Applying Evaluation Criteria to Ontology Modules

Zubeida Casmod KHAN [1]
Council for Scientific and Industrial Research, Pretoria, South Africa

Abstract. Modularity has been increasingly used as a solution to assist with the use and maintenance of large ontologies. However, there is lack of evaluation methods available for determining the quality of an ontology module. While initial work has been done on producing a comprehensive list of evaluation characteristics that can be used to check the quality of a module, certain aspects are still unclear. For instance, the initial list has not been structured into different groups for different types of evaluation criteria. It is also unclear on how to apply these characteristics to a particular module, and the metrics have not been experimentally validated for some types of ontology modules. In this paper, we structure the comprehensive evaluation criteria into groups, provide practical examples on how to use the evaluation characteristics to assess a module, and validate the evaluation characteristics with a set of ontology modules.

Keywords. ontology evaluation, ontology metrics, ontology modules, ontology modularisation

1. Introduction

Over the last few years, there has been a growth in using modularity to assist with maintaining and using large ontologies. The general concept of modularity refers to dividing and separating the components of a large system such that modules can be recombined. Modularity is used to simplify and downsize an ontology for the task at hand; to modularise a large ontology into smaller manageable ontologies. Modularity has been successfully applied to a number of different ontologies to improve usability and assist with complexity. Examples include the myExperiment ontology [16], which is a collaborative environment where scientists publish and share their work-flows and experiment plans among groups, the Semantic Sensor Net ontology where there are various modules to describe sensors and observations [9], and BioTop ontologies for life sciences in which the principle of modularisation have been applied [22].

An issue concerning ontology modules is the lack of evaluation metrics. The existing works on evaluation metrics focus on only some metrics that suit the modularisation technique [20,21,26], and there is not always a quantitative approach to calculate them. Overall, the metrics are not comprehensive enough to apply to a variety of modules. It is therefore not clear on how to determine whether a module is of good quality.

[1]Correspondance address: Zubeida Casmod Khan, Department of Computer Science, University of Cape Town, South Africa, and Council for Scientific and Industrial Research, Pretoria, South Africa. zkhan@csir.co.za

We have already published initial work done on determining whether a module is a good or bad module with evaluation metrics [10]. The initial study [10] revealed 13 criteria from the literature, and 3 new ones, some of which were short of a metric for quantitative evaluation that have now been devised. In this work, we categorise the 16 criteria into 5 broader categories to structure them so that ontology developers can easily identify which ones to use for an application. We also demonstrate usage of the criteria with practical examples. Lastly, we perform an experimentation evaluation on a set of ontology modules using the evaluation criteria revealing how the modules measure in terms of quality.

The remainder of the paper is structured as follows. We discuss related works in Section 2. This is followed by a section on the evaluation criteria in Section 3. In Section 4 we discuss the types of modules we are going to evaluate. The experimental evaluation using the evaluation metrics is performed in Section 5. Finally we conclude in Section 6.

2. Related Works

d'Aquin et. al [3] present some criteria for evaluating ontology modules including logical criteria, e.g., local correctness, structural criteria e.g., size of module, and intra-module distance, software criteria, e.g., encapsulation, and independence, quality criteria, e.g., module cohesion, and relational criteria, e.g., connectedness, and inter-module distance. The study [3] also revealed that existing criteria for ontology modularisation evaluation is not sufficient, and that not all the proposed criteria can be used on all ontology modules. In other work, Loebe [14], proposed a number of requirements for logical modules, such as logical correctness and completeness. Loebe also acknowledges that the requirements do not hold for all applications and that specialised methods should be applied for different applications.

Tartir et. al [23] propose richness criteria to measure the quality of ontologies. This criteria is based on how rich the ontology is with regard to attributes and subclasses. For cohesion metrics, Yao et. al propose metrics such as the number of root classes, number of leaf classes, and average depth of inheritance tree of all leaf node [26]. These metrics, however, does not reveal how the entities are related in a module as compared to the original ontology. The work done by Oh et. al on cohesion metrics, however, does measure the strength of the relations in a module [18]. To measure the coupling of an ontology, researchers propose metrics based on the number of externally defined referenced concepts [19]. This, however, does not take into account external links that different modules share. Oh and Ahn [17] have improved on this to consider the external links between different modules based on whether the link is hierarchical or relational. However, their metric is simply a sum value of the number of each type of links between modules, which does not measure the complete interdependence of a module since it only considers one type of variable in the module.

3. Evaluation Metrics for Modules

In previous work, we already defined a list of evaluation criteria alongside their mathematical metrics for modularity [10]. We now group them into categories of criteria, and provide some examples demonstrating how to use them. For the set of criteria, let O be an ontology with a corresponding set of axioms, $Axioms(O)$, and M be a module with a corresponding set of axioms, $Axioms(M)$. Let i, j be arbitrary entities in an ontology.

3.1. Structural Criteria

Structural criteria are calculated based on the structural and hierarchical properties of the module. These criteria are calculated by inspecting the syntax of the ontology. It is usually based on counting components of the ontology such as axioms, entities, etc., and is a numerical value. Calculating structural criteria involves evaluating the size, relations, and placement of entities within a module. We now list the structural criteria, alongside practical examples.

Size: Size is the number of entities in a module (the number of classes, object properties, data properties, and individuals in a module) [2,3,18,21,20].

Relative size: The relative size is the size of the module, i.e., the number of entities in a module compared to the original ontology [12].

$$Relative\ size(M) = \frac{|M|}{|O|} \tag{1}$$

Example 1 *The GFO-Basic ontology [8] module contains 47 classes, 0 individuals, 41 object properties, and 0 data properties. The source ontology, GFO, contains 78 classes, 0 individuals, 67 object properties, and 0 data properties. Hence the relative size is* $\frac{47+0+41+0}{78+0+67+0} = 0.61$.

Appropriateness: Appropriateness is measured by mapping the size of an ontology module to some appropriateness function value between 0 and 1 to reflect the defect density [21].

$$Appropriate(x) = \frac{1}{2} - \frac{1}{2}cos(x.\frac{\pi}{250}) \tag{2}$$

where x is the number of axioms in the module. The authors propose the function with 250 axioms as it is based on the fact that the optimal size of software lines in a code is 200-300 lines [21].

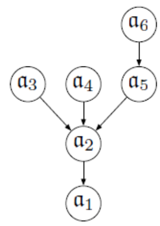
Example 2 *The Temporal Relations module of the DOLCE-ExtendedDnS descriptions and situations ontology [15] has 435 axioms. Therefore, based on the equation for calculating appropriateness, the value is* $\frac{1}{2} - \frac{1}{2}cos(435.\frac{\pi}{250}) = 0.16$.

Atomic size: The atomic size of a module is the average size of a group of interdependent axioms in a module. This is based on the notion of atoms in modules, which are defined as a group of axioms with dependencies between each other [24]. Dependent axioms are those that describe the same named entity.

$$Atomic\ Size(M) = \frac{|Axiom|}{|Atom|} \tag{3}$$

Example 3 *Consider the example in Figure 1 of an atomic decomposition [25]. The number of atoms in the example is 6 and there are 7 axioms in total. The atomic size is hence* $\frac{7}{6} = 1.17$. *This tells us that there is an average of 1.17 axioms per atom for the example.*

$\alpha_1 =$ '$\texttt{Animal} \sqsubseteq (= 1\texttt{hasGender}.\top)$',
$\alpha_2 =$ '$\texttt{Animal} \sqsubseteq (\geq 1\texttt{hasHabitat}.\top)$',
$\alpha_3 =$ '$\texttt{Person} \sqsubseteq \texttt{Animal}$',
$\alpha_4 =$ '$\texttt{Vegan} \equiv \texttt{Person} \sqcap \forall\texttt{eats}.(\texttt{Vegetable} \sqcup \texttt{Mushroom})$',
$\alpha_5 =$ '$\texttt{TeeTotaller} \equiv \texttt{Person} \sqcap \forall\texttt{drinks}.\texttt{NonAlcoholicThing}$',
$\alpha_6 =$ '$\texttt{Student} \sqsubseteq \texttt{Person} \sqcap \exists\texttt{hasHabitat}.\texttt{University}$',
$\alpha_7 =$ '$\texttt{GraduateStudent} \equiv \texttt{Student} \sqcap \exists\texttt{hasDegree}.\{\mathrm{BA}, \mathrm{BS}\}$'



Here the $\perp$-atoms in the AD contain the following axioms respectively: $\mathfrak{a}_1 = \{\alpha_1, \alpha_2\}$, $\mathfrak{a}_2 = \{\alpha_3\}$, $\mathfrak{a}_3 = \{\alpha_4\}$, $\mathfrak{a}_4 = \{\alpha_5\}$, $\mathfrak{a}_5 = \{\alpha_6\}$, $\mathfrak{a}_6 = \{\alpha_7\}$.

Figure 1. An ontology's atomic decomposition. See example 3 for details. Source: [25].

Intra-module distance: The intra-module distance in a module is the distance between entities in a module [3]. To measure it, Freeman's farness equation is used to measure the sum of distances to all other nodes [4]; the full equation is in [12].

$$\textit{Intra-module distance}(M) = \sum_{i}^{|M|} \textit{Farness}(i) \quad (4)$$

Relative Intra-module distance: The relative intra-module distance is the difference between entities in a module and entities in a source ontology [12].

$$\textit{Relative intra-module distance}(M) = \frac{\textit{Intra-module distance}(O)}{\textit{Intra-module distance}(M)} \quad (5)$$

Example 4 *Consider the example of a source ontology O and a module M shown in Figure 2, and the farness values in Table 1. Using the farness values in the intra-module distance equation, we calculated the intra-module distance of the source ontology O to be 32 and of the module M to be 16. The relative intra-module distance is* $\frac{32}{16} = 2$*, hence the module entities are twice as close as the original ontology.*

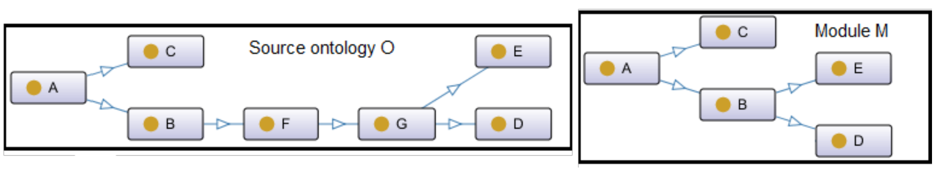


Figure 2. A source ontology and corresponding module for which we calculate intra-module distance. The arrows between the entities indicate the shortest-path relation between them.

Cohesion: Cohesion refers to the extent to which entities in a module are related to each other. [5,18,17,26]. To measure it, we use Oh et. al's equation [18]; the full equation is in [12].

Table 1. The farness values for the source ontology and corresponding module alongside inverse farness values the corresponding module.

	Farness, *O*						**Farness, *M***						**1/farness, *M***					
	A	**B**	**C**	**D**	**E**	$\sum$	**A**	**B**	**C**	**D**	**E**	$\sum$	**A**	**B**	**C**	**D**	**E**	$\sum$
A	-	1	1	4	4	10	-	1	1	2	2	6	-	$\frac{1}{1}$	$\frac{1}{1}$	$\frac{1}{2}$	$\frac{1}{2}$	3
B	1	-	0	3	3	7	1	-	0	1	1	3	$\frac{1}{1}$	-	0	$\frac{1}{1}$	$\frac{1}{1}$	3
C	1	0	-	0	0	1	1	0	-	0	0	1	$\frac{1}{1}$	0	-	0	0	1
D	4	3	0	-	0	7	2	1	0	-	0	3	$\frac{1}{2}$	$\frac{1}{1}$	0	-	0	$\frac{3}{2}$
E	4	3	0	0	-	7	2	1	0	0	-	3	$\frac{1}{2}$	$\frac{1}{1}$	0	0	-	$\frac{3}{2}$
Intra-module distance(O)						32	*Intra-module distance(M)*					16	*1/farness (M)*					10

$$Cohesion(M) = \begin{cases} \sum\limits_{E_i \in M} \sum\limits_{E_j \in M} \frac{SR(e_i, e_j)}{|M|(|M|-1)} & if |M| > 1 \\ 1 & otherwise \end{cases} \tag{6}$$

Example 5 *Consider module M, from Figure 2. The sum of all the 1/farness values is 10 as shown in Table 1. The number of entities in M is 5. Hence the cohesion value is as follows: Cohesion =* $\frac{10}{5(4)}$ *= 0.5.*

3.2. Logical Criteria

By definition, an ontology is: "a logical theory accounting for the intended meaning of a formal vocabulary, i.e. its ontological commitment to a particular conceptualisation of the world" [7]. As such, it is possible to evaluate ontology modules by the logical criteria that they hold. We now list the logical criteria, alongside practical examples.

Correctness: This states that every axiom that exists in the module also exists in the source ontology and that nothing new should be added to the module [14,1,3,20].

$$Correctness(M) = \begin{cases} true & if Axioms(M) \subseteq Axioms(O) \\ false & otherwise \end{cases} \tag{7}$$

Example 6 *The GFO-Abstract-Top ontology is a subset of the GFO ontology [8]. No new axioms have been added to the GFO-Abstract-Top ontology; it only contains those axioms which exist in the GFO ontology. Thus, the GFO-Abstract-Top ontology is logically correct. The GFO-Basic ontology, however, is a smaller module based on the GFO ontology but it also contains new axioms that do not exist in the GFO ontology. For instance, the entity* Processual Structure *exists in the GFO-Basic module but not in the source ontology, GFO. Thus, the logical correctness property does not hold for the GFO-Basic module.*

Completeness: Completeness is when the meaning of every entity in a module is preserved as in the source ontology [14,1,3,20].

$$Completeness(M) = \begin{cases} true & if \sum\limits_{i}^{|M|} Axioms(Entity_i(M)) \models Axioms(Entity_i(O)) \\ false & otherwise \end{cases} \tag{8}$$

Example 7 *Consider a source ontology, DOLCE-Lite [15], where the* endurant *entity is defined as follows:*

∗ *endurant* $\sqsubseteq \forall$ *part.endurant*
∗ *endurant* $\sqsubseteq$ *spatio-temporal-particular*
∗ *endurant* $\sqsubseteq \exists$ *participant-in.perdurant*
∗ *endurant* $\sqsubseteq \forall$ *specific-constant-constituent.endurant*
∗ *endurant* $\sqsubseteq \neg$ *quality*
∗ *endurant* $\sqsubseteq \neg$ *perdurant*
∗ *endurant* $\sqsubseteq \neg$ *abstract*

If DOLCE were to be modularised to create a branch module, containing only the branch of Endurant entities, with the removal of perdurant entities called DOLCE-endurants, then the endurant *entity is defined as follows:*

∗ *endurant* $\sqsubseteq \forall$ *part.endurant*
∗ *endurant* $\sqsubseteq$ *spatio-temporal-particular*
∗ *endurant* $\sqsubseteq \exists$ *participant-in.perdurant*

The meaning of the endurant *entity was not preserved in the module since the axiom* endurant $\sqsubseteq \forall$ specific-constant-constituent.endurant *existed in the original ontology but not in the module. Therefore the DOLCE-endurants module has a false value for the completeness metric.*

3.3. Relational Criteria

Relational criteria deal with the relations and behaviour that modules exhibit with other modules in a system of interrelated modules. We now list the relational criteria, alongside practical examples.

Inter-module distance: The inter-module distance in a set of modules has been described as the number of modules that have to be considered to relate two entities [2,3]. This is measured by calculating the sum of modules that have to be considered to relate two entities divided by the number of all possible relations in a set of modules.

$$\textit{Inter-module-distance(M)}= \begin{cases} \sum\limits_{E_i,E_j \in (M_i,,M_n)} \frac{NM(E_i,E_j)}{|(M_i,..,M_n)|(|(M_i,..,M_n)|-1)} & |(M_i,..,M_n)| > 1 \\ 1 & otherwise \end{cases} \tag{9}$$

where $NM(E_i,E_j)$ is the number of modules to consider to relate entities i and j. The product of $|(M_i,..,M_n)|(|(M_i,..,M_n)|-1)$ represents the number of possible relations between entities in a set of modules $M_i,.,M_n$.

Example 8 *Consider the set of inter-related modules in Figure 3. For each entity pair, we have the number of modules, NM, that have to be considered to relate them in Table 2. The sum of NM is 126. The number of entities is 9, hence the* $|(M_i,..,M_n)|(|(M_i,..,M_n)|-1)$ *value = 9(8). Thus the inter-module distance is* $\frac{112}{9(8)} = 1.75$*. For the set, it takes 1.75 modules to relate two entities in the set.*

Coupling: A measure of the degree of interdependence of a module [5,18,17,19]. To measure the coupling of a module, we calculate a ratio of the number of external links between a modules to every possible external link between modules.

Table 2. The number of modules, *NM*, that have to be considered to relate two entities in the set of modules.

	A	B	C	D	E	F	G	H	I	Sum
A	-	1	2	2	2	1	2	2	2	14
B	1	-	2	2	2	1	2	2	2	14
C	2	2	-	2	2	2	2	1	1	14
D	2	2	2	-	1	2	1	2	2	14
E	2	2	2	1	-	2	1	2	2	14
F	1	1	2	2	2	-	2	2	2	14
G	2	2	2	1	1	2	-	2	2	14
H	2	2	1	2	2	2	2	-	1	14
I	2	2	1	2	2	2	2	1	-	14
$NM(E_i, E_j)$										126

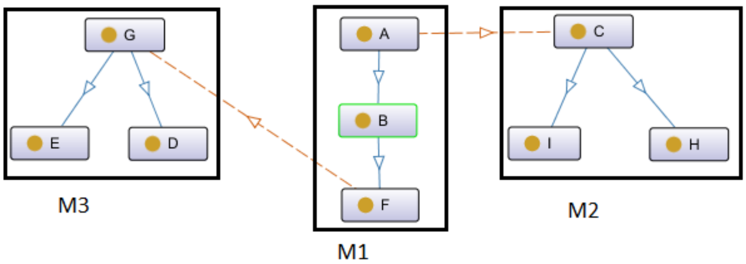


Figure 3. A set of modules with inter-related links. The plain arrow links between entities denote relations between entities in the same module while the red dotted arrow links denote relations between entities in different modules.

$$Coupling(M_i) = \begin{cases} \sum_{i=0}^{n} \sum_{\substack{j=0 \\ i \neq j}}^{n} \frac{NEL_{M_i,M_j}}{|M_i||M_j|} & NEL_{M_i,M_j} > 0 \\ 0 & otherwise \end{cases} \quad (10)$$

where $|M_i|$ is the number of entities in the current module and $|M_j|$ is the number of entities in a related module in the set of n modules.

Example 9 *Consider module M_1 from the set of inter-related modules in Figure 3. The number of external links that have to be considered to relate M_1 to other modules in the set, is 2. The number of possible external link between a module M_1 and the other modules in the system is calculated as follows: $|M_1|(|M_2|) + |M_1|(|M_3|) = 18$. Hence the coupling($M_1$) = $\frac{2}{18} = 0.11$ which indicates a low interdependence toward other modules in the system.*

Redundancy: Redundancy is the duplication of axioms within a set of ontology modules [21]. To measure redundancy in a set of modules, we calculate the fraction of duplicated axioms.

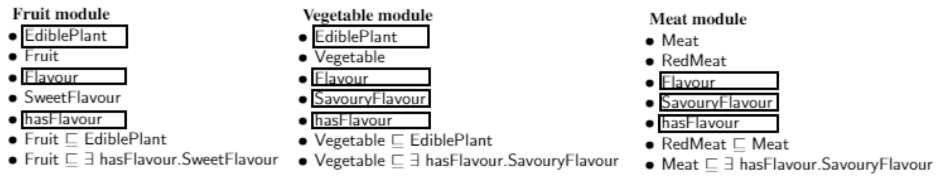


Figure 4. The axioms from a toy food ontology. The axioms in blocks are those that have been repeated more than once.

$$Redundancy(M) = \frac{(\sum_{i=1}^{k} n_i) - n}{\sum_{i=1}^{k} n_i} \tag{11}$$

where $\sum_{i=1}^{k} n_i$ is the total number of axioms and n is the number of distinct axioms in a module. The resulting fraction is a value of redundancy.

Example 10 *Consider the class declarations and axioms in the set of modules with no inter-related links that have been partitioned from a food ontology as shown in Figure 4. There are 3 ontology modules: Fruit, Vegetable, and Meat. Axioms that have been repeated more than once (redundant axioms) are shown in blocks. From the three modules, there is a total of 21 axioms, i.e., the Ax_t value is 21. There are 15 distinct axioms that exist in the set of modules (these axioms exist at most once and are those that are not blocks), hence Ax_d is 15. The redundancy of the set of partitioned modules is thus $\frac{21-15}{21} = 0.29$. Hence, 29% of the axioms in the set of modules are redundant.*

3.4. Information Hiding Criteria

Ontology modules sometimes deal with hiding aspects of the source ontology from the module for privacy and simplification reasons. Information hiding within modules assesses whether the module encapsulates all the information in the module such that the privacy is preserved for each module. We now list the information hiding criteria, alongside practical examples.

Encapsulation: Encapsulation is a metric that holds when "a module can be easily exchanged for another, or internally modified, without side-effects on the application can be a good indication of the quality of the module" [3]. It is measured using the number of axioms in the given module and the number of axioms that occur in both the given module and related modules.

$$Encapsulation(M\_i) = 1 - \frac{\sum_{j=1}^{n-1} \frac{|Ax_{ij}|}{|Ax_i|}}{n} \tag{12}$$

Example 11 *Consider the 3 ontology modules Fruit, Vegetable, and Meat, from Example 10. We calculate the encapsulation of the Fruit module as follows. There are 7 axioms in the Fruit module, i.e., the $Ax_i = 7$. In the Vegetable module, there* *are 3 overlapping axioms, i.e., they also exist in the Fruit module. In the Meat module, there are 2 overlapping axioms, i.e., they also exist in the Fruit module. Hence, the Encapsulation(Fruit) is calculated as* $1 - \frac{\frac{3}{7}+\frac{2}{7}}{3} = 0.76$. *Thus, 0.76 (76%), or a large amount of the domain knowledge is encapsulated in the Fruit module but the complete privacy of the Fruit module is not preserved.*

Independence: Independence evaluates whether a module is self-contained and can be updated and reused separately [3]. A module is independent if it has an encapsulation value of 1 and a coupling value of 0.

$$Ind(M\_i) = \begin{cases} true & Encapsulation(M_i) = 1 \text{ and } Coupling(M_i) = 0 \\ false & otherwise \end{cases} \tag{13}$$

where $|M_i|$ is the number of entities in the current module and $|M_j|$ is the number of entities in a related module in the set of n modules.

Example 12 *Consider the 3 ontology modules in Example 10. We have already worked out the encapsulation value for the Fruit module in Example 11 as 0.76. There are no inter-related links between the modules hence the coupling value is 0. Since the encapsulation value is not 1, the conditions for independence do not hold for the Fruit module hence it is not independent.*

3.5. Richness Criteria

The richness or amount of information in an ontology is designed as one aspect to measure the quality of an ontology. For modules, this is important to measure in cases where abstraction is employed to compare the granularity of the source ontology to that of the module. Tartir et al. [23] propose measurable richness schema metrics. We now list the richness criteria, alongside practical examples.

Attribute richness: Attribute richness is defined as the average number of attributes per class [23].

$$AR(M) = \frac{|att|}{|C|} \tag{14}$$

where *att* is measured by the number of data properties in the module $|DP|$ and $|C|$ is the number of classes in the module. In an ontology, an attribute is used to describe an entity and each attribute, or data type, has a name and value.

Example 13 *The pizza ontology has no data properties (attributes) defined. The AR value is 0, therefore there is no attribute richness in the pizza ontology.*

Inheritance richness: Inheritance richness is defined as the number of subclasses per class in an ontology [23].

$$IR_S(M) = \frac{\sum_{C_i \in C} |H^C(C_1, C_i)|}{|C|} \tag{15}$$

where $|H^C(C_1, C_i)|$ is the number of subclasses per class and $|C|$ is the total number of classes in the ontology.

Example 14 *Refer back to ontology O and module M from Figure 2. For module M, the entities which have subclasses are entity A with 2 subclasses, and entity B with 2 subclasses. Hence the sum of these subclasses is* $2 + 2 = 4$. *There are 5 classes in total in M. The inheritance richness value for M is thus* $\frac{4}{5} = 0.8$. *Using the same method we work out the inheritance richness value for ontology O, which is* $\frac{6}{7} = 0.85$.

To assist with the lack of evaluation metrics and corresponding formulae in ontology modules, we presented a comprehensive list of evaluation criteria for modules together with examples on how to operationalise them for ontology modules. To put context to the values for the metrics, experimentation was performed in other work stating which values are appropriate for the metrics for particular module types [12]. In Section 5, we show how each metric is used and what some expected values are.

4. Types of Modules for Evaluation

In this work, we focus on evaluating certain types of ontology modules, i.e., those that are lacking from an existing ontology modularisation framework and experiment [11]. The modules were generated using the NOMSA modularisation tool [2]. We briefly describe each module type here with its corresponding abbreviation which we will use to describe them in the remainder of the paper.

- Axiom abstraction (AxAbs): This is a module containing hierarchical relations between entities, i.e., other relations are removed resulting in a bare taxonomy.
- Vocabulary abstraction (VocAbs): This is a module where certain types of entities are removed, for instance, the object properties or data properties of an ontology.
- High-level abstraction (HLAbs): This is a module where entities at a higher level in the hierarchy have precedence over the others.
- Weighted abstraction (WeiAbs): This is a module where weighting is assigned to entities that are referenced by axioms more than others.
- Feature expressiveness (FeatExp): This is a module where some axioms of the ontology are removed based on its language features.

5. Experimental Evaluation

In order to uncover information about how evaluation metrics relate to ontology modules, we use the Tool for Ontology Modularity Metrics (TOMM) software tool [12] which encompasses all the evaluation metrics described in Section 3. In this experiment, we evaluate the module types from Section 4, in terms of its quality.

5.1. Materials and Methods

The method for the experiment is as follows:

---
[2] `http://www.thezfiles.co.za/modularisation/`

1. Take a set of modules.
2. Run the TOMM metrics tool [12] for the modules to acquire module metrics.
3. Conduct an analysis from the metrics for each module.

The materials used for the experiment were as follows: TOMM metrics tool [12], and a set of 128 ontologies that had been modularised using NOMSA ontology modularisation tool [2] but derived from the set of ontologies described elsewhere [6,13]. Our tests were carried out on a 3.00 GHz Intel Core 2 Duo PC with 4 GB of memory running Windows 7 Enterprise. All the test files are available at `http://www.thezfiles.co.za/modularisation/testfiles_NOMSA.zip`.

5.2. Results

Each of the modules were evaluated using the metrics that were implemented in the TOMM tool and the results for the numerical metrics are shown in Table 3. The atomic sizes of the modules indicates that there are on average between 2.34- 3.80 axioms that are grouped together in an atom for the modules. The appropriateness, which maps the size of the ontology module to a function is less than 0.3 for all the modules. Seeing that 1.0 is the optimal value for appropriateness, all the modules perform poorly for this metric. For a module to have an optimal value, it must have a value close to 250 axioms. The intra-module distance values which indicate the distance between entities in a module differ considerably, with the HLAbs modules having the lowest value of 142 698.4 and the AxAbs having the highest value of 866 354.60. The cohesion of a module indicates how closely related its entities are to each other, with higher values having a large number of relations among entities. The cohesion is small for all the modules in the set (less than 0.07). Most of the modules in this set do not contain attributes, as the attribute richness is less than 1 for all the modules. The number of subclasses per class is between 2.72 to 4.84 as indicated by the inheritance richness for the modules. In this set, all the modules were considerably smaller than the original ontologies, as indicated by relative size values that are less than 1. WeiAbs modules are the smallest at 0.26 (26% the size of the original ontology) while VocAbs modules are 0.85 (85% the size of the original ontology). It is also important to determine whether the entities in the modules have moved closer to each other in a module compared to the original ontology. AxAbs modules, VocAbs modules, and FeatExp modules have values less than 1 for the relative intra-module distance, which indicate that their entities are further in relation to the entities of the original ontology. While in HLAbs and WeiAbs the intra-module distance is larger, indicating that for these modules, the entities have moved closer as compared to the original ontology.

We have discussed the evaluation metric values for the set of ontology modules in isolation, i.e., without an indication of whether the modules in question appear to be good or of poor quality. To check if the modules are of good quality we refer to an existing benchmark dependency between the proposed ontology metrics and various types of ontology modules. The notion here is that for each module type, there is a pattern or dependencies between type and some set of metrics. This is shown in Table 4 with expected values for each type of module. The values that are underlined were not met for this experiment. For instance, reading Table 4, it states that for WeiAbs modules the cohesion value of range 0-0.25 and the relative size value of range 0.26-0.5 is met. However, for AxAbs, the cohesion value is met but the correctness value is not met.

Table 3. The average values for the metrics for all the generated modules; app. = appropriateness, cohes = cohesion, AR = attribute richness, IR = inheritance richness, rel = relative, IMD = intra-module distance.

	Size	Atomic size	App.	Intra-module distance	Cohes	AR	IR	Rel. size	Rel. IMD
AxAbs modules	238.04	2.34	0.19	866345.6	0.06	0.49	4.84	0.71	0.68
VocAbs modules	443.38	3.24	0.19	848372.2	0.06	0.45	4.78	0.85	0.79
HLAbs modules	202.77	3.48	0.24	166797.1	0.03	0.47	4.86	0.67	18.66
WeiAbs modules	138.58	3.40	0.30	142698.4	0.07	0.39	2.72	0.26	3.96
FeatExp modules	291.89	2.44	0.18	757305.1	0.06	0.25	4.80	0.72	0.70
Original ontologies	464.67	3.80	0.15	1866430	0.04	1.04	4.78	-	-

Table 4. The benchmark metrics and values for each module type. The metrics that are underlined are ones that fail for the set of modules that were evaluated.

	Cohesion	Correctness	Appropriateness	Relative size
AxAbs modules	0.00-0.25	true	-	-
VocAbs modules	0.00-0.25	true	0.75-1.00	-
HLAbs modules	0.00-0.25	-	0.75-1.00	-
WeiAbs modules	0.00-0.25	-	-	0.26-0.50
FeatExp modules	0.00-0.25	-	-	-

We now examine the failed metrics. For appropriateness, it appears that the number of axioms in a module need to be between 167- 333 to have a value of between 0.75-1. In some cases, the original ontologies were already less than 167 axioms in size, so there could never reach an appropriate value. For the case of ontology modules, which by definition are a reduction of an ontology, this appropriateness metric needs to be re-designed to include modules that are less than 167 axioms. The next failed metric is the correctness value, which needs to be true, but tested false for some modules, meaning that some new axioms were added to the modules, i.e., axioms that were not from the original ontology. An inspection of the log files for the metrics revealed that OWL enumeration was used to declare individuals in an original ontology. In resulting modules, when the collection of individuals were broken up, they were re-represented as named instances using class identifiers. This means that new knowledge was not added to the module, but because of language syntax, it appears so. The ontology metrics tool therefore needs to recognise that this is not a new axiom to measure the correctness value accurately.

5.3. Discussion

The evaluation criteria for ontology modules were already presented in previous work [10]. However, there was limited support for operationalising these metrics. In this paper, we provide practical examples on how the metrics can be applied to an ontology module to measure them. We also structured and grouped together the list of evaluation criteria to various higher level categories to differentiate those that examine, say structural aspects of a module (structural criteria) to those that examine how well the modules hide certain information of the source ontology (information hiding criteria). This could aid ontology developers in selecting the relevant evaluation criteria for their use-case.

We performed an experiment using various types of modules that were generated using NOMSA modularisation tool. The modules were evaluated with TOMM to reveal their metrics. The metrics indicate important information such as, the rate at which entities in a module move closer to each other (using the relative intra-module distance metrics), whether a module is rich with attributes, at what rate the size of the module differs as compared to the source ontology. These metrics are important for gauging how well the modules would fare with visualisation and comprehension tools or if there would be some performance issues due to redundancy, among other cases.

6. Conclusion

Initial work on evaluation criteria was refined by grouping the evaluation criteria list into categories to structure them to aid ontology developers. We summarised the list of evaluation criteria, together with practical examples on how to use them. An experimental evaluation revealed how the metrics can be used to evaluate modules and uncover information about how well certain modules fare for certain applications. We also identified a problem with the appropriateness value, which cannot be applied in some cases if a module has a small axiom size which needs to be changed. Another problem that was identified is that some knowledge is recognised as new knowledge by the evaluation metrics tool due to OWL representation issues.

References

[1] Cuenca Grau, B., Parsia, B., Sirin, E., Kalyanpur, A.: Modularity and Web Ontologies. In: 10th International Conference on Principles of Knowledge Representation and Reasoning (KR'06). pp. 198–209. AAAI Press (2006), june 2-5, Lake District, United Kingdom.

[2] d'Aquin, M., Schlicht, A., Stuckenschmidt, H., Sabou, M.: Ontology modularization for knowledge selection: Experiments and evaluations. In: Wagner, R., Revell, N., Pernul, G. (eds.) The 18th International Conference on Database and Expert Systems Applications (DEXA'07). Lecture Notes in Computer Science, vol. 4653, pp. 874–883. Springer (2007), regensburg, Germany, September 3-7, 2007

[3] d'Aquin, M., Schlicht, A., Stuckenschmidt, H., Sabou, M.: Criteria and evaluation for ontology modularization techniques. In: Modular Ontologies: Concepts, Theories and Techniques for Knowledge Modularization, Lecture Notes in Computer Science, vol. 5445, pp. 67–89. Springer (2009)

[4] Freeman, L.C.: Centrality in social networks conceptual clarification. Social Networks 1(3), 215–239 (1978)

[5] García, J., Peñalvo, F.J.G., Therón, R.: A survey on ontology metrics. In: Lytras, M.D., de Pablos, P.O., Ziderman, A., Roulstone, A., Maurer, H.A., Imber, J.B. (eds.) Third World Summit on the Knowledge Society, (WSKS'10). Communications in Computer and Information Science, vol. 111, pp. 22–27. Springer (2010), corfu, Greece, September 22-24

[6] Gardiner, T., Tsarkov, D., Horrocks, I.: Framework for an automated comparison of description logic reasoners. In: The 5th International Semantic Web Conference (ISWC'06). Lecture Notes in Computer Science, vol. 4273, pp. 654–667. Springer (2006), athens, GA, USA, November 5-9, 2006

[7] Guarino, N.: Formal ontology and information systems. In: Guarino, N. (ed.) Proceedings of the 1st International Conference on Formal Ontology in Information Systems (FOIS'98). pp. 3–15. Frontiers in Artificial intelligence and Applications, IOS Press, Amsterdam, The Netherlands (1998)

[8] Herre, H.: General Formal Ontology (GFO): A foundational ontology for conceptual modelling. In: Theory and applications of ontology: computer applications, pp. 297–345. Springer (2010)

[9] Janowicz, K., Compton, M.: The Stimulus-Sensor-Observation Ontology Design Pattern and its Integration into the Semantic Sensor Network Ontology. In: 3rd International Workshop on Semantic Sensor Networks. CEUR Workshop Proceedings, vol. 668. CEUR-WS.org (2010), november 7, Shanghai, China.

[10] Khan, Z.C.: Evaluation metrics in ontology modules. In: 29th International Workshop on Description Logics (DL'16). CEUR Workshop Proceedings, vol. 1577. CEUR-WS.org (2016), 22-25 April 2016, Cape Town, South Africa

[11] Khan, Z.C., Keet, C.M.: An empirically-based framework for ontology modularisation. Applied Ontology 10(3-4), 171–195 (2015)

[12] Khan, Z.C., Keet, C.M.: Dependencies between modularity metrics towards improved modules. In: 20th International Conference on Knowledge Engineering and Knowledge Management (EKAW'16). pp. 19–23. Lecture Notes in Artificial Intelligence LNAI, Springer (2016), 19-23 November 2016, Bologna, Italy

[13] Ławrynowicz, A., Keet, C.M.: The TDDonto tool for test-driven development of dl knowledge bases. In: The 28th International Workshop on Description Logics (DL'16). CEUR-WS, vol. 1577 (2016), 22-25 April 2016, Cape Town, South Africa

[14] Loebe, F.: Requirements for logical modules. In: Haase, P., Honavar, V., Kutz, O., Sure, Y., Tamilin, A. (eds.) The 1st International Workshop on Modular Ontologies (WoMO'06). CEUR Workshop Proceedings, vol. 232. CEUR-WS.org (2006), iSWC'06 November 5, Athens, Georgia, USA

[15] Masolo, C., Borgo, S., Gangemi, A., Guarino, N., Oltramari, A.: Ontology library. WonderWeb Deliverable D18 (ver. 1.0, 31-12-2003). (2003), `http://www.loa.istc.cnr.it/old/Papers/D18.pdf` last accessed: 20 June 2017.

[16] Newman, D., Bechhofer, S., Roure, D.D.: myExperiment: An ontology for e-Research. In: Proceedings of the Workshop on Semantic Web Applications in Scientific Discourse (SWASD 2009). CEUR Workshop Proceedings, vol. 523. CEUR-WS.org (2009), washington DC, USA, October 26

[17] Oh, S., Ahn, J.: Ontology module metrics. In: International Conference on e-Business Engineering, (ICEBE'09). pp. 11–18. IEEE Computer Society (2009), macau, China, 21-23 October

[18] Oh, S., Yeom, H.Y., Ahn, J.: Cohesion and coupling metrics for ontology modules. Information Technology and Management 12(2), 81–96 (2011)

[19] Orme, A.M., Yao, H., Etzkorn, L.H.: Coupling metrics for ontology-based systems. IEEE Software 23(2), 102–108 (2006)

[20] Pathak, J., Johnson, T.M., Chute, C.G.: Survey of modular ontology techniques and their applications in the biomedical domain. Integrated Computer-Aided Engineering 16(3), 225–242 (2009)

[21] Schlicht, A., Stuckenschmidt, H.: Towards structural criteria for ontology modularization. In: Proceedings of the 1st International Workshop on Modular Ontologies, (WoMO'06). CEUR Workshop Proceedings, vol. 232. CEUR-WS.org (2006), iSWC'06 November 5, 2006, Athens, Georgia, USA

[22] Schulz, S., Boeker, M.: BioTopLite: An upper level ontology for the life sciences. evolution, design and application. In: Informatik 2013, 43. Jahrestagung der Gesellschaft für Informatik e.V. (GI), Informatik angepasst an Mensch, Organisation und Umwelt. LNI, vol. 220, pp. 1889–1899. GI (2013), 16-20. September 2013, Koblenz

[23] Tartir, S., Arpinar, I.B., Moore, M., Sheth, A.P., Aleman-Meza, B.: OntoQA: Metric-based ontology quality analysis. IEEE Workshop on Knowledge Acquisition from Distributed, Autonomous, Semantically Heterogeneous Data and Knowledge Sources 9 (2005)

[24] Vescovo, C.D.: The modular structure of an ontology: Atomic decomposition towards applications. In: Proceedings of the 24th International Workshop on Description Logics (DL 2011). CEUR Workshop Proceedings, vol. 745. CEUR-WS.org (2011), barcelona, Spain, July 13-16

[25] Vescovo, C.D., Gessler, D., Klinov, P., Parsia, B., Sattler, U., Schneider, T., Winget, A.: Decomposition and Modular Structure of BioPortal Ontologies. In: 10th International Conference on The International Semantic Web Conference (ISWC'10). Lecture Notes in Computer Science, vol. 7031, pp. 130–145. Springer (2011), october 23-27, Bonn, Germany.

[26] Yao, H., Orme, A.M., Etzkorn, L.: Cohesion metrics for ontology design and application. Journal of Computer science 1(1), 107 (2005)