->size() +
external_classes->collect(c| c.triplewithobject)->size()
```

#### 3.3.3 RI measure

RDF/XML requires that URIrefs used as attribute values must be written out, rather than abbreviated as a QName. XML entities can be used in RDF/XML to improve readability in such cases, by providing an additional abbreviation facility for URIrefs. An XML entity declaration essencially associates a name with a string of characters. When the entity name is referenced elsewhere within an XML document, XML processors replace the reference with the corresponding string. This provides a way to abbreviate long strings such as URIrefs, and can help make XML documents containing such strings more readable.

For example, the ENTITY declaration (specified as part of a DOCTYPE declaration at the beginning of the RDF/XML document):

<!DOCTYPE rdf:RDF [<!ENTITY xsd "http://www.w3.org/2001/XMLSchema">]>

In RDF/XML documents, entities are generally declared within the document itself, i.e., using only an internal DTD subset (one reason for this is that RDF/XML is not intended to be validated, and non-validating XML processors are not required to process external DTD subsets). The use of XML entities as an abbreviation mechanism is optional in RDF/XML, and hence the use of an XML DOCTYPE declaration is also optional in RDF/XML. (For readers familiar with XML, RDF/XML is only required to be 'well-formed' XML). RDF/XML is not designed to be validated against a DTD by a validating XML processor.

## 4. COMPARISON OF MEASURES

We consider that the more clear definition of measures is the proposal of Tatir et al. Each measure was defined using a format: the intent of the measure, a mathematical definition, an example and it also describes the way of the measure value is used. Orme et al. also describe each item for each measure, but Tatir el al. is more rigorous and clear that Orme in their measure definition. The rest of the proposal lacks in one or two aspects.

Although Gangemi et al. provides a natural language and mathematical definition, each measure has

**Table 1. Ontologies' Measures**

| Criteria | Orme et al. | Tartir et. al | Yao et al. | Gangemi et al. |
|---|---|---|---|---|
| **Metric's Intent** | coupling | different intents | cohesion | different intents |
| **Metric's Explanation** | Yes | Yes | No | No |
| **Examples** | Yes | No | Yes | No |
| **Formal Definition** | Simple formalism | Model defined in [17] | No | No |
| **Theoretical Validation** | Yes, using [18] | No | Yes, using [18] and [19] | No |
| **Empirical Study** | Yes, XML parser/ case study | Yes, OntoQA. | Yes, XML parser call OMP | No |
| **Empirical Validation** | Yes | No | Yes | No |

no description of its intent with exception of the intent of the group each measure is contained. The intent of the group is also described for the Yao et al. cohesion measures, but not for each measure. Only the proposals of Orme et al. and Yao et al. describe a theoretical validation of the measures. However we believe that the theoretical validation of Yao et al should be revised.

The fourth one provides a empirical study or case study as an empirical validation of the measure. Empirical validation is an iterative process and not always is possible to know when is sufficient to prove that measures are empirically validated, however, we believe that the Tartir el al. empirical study is useful to show how the measure are used but not as an empirical validation of the utility in practice of the measures.

Besides, the large number of metrics presented in Gangemi et al. makes their model difficult to understand, for example what is the purpose of having different variation of the same measure? For example what the reason of defining absolute, average and maximal depth; absolute, average and maximal breath, etc. Why this three variations is constant in the definition of many measures?

Table 1 shows the main aspect of the outlined proposal for ontology measures.

Table 2 and 3 show the set of measures of each pro-

**Table 2. Comparison of Ontologies' Measures**

| source & acronym | Short Definition | sch./ inst. | inst./ class |
|---|---|---|---|
| [14] NEC | Number of External classes | sch. | ont. |
| [14] REC | Reference to External classes | Sch. | ont. |
| [14] RI | Referenced Includes | Sch. | ont. |
| [5] RR | Relationship Richness | sch. | ont. |
| [5] AR | Attribute Richness | sch. | ont. |
| [5] IR | Inheritance Richness | sch. | ont. |
| [5] CR | Class Richness | inst. | ont. |
| [5] P | Average Population | inst. | ont. |
| [5] Coh | Cohesion | inst. | ont. |
| [5] Imp | Importance | inst. | Class |
| [5] F | Fullness | inst. | Class |
| [5] IRc | Inheritance Richness of a class | inst. | Class |
| [5] RRc | Relationship Richness | inst. | Class |
| [5] Cn | Connectivity | inst. | Class |
| [5] Rd | Readability | inst. | ont. |
| [6] NoR | Number of Root Classes | Sch. | ont. |
| [6] NoL | Number of Leaf Classes | Scheme | ont. |
| [6] ADIT-N | Average depth of inheritance tree of Leaf Node | Sch. | ont. |
| [15] M1-3 | Absolute/Average/Max. depth | sch. | ont. |
| [15] M4-6 | Absolute/Average/Max. breath | sch. | ont. |
| [15] M7 | Tangledness | sch. | ont. |
| [15] M8 | Absolute leaf cardinality | sch. | ont. |
| [15] M9 | Ratio of leaf fan-outness | sch. | ont. |
| [15] M10 | Weighted ratio of leaf fan-outness | sch. | ont. |
| [15] M11 | Maximal leaf fan-outness | sch. | ont. |
| [15] M12 | Absolute sibling cardinality | sch. | ont. |
| [15] M13 | Ratio of sibling fan-outness | sch. | ont. |
| [15] M14 | Weighted ratio of sibling fan-outeness | sch. | ont. |

posal. First column includes the author of the measure and the measure acronym. Second column shows a short definition. The third column discriminates whether the measure is applied to the schema (show as *sch.*) or the instance (show as *inst.*) of an ontology. The fourth column it also distinguishes between measures defined for the ontology from measures which are defined for an ontology class. In addition, we can add that some proposals had explicitly express which is the measurement concept captured by the measure: NEC REC and RI are presented as coupling measures, Coh, NoR, NoL and ADIT-N are shown as cohesion measures.

**Table 3. Comparison of Ontologies' Measures (cont)**

| source & acronym | Short Definition | sch./ inst. | inst./ class |
|---|---|---|---|
| [15] M15 | Average sibling fan-outness | sch. | ont. |
| [15] M16 | Maximal sibling fan-outness | sch. | ont. |
| [15] M17-8 | Average sibling fan-outness without metric space/without lists of values | sch. | ont. |
| [15] M20 | Ratio of sibling nodes featuring a shared differentia specifica | sch. | ont. |
| [15] M21 | Ratio of sibling sets featuring a shared differentia specifica among elts | sch. | ont. |
| [15] M22 | Modularity rate | sch. | ont. |
| [15] M23 | Module overlapping rate | sch. | ont. |
| [15] M24 | Consistency ratio | sch. | ont. |
| [15] M25 | Generic complexity | sch. | ont. |
| [15] M26 | Anonymous classes ratio | sch. | ont. |
| [15] M27 | Cycle ratio | sch. | ont. |
| [15] M28 | Inverse relations ratio | sch. | ont. |
| [15] M29 | Class/relation ratio | sch. | ont. |
| [15] M30 | Axiom/class ratio | sch. | ont. |
| [15] M31 | Individual/class ratio | inst. | ont. |
| [15] M32 | Meta-consistency ratio | sch. | ont. |

## 5. CONCLUSIONS

The main contribution of this paper is the comparison of 51 measures for ontologies defined by Tartir et al. [13], Orme et al. [14], Yao et al. [6] and Gangemi et al. [15],[7]. For the purpose of reasoning about the definitions of the measures and to compare them using a shared framework, the set of measures were formal defined using OCL upon the ODM metamodel. A formal definition of the measure is useful to avoid misunderstanding and misinterpretation between the users of ontologies. Due to the fact that ontologies provides a shared conceptualization of a community, it is important to define their metrics properly, using a same metamodel to avoid interpretation and calculus errors. Refactoring techniques [22], [23] and MDA-based systems for ontology engineering [24] which improve the design of ontologies can take advantage of using the formal definition of measures. Measure value can be computed before and after the refactoring is applied, to evaluate the change according the quality of the ontology.

**ACKNOWLEDGMENTS**

This research is part of the 048/12 'Hacia el Fortalecimiento de la Sociedad en el Uso y Aplicación Geoespacial y las TICS' of Patagonia San Juan Bosco University and the 'Modelos y Tecnologías para Gobierno Electrónico' of Comahue University (Argentina).

## References

[1] C. Cao, Y. Siu, and Y. Sun, "Logical Connections of Statements at the Ontological Level," *Int. Journal of Cognitive Informatics and Natural Intelligence, 4(3), 59-85.*

[2] J. Ge, Z. Cheng, J. Pen, and T. Li, "An Ontology-based Method for Personalized Recommendation," *Proc. of the 11th IEEE Int. Conference on Cognitive Informatics and Cognitive Computing*, pp. 522–526, 2012.

[3] R. Harrison, D. Obst, and C. Chan, "Design of an Ontology Management Framework," *Proc. of the 4th IEEE Int. Conf. on Cognitive Informatics, ICCI*, pp. 260–266, 2005.

[4] S. Suqin Tang and Z. Cai, "Tourism Domain Ontology Construction from the Unstructured Text Documents," *Proc. of the 9th IEEE Int. Conf. on Cognitive Informatics, ICCI*, pp. 297–301, 2010.

[5] S. Tartir, I. Arpinar, M. Moore, A. Sheth, and B. Aleman-Mea, "OntoQA: Metric-based Ontology Quality Analysis," *IEEE ICDM 2005 Workshop on Knowledge Acquisition from Distributed, Autonomous, Semantically Heterogeneous Data and Knowledge Sources*, pp. 45–53, 2005.

[6] H. Yao, A. M. Orme, and L. Etzkorn, "Cohesion Metrics for Ontology Design and Application," *Journal of Computer Science. Science Publications*, vol. 1, no. 1, pp. 107–113, 2005.

[7] A. Gangemi, C. Catenacci, M. Ciaramita, and J. Lehmann, "Ontology Evaluation and Validation: An Integrated Formal Model for the Quality Diagnostic Task," *Technical Report. Laboratory for Applied Ontology. ISTC-CNR.*, 2005.

[8] N. Noy and D. McGuiness, "Ontology Development 101: A Guide to Creating your First Ontology," *Stanford Knowledge Systems Laboratory Technical Report KSL-0105 and Stanford Medical Informatics Technical Report SMI-2001.0880*, 2001.

[9] N. Sugiura, Y. Shigeta, N. Fukuta, N. Izumi, and T. Yamaguchi, "Towards On-the-fly Ontology Construction-Focusing on Ontology Quality

Improvement," *Proc. of the 1st European Semantic Web Symposium*, 2004.

[10] B. Parsia, E. Sirin, and A. Kalyanpur, "Debugging OWL Ontologies," *Proceedings of WWW*, 2005.

[11] N. Noy and C. Hafner, "The State of the Art in Ontology Design: A Survey and Comparative Review," *AI Magazine*, vol. 18, no. 3, pp. 53–74, 1997.

[12] A. Lozano-Tello and A. Gomez-Perez, "Ontometric: A Method to Choose the Appropiate Ontology," *Journal of Database Management*, vol. 15, pp. 1–18, 2004.

[13] S. Tartir, I. Arpinar, and A. Sheth, "Ontological Evaluation and Validation," *TAO-Theory and Applications of Ontology. Computer Applications. Springer-Berlin*, vol. 2, 2009.

[14] A. M. Orme, H. Yao, and L. Etzkorn, "Coupling Metrics for Ontology-based Systems," *IEEE Software*, vol. 23, no. 2, pp. 102–108, 2006.

[15] A. Gangemi, C. Catenacci, M. Ciaramita, and J. Lehmann, "Modelling Ontology Evaluation and Validation," *The Semantic Web: Research and Applications. 3rd European Semantic Web Conference*, pp. 140–154, 2006.

[16] OMG, "Ontology Definition Metamodel," *OMG, formal/2009-05-01, version 1.0*, 2009.

[17] A. Maedche, "Ontology Learning for the Semantic Web," *Kluwer International Series in Engineering and Computer Science*, vol. 665, 2002.

[18] B. Kitchenham, S. L. Pfleeger, and N. Fenton, "Towards a Framework for Software Measurement Validation," *IEEE Transaction on Software Engineering*, vol. 21, no. 12, pp. 929–944, 1995.

[19] L. C. Briand, S. Morasca, and V. R. Basili, "Property-Based Software Engineering Measurement," *IEEE Transaction on Software Engineering*, vol. 22, no. 1, pp. 68–86, 1996.

[20] G. A., C. Catenacci, M. Ciaramita, and J. Lehmann, "A Theoretical Framework for Ontology Evaluation and Validation," *Semantic Web Applications and Perspective. Proc. of the 2nd Italian Semantic Web Workshop*, 2005.

[21] N. Guarino and C. Welty, "Evaluating Ontological Decisions with Ontoclean," *Comm. ACM*, vol. 45, no. 2, pp. 61–65, 2002.

[22] D. A. Ostrowski, "Ontology Refactoring," *2012 IEEE Sixth Int. Conf. on Semantic Computing*, vol. 0, pp. 476–479, 2008.

[23] M. R. Hacene, S. Fennouh, R. Nkambou, and P. Valtchev, "Refactoring of Ontologies: Improving the Design of Ontological Models with Concept Analysis," *ICTAI (2)*, pp. 167–172, 2010.

[24] Y. Pan, G. Xie, L. Ma, Y. Yang, and Z. M. Qiu, "An MDA-Based System for Ontology Engineering," *RC23795, Computer Science, IBM Research Report*, 2005.