Production Systems and Information Engineering
Volume 11 (3), pp. 100-114
doi: 10.32968/psaie.2023.3.7

ATTRIBUTE ORIENTED ONTOLOGY METRICS

ANITA AGÁRDI
University of Miskolc, Hungary
Institute of Information Technology
agardianita@iit.uni-miskolc.hu

Abstract.

The development of an appropriate ontology model is generally a hard task. One of the main issues is that ontology developers mainly concentrate on classes and neglect the role of attributes. The paper analyzes the role of an appropriate attribute part in providing a high level of interoperability and reusability of the constructed ontology models. In this paper, novel quality metrics are introduces and novel ontologies are evaluated. The following state-of the art metrics are introduced: Base metrics, Class axioms, Object property axioms, Data property axioms, Individual axioms, Annotation axioms, Schema metrics, Knowledgebase metrics, Class metrics, Graph metrics. In addition, own Attribute-oriented metrics are also presented. Through a total of 16 ontology systems are analyzed, which were downloaded from github.

Keywords: ontology, metrics

1. Introduction

The development of an appropriate ontology model is usually a hard task for many reasons. Many efforts have been directed to the creation of methodologies for guiding users in the development of ontologies. There are methodological tool like On-To-Knowledge [1] to help users building ontologies from scratch. One issue during the ontology development is the appropriate selection of ontology components (classes and properties). Considering many ontology examples, we can see that authors usually concentrate on the class level. On the other hand, in other semantic modeling languages, like UML model or concept lattices, the attributes play a key role in the model structure. Our main motivation is to show that only an appropriate set of related set of concepts and attributes can provide a high level of interoperability and reusability of the created ontology models. To help designers to use an appropriate set of attributes, we introduce novel quality metrics on the attribute aspects of ontology modeling.

In this paper, we give first an overview of the related ontology quality metrics. After the survey part we introduce the attribute-based quality metrics.

2. Ontology modeling and quality measures

2.1. Quality metrics

There are many quality requirements on ontology models that should be considered during the ontology construction process. In the literature, we can find some works on the analysis of the requirements [2-3] but due to the complexity of ontology management, there is not a general and widely accepted theoretical and technical foundation of the synthesis of the requirements. On the other hand, from the viewpoint of practical applications, a better support for the quality of ontology development is a key factor in the desired success of the ontology model. Thus the development of efficient tools for supporting ontology modeling is still an actual and relevant research topic in the knowledge engineering community.

García J., García-Peñalvo F.J., Therón R. [7] analyzed several ontological metric frameworks in the survey article. The following frameworks were analyzed: Vrandecic, Alani, Orme, Yinglong, OntoClean, Ontometric, Protégé, OntoQA, Ontology Metrics, Yang. Metrics were analyzed based on Semantic / Structure, Ranking, Cohesion, Coupling. There was not any metric that performed equally well in all analyzes.

Vrandečić, D., & Sure, Y. [8] report how better ontological metrics could be created. Measuring and comparing ontologies is very important, but this can only be done well if we know what we want to compare them from. After presenting the state-of-art literature, the article also cites examples from the literature.

Yao, H., Orme, A. M., & Etzkorn, L. [9] present and analyze ontology cohesion metrics. Number of Root Class, Number of Leaf Classes, Average Depth of Inheritance Tree of Leaf Nodes are presented as ontology metrics.

Tartir, S., Arpinar, I. B., Moore, M., Sheth, A. P., & Aleman-Meza, B. [10] also compared ontologies based on the values of the metrics. The following metrics were used for the analyzes: Relationship Richness, Attribute Richness, Inheritance Richness, Class Richness, Average Population, Cohesion, Importance, Fullness, Inheritance Richness, Relationship Richness, Connectivity, Readability. Three sample ontologies were analyzed, which had completely different numbers of classes (44-3,299-352) and instances (813,217-70,850-2,034).

Considering the difficulties in practical ontology modeling, we can emphasize the following factors:

- Many developers are coming from the database domain, where a closed world approach is the dominating model. In contrast, the ontology model uses an open world approach is the dominating model.
- The ontology should cover a wide range of concepts and global ontology is usually constructed from many partial (domain) ontology models having different granurality and functionality. Due to the large size of ontology models, an automatic integration tool is required that can discover the hidden inconsistencies.
- Subjectivity. There are no golden rules and guidelines for ontology design. There exist different approaches in ontology development like inductive or deductive approaches resulting in very different ontology models for the same domain.
- In many OOP models the main relationship between the classes is the specialization relationship. The child classes inherit the attributes of the parent class automatically. In an ontology model, the declaration of a domain axiom on a property does not mean automatic inheritance, i.e. a separate subclass axiom must be added to the corresponding ontology.

Considering the different approaches in the literature, we can categorize the ontology-specific quality aspects into the following three main areas [2] [4]:

- schema and type definitions,
- the amount and the resolution of the data,
- clarity, compatibility and usability.

2.2. Structural Measures

There are several structural measures, the following measures contains the ONTOMETRICS [5-6] system:

- Base metrics: the number of axioms, logical axioms, classes, object properties, data properties and individuals.
- Class axioms: number of subclasses, equivalent classes and disjoint classes.
- Object property axioms: the number of SubObjectPropertyOf axioms, equivalent object property axioms, inverse object properties axioms, disjoint object properties axioms, functional object properties axioms, inverse functional object properties axioms, transitive object property axioms, symmetric object property axioms, asymmetric object property axioms, reflexive object property axioms, irreflexive object property axioms, object property domain axioms, object property range axioms, SubPropertyChainOf axioms.
- Data property axioms: the number of SubDataPropertyOf axioms, equivalent data properties axioms, disjoint data properties axioms, functional data property axioms, data property domain axioms, data property range axioms
- Individual axioms: the number of class assertion axioms, object property assertion axioms, negative object property assertion axioms, data property assertion axioms, negative data property assertion axioms, same individuals axioms, different individuals axioms.
- Annotation axioms: the number of annotation axioms, annotation assertion axioms, annotation property domain axioms, annotation property range axioms.
- Schema metrics: Schema metrics deal with ontology design. The following metrics are distinguished:
    - Attribute richnesses: The number of attributes defined for classes indicates the quality of the ontology design and the amount of information about the individuals. The higher this number, the more knowledge the ontology conveys.
    - Attribute richness (AR) is the average number of attributes per class.
    - Inheritance Richness: Specifies the distribution of information at different levels of the ontology inheritance tree. It specifies how well the knowledge is grouped in the ontology into different categories and subcategories. There are two types, the horizontal and the vertical ontology. Horizontal ontology means that classes contain a large number of direct subclasses. This indicates that the ontology provides a wide range of general knowledge, with low detail Vertical ontology means that classes contain a small number of direct subclasses, which means that the ontology covers a particular area in detail.
- Inheritance richness is given by the average number of subclasses per class.
    - Relationship Richness: Specifies the diversity of relationship types. If an ontology contains only inheritance relationships, it usually conveys less information than an ontology that contains a wide variety of relationship groups. Relationship richness is the percentage of (non-hereditary) relationships between classes.
    - Attribute-Class Ratio: Specifies the ratio of the classes that contain the attributes to all the classes.
    - Equivalence Ratio: specifies the ratio between equal classes and all classes of the ontology.
    - Axiom Class Ratio: Specifies the ratio of axioms to classes.

  - Inverse Relations Ratio: The ratio of inverse relations to total relations.
  - Class Relation Ratio: Specifies the ratio of classes to attributes.
- Knowledgebase metrics: Metrics that describe the entire knowledge base and metrics that specify how each class is used in the knowledge base. The following metrics are known:
  - Average Population: Indicates the number of instances relative to the number of classes. The average number of individuals (AP) is defined by dividing the number of individuals (I) by the number of classes (C).
  - Class Richness: Indicates whether each class has an individual. Class richness (CR) is determined by the quotient of non-empty classes (classes with instances) (C ') and the total number of classes (C) defined in the ontology schema.
- Class metrics:
  - Class connectivity: what classes are connected by instances in the ontology. The connectivity of a class $(\mathrm{Conn}\,(C_i))$ is the number of connections of instances of a class to instances of other classes (NIREL).

    $$\mathrm{Conn}(C_i) = |\mathrm{NIREL}(C_i)|$$
  - Class Fullness: Class C_i completeness (F) is defined as the quotient of the actual number of instances belonging to the root alpha C_i $(C_i(I))$and the expected number of instances belonging to the root subtree C_i $(C_i'(I))$.
  - Class Importance: The percentage of instances in the inheritance subtree classes from the current class relative to the total number of instances.
  - Class Inheritance Richness: The inheritance richness (IRc) of class C_i is the average number of subclasses per class in the subtree.
  - Class Readability: Indicates that there are human-readable descriptions in the ontology, such as comments, labels, or captions.
  - Class Relationship Richness: How much of the class relationships are actually used by the instances. The relationship richness (RR) of class C_i with the percentage of relationships used by instances I_i and belonging to $C_i(P(I_i, I_j))$ compared to the number of relationships belonging to class C_i with $P(C_i, C_j))$.
  - Class children: number of direct child classes.
  - Class instances: Represents the number of instances of a given class.
  - Class properties: Indicates the number of properties in the class.
- Graph metrics: Shows the structure of ontologies.
  - Absolute root cardinality: shows the number of root nodes in the graph:
  - Absolute leaf cardinality: indicates the number of leaf nodes in the graph:
  - Absolute sibling cardinality: gives the number of sibling nodes.
  - Absolute depth
  - Average depth
  - Maximum depth
  - Absolute breadth
  - Average breadth
  - Maximum breadth
  - Ratio of leaf fan-outness: the result of leaf cardinality and graph cardinality
  - Ratio of sibling fan-outness: the quotient of absolute sibling cardinality and graph cardinality
  - Tangledness: This means that this node has multiple inbound edges.
  - Total number of paths: the sum of the different paths in the graph.
  - Average number of paths: the quotient of the number of each path and the number of graphs

3. Attribute-oriented metrics

In this section specific, attribute-based quality metrics are introduced. The main principle is that the attribute distribution in the taxonomy must be well-balanced and consistent. To measure this quality, we propose the following measures, where A means the attributes, C means the concepts [11]:

- $m_A = \frac{|A|}{|C|}$: the relative number of the attributes. If the value is low (near or below 1), there are too few attributes. If the number is too high, most of the attributes are not used in the taxonomy construction
- $m_l = \frac{|C_\emptyset|}{|C|}$: the ratio of concepts with empty local (not inherited) attribute set. If this value is greater than 0, then the ontology is invalid.
- $m_= = \frac{|C_=|}{|C_A|}$: the ratio of concepts having not unique attribute set, whereC_A means the concepts having attributes. If this value is greater than 0, then the ontology is invalid.
- $m_l = \frac{\sum_{c \in C_\emptyset}(|A(c)|)}{|C|}$: This measure shows the average length of the local (not empty) attribute sets. A high value means that many attributes are not relevant in the taxonomy construction.

4. Ontology schemas and evaluations

4.1. The ontology schemas

A total of 16 ontological sample models were evaluated. I downloaded the sample models from GitHub. The sample models cover a variety of topics. I will first present the topics of the sample models and then the values of the metrics of the models.

Table 1. Ontologies

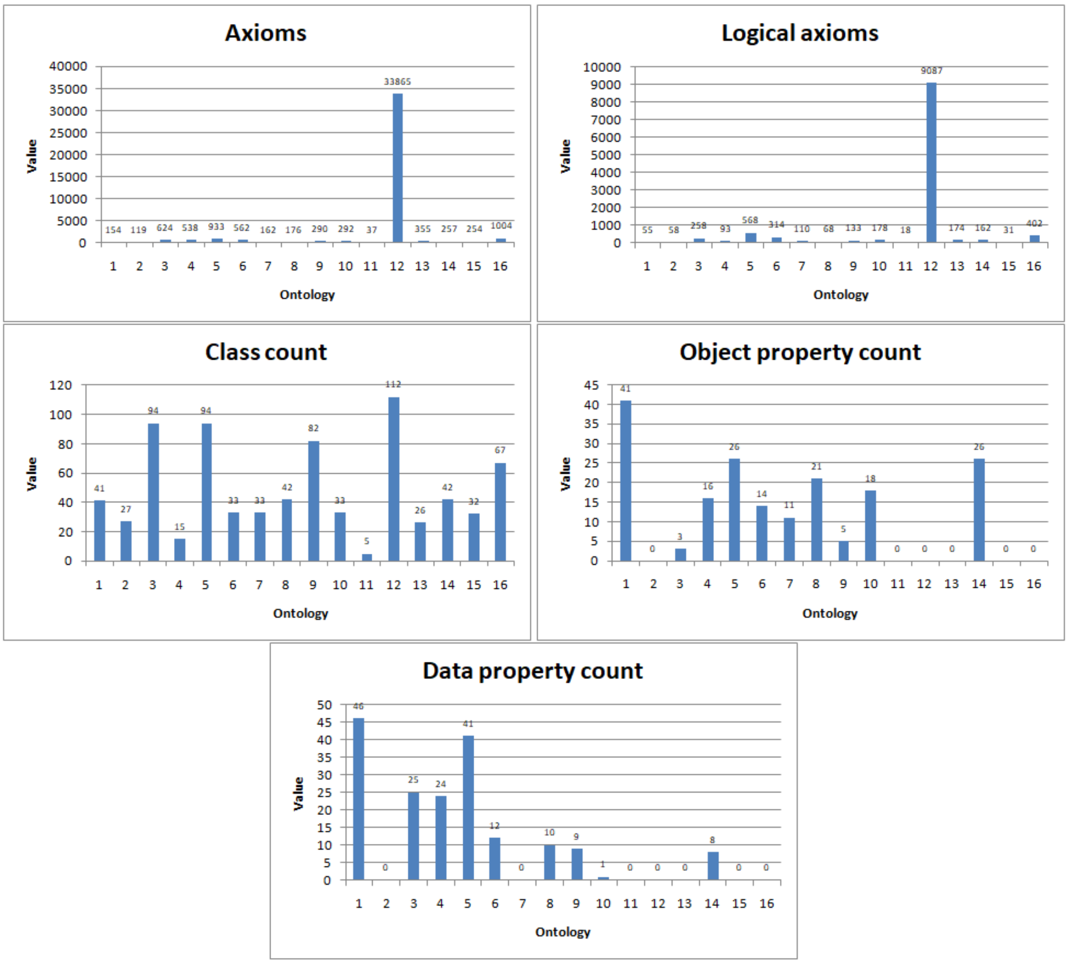
1	College Mngt Sys.owl: https://github.com/ayesha-banu79/Owl-Ontology
2	Companies.owl: https://github.com/detnavillus/rdf-owl-ontologies
3	DhiQar.owl: https://github.com/Epistematica/dhiqar-ontology
4	doacc.owl: https://github.com/DOACC/doacc
5	fertilizer.owl: https://github.com/nidhi-malik/agriculture
6	filfestival.owl: https://github.com/thodoris/FilmFestival-OWL-Ontology
7	funny-salad-ontology.owl: https://github.com/basselkassem/funny-salad-ontology.git
8	hls_2015-12-6.owl: https://github.com/nabito/hls
9	Laser-Thermal-Mircostructure.owl: https://github.com/iassouroko/AMontology
10	Library Ontology.owl: https://github.com/ayesha-banu79/Owl-Ontology
11	Literature.owl: https://github.com/detnavillus/rdf-owl-ontologies
12	Music.owl: https://github.com/detnavillus/rdf-owl-ontologies
13	Politics.owl: https://github.com/detnavillus/rdf-owl-ontologies
14	SafetyOntology.owl: https://github.com/mahsa-teimourikia/Safety-Ontology
15	sem.owl: https://github.com/lindenb/semontology
16	SoftwareTechnology.owl: https://github.com/detnavillus/rdf-owl-ontologies

- College Mngt Sys: presents an ontology of a college. The ontology includes classes such as College, Course, Event, Library, Person, including Employee, Student. It also includes a project and a publication.

- Companies: illustrates certain types of companies. For example, Energy, Financial_Services, Food, Foreign_Corporations, Health_Care, Hospitality, Manufacturing.
- DhiQar: used to describe records such as boards, cylinders, and envelopes from the Dhi Qar region (Iraq, Nassiriya region). The ontology also describes the characteristics of the find, such as shape, material, size, and other visual characteristics, and so on.
- Doacc: An ontology for describing cryptocurrency. It contains classes such as Blockchain protocol, Collection, Concept, Cryptocurrency, Algorithm (dbpedia: Algorithm), Repository, Version.
- Fertilizer: Fertilizer ontology. It contains classes such as Processing Method (ApplicationMethod), Processing Time (ApplicationTime), Manure Types (Fertilizer). Within fertilizer types (Fertilizer), organic fertilizer (Biofertilizer), chemical object (ChemicalFertilizer), etc. Other classes are Nutrient, Season, Solubility, State, Unit, Year, Zone.
- Film Festival: An ontology models a film festival. There is a film festival (dbo: FilmFestival) class, Event, person (foaf: Person and Person) classes. The Person can be an Actor, a Director. There is also a Place class and Movie (schema: Movie), Restaurant (schema: Restaurant) classes.
- Funny salad ontology: this ontology describes salad types. There is a country class, the food class is derived from lettuce (salad), salad ingredients (saladIngredient), salad topping (saladTopping), and spices (spices).
- Hls_2015-12-6: Human Localization Sensor Ontology (HLS), human location ontology. It contains classes such as Location, InformationObject, Input, Process, including Sensing, which can be LocationMethod or UserInterfaceSensing. In addition, there is a natural person (NaturalPerson), an object (Object), a sensor (Sensor), and so on.
- Laser-Thermal-Mircostructure: this ontology for the development of the additive manufacturing concept, part of NIST's Systems Integration for Additive Manufacturing project.
- Library Ontology: this ontology models a library. The ontology includes library users (LibraryMember) including admin (AdminStaff), faculty (Faculty), guest (GuestUser), student (Student). It also includes library staff (LibraryPersonnel), library resources (LibraryResource) such as Book, CD, Journal, NewsPaper, OnlineJournal, and Thesis. It also includes library services (LibraryService).
- Literature: a small ontology describing literary books. It contains only a few classes. There is an Author, Book class. The subclasses of the Fiction class are the Murder_Mystery and the Science-Fiction novel.
- Music : An ontology describing music. It includes classes such as Compositions, Festival, Genre, Instrument, Instrument Manufacturers, Music School, Music Writer, Musician. , music producer (Record Producer), recording (Recording), etc.
- Politics: An ontology that models politics. It includes the following main classes: Political_Party, Politician, Supreme_Court_Decision. The Politician includes the following departments: Attorney_General, Congressman, Governor, Prime_Minister, Secretary_Of_Defense, Secretary_Of_Homeland_Security, etc.
- SafetyOntology: Occupational safety ontology, which includes the following main classes: Consequence, Entity, Environment, Object, Subject, WorkActivity, HazardousEvent , IoT service (IoTService), device

monitoring (MonitoringDevices), prevention strategy (PreventiveStrategy), role (Role), etc.

- Sem: Software Evaluation Metrics Ontology (SEMO). An ontology that gives users a clear idea of what the software and its documentation are like. For example, your own code (I_have_written_it). The comment is automatically general (autogenerated), has documentation (documentation_exists). The documentation is embedded - comments- (documentation_is_embbeded) or the code itself is the documentation (the_code_is_the_documentation). Documentation may be missing (documentation_is_missing). I don't remember why I wrote this snippet (i_do_not_remember_why_I_wrote_it), I don't understand how the code can work (i_do_not_understand_how_it_could_work) (it_compiles), obsolete code (it_is_deprecated), please do not view the code (please_do_not_look_at_the_code), etc. also includes classes.
- SoftwareTechnology: ontology related to software technology. It includes classes such as Software Company (Software_Company), Software Language (Software_Language), including two classes, Data Description Language (Data_Description_Language), and Programming Language (Programming_Language). In addition, the ontology also includes software license (Software_License), platform (Software_Platforms), software product (Software_Product), protocol (Software_Protocol), system monitoring (System_Monitoring).



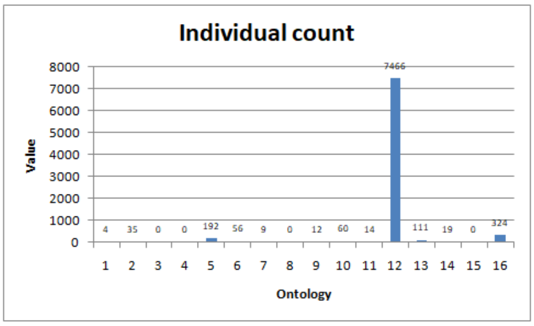


Figure 1.: The base metrics

The number of axioms is between 37 and 33865, but most ontologies contain only a few hundred axioms. The number of logical axioms is between 18 and 9087, but even here most ontologies contain only a few hundred logical axioms. The number of classes is between 5 and 112. Most ontologies contain less than 50 classes. The number of object properties is between 0 and 41 for each ontology. It is 0 for six ontologies, around 10-20 for the other ontologies, and 41 for a single ontology. The number of data properties is between 0 and 46. More than 40 for two ontologies and 0 for seven ontologies. The number of individuals is between 0 and 7466 for each ontology. Four ontologies have a value of 0 and some ontologies have a value close to 0. It is outstandingly very high for a single ontology (7466), but the value of 324 can also be said to be high.

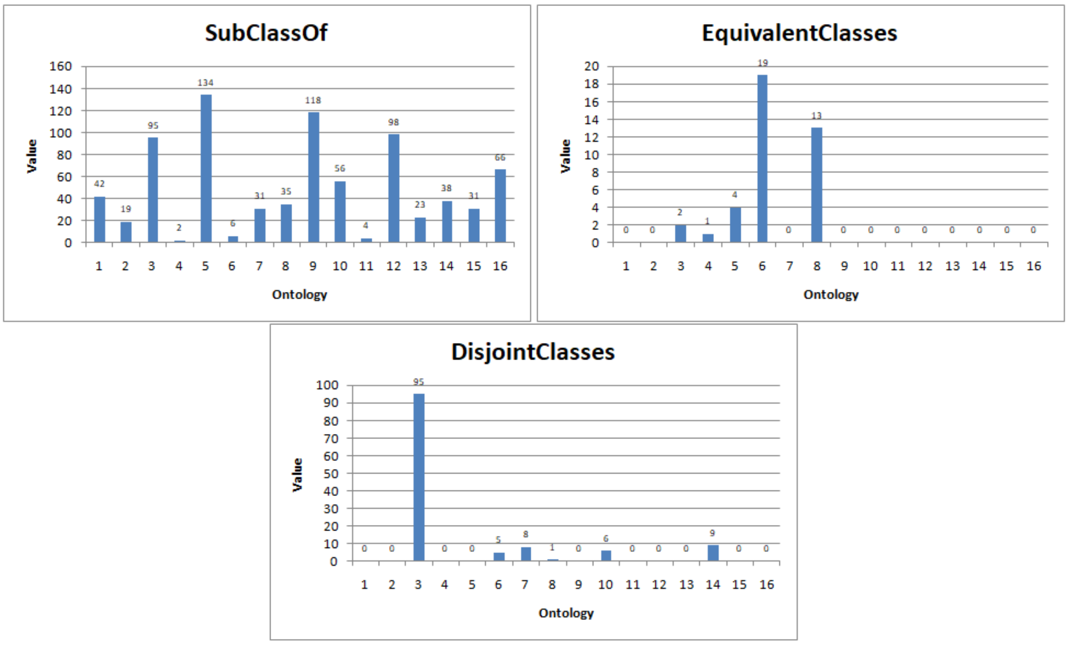


Figure 2: The class axioms

The number of subclasses is between 2 and 134. Most ontologies are below 50. The number of equal classes for most ontologies is 0. Their number is between 0 and 19 for each ontology examined. The number of different classes in a single ontology was very high, 95.

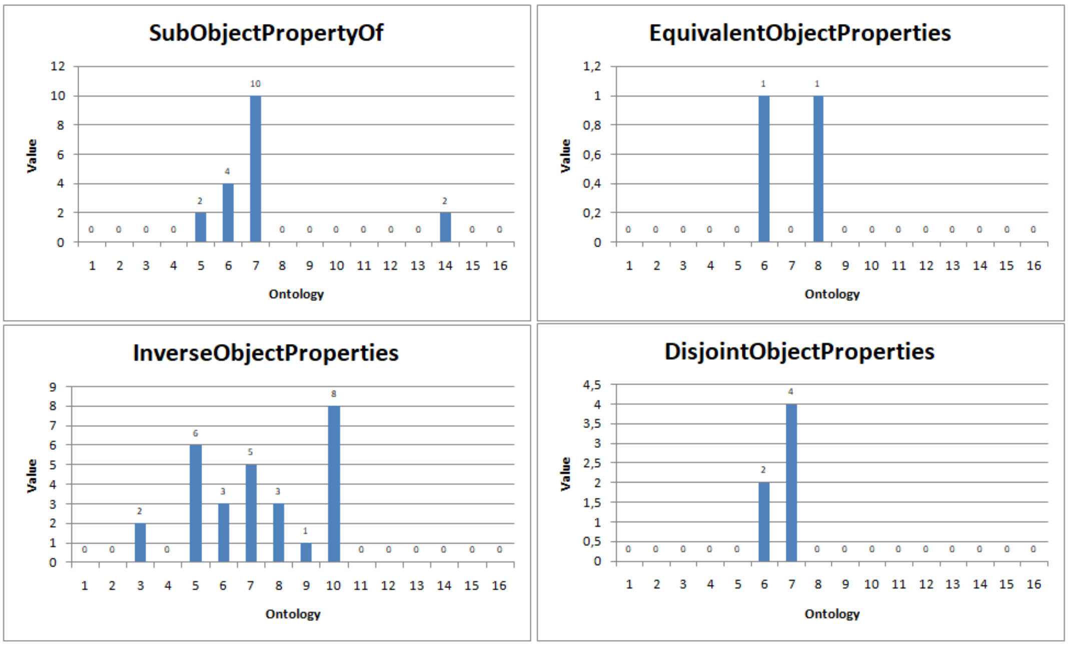
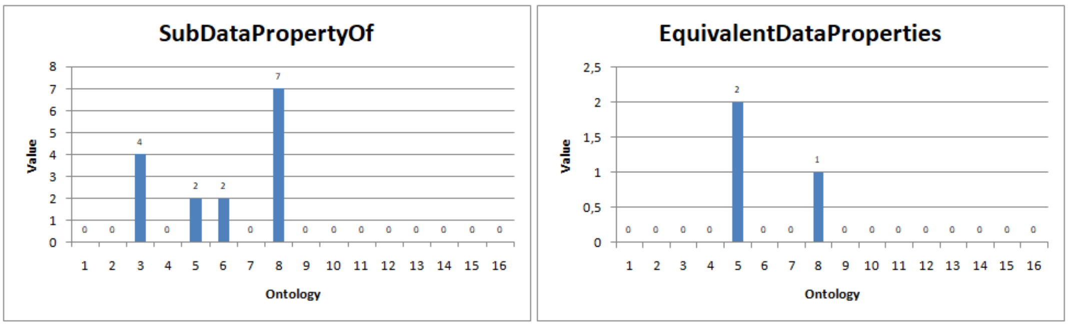


Figure 3: The object property axioms

The number of derived object properties for each ontology ranged from 0 to 10. It was 0 on most ontologies. The number of equal object properties was 0 or 1 for ontologies. The number of inverse object properties ranged from 0 to 8. For most ontologies, this was 0. The number of different object properties ranged from 0 to 4, and for most ontologies this value was 0. The metric values of the functional object properties were also low for the ontologies, this value was 0 in most cases and the maximum value was 5. The number of inverse functional object properties was 0 in almost all ontologies, 1 in one case and 2 in one case. Transitive object property values range from 0 to 1, with almost all ontologies having a value of 1. The symmetric object property metric was 0 or 1 for each ontology. The number of asymmetric object properties for each ontology was 0. The number of reflexive object properties for each ontology was 0. The number of irreflexive object properties for each ontology was 0. Object property domain values ranged from 0 to 25 for ontologies. Most ontologies had a number of 0. Object property range values range from 0 to 25 for ontologies. For most ontologies, this metric is 0 or close to 0. The value of the derived property chain metric except for a single ontology was 0.



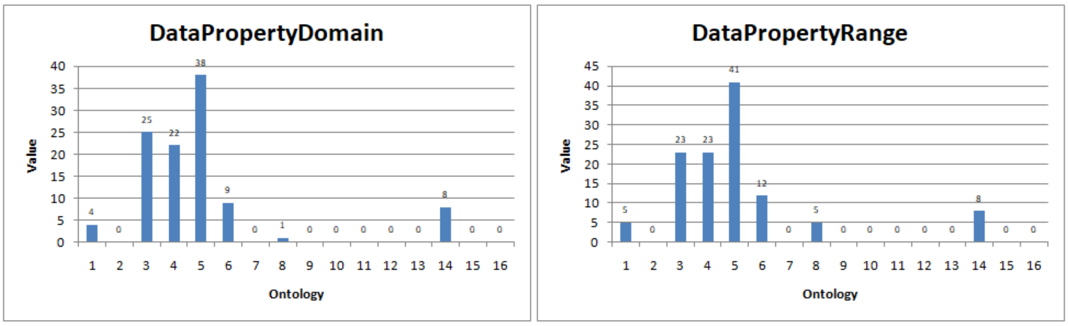


Figure 4: The data property axioms

The sub data property was 0 for each ontology in most cases. The highest value was 7, achieved by a single ontology system. The metric values of the equal data property were 0 for almost all ontologies, except for two, one for 1, and 2 for the other. The different data property values for each ontology were 0. The functional data property value was 0 for almost all otologies, 1 except for one and 7 for the other. The values of the data property domain metrics were 0 for most ontologies, over 20 for some ontologies, and 38 for the highest. The data property range values were also 0 for most ontologies, over 20 for some ontology models, and 41 for the highest.

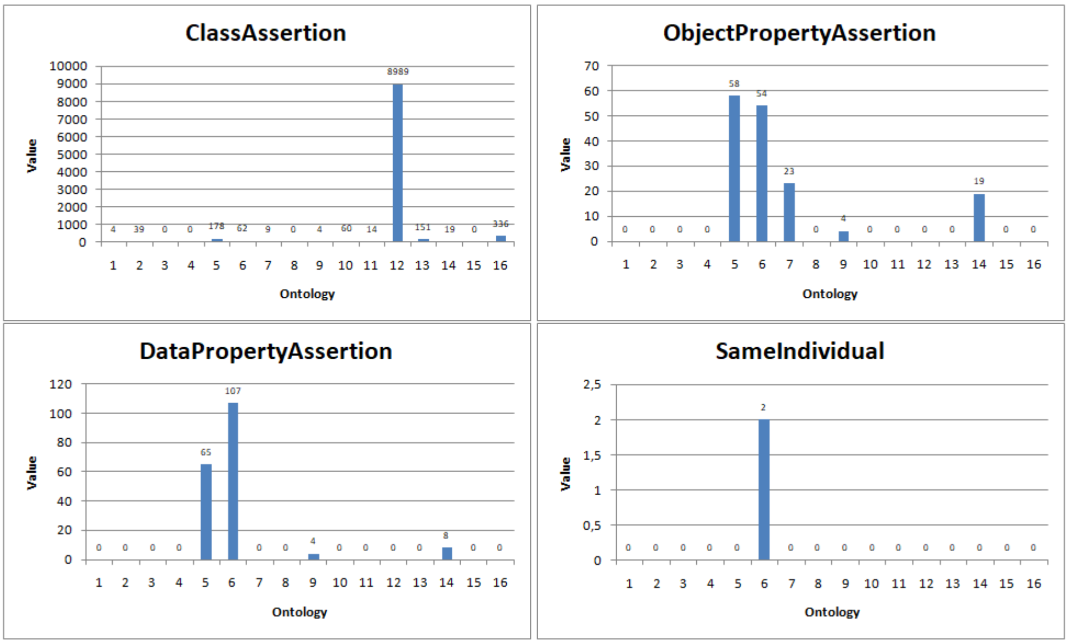


Figure 5: The individual axioms

The class assertion metric is low for most classes between 0 and 100. Outstanding for a single class, 8989. The values of the object property statement were 0 in most cases, more than 50 in two cases, and 58 was the highest. The data property statement values were 0 in most cases, very high in one case 65 , and another case 107. The NegativeObjectPropertyAssertion was 0 in all cases. NegativeDataPropertyAssertion was 0 in all cases. The number of identical individuals was 0 except for a single ontology. The number of different individuals for each ontology model was 0.

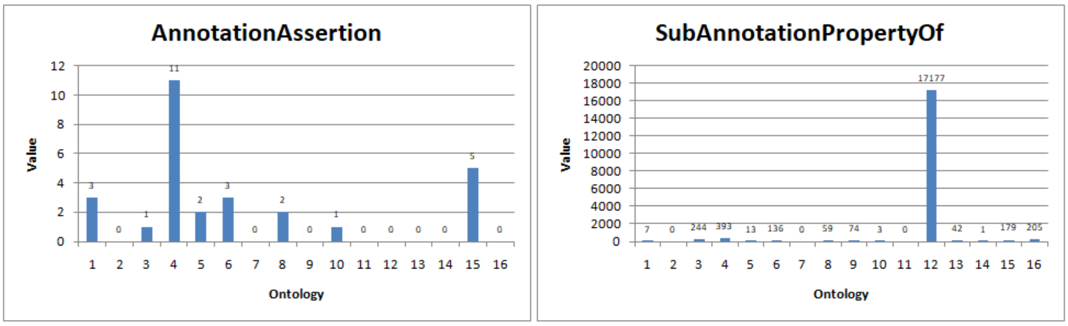


Figure 6: The annotation axioms

The number of annotation statements was low for all ontologies, with a value of 0 in most cases and a maximum of 11. Derived annotation property values were no longer low in many cases, reaching hundreds of values. In one case, it was over 17,000. Values ranged from 0 to 17177. Annotation property domain values for all ontologies were 0. The annotation property range values were also 0 for all ontologies.

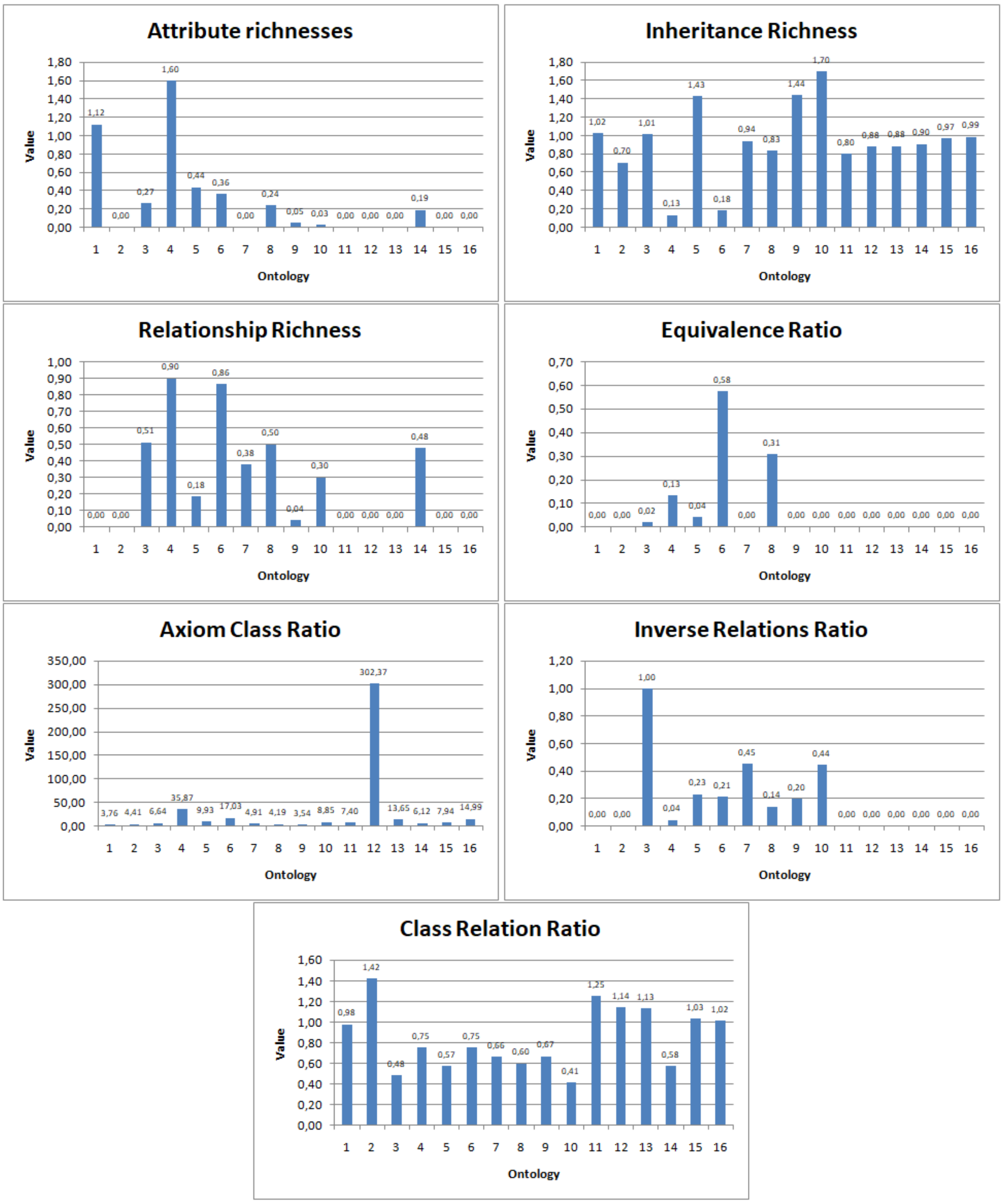


Figure 7: The schema axioms

Attribute richness values range from 0 to 1.6. In most cases, the value of this metric for ontologies was 0. Inheritance richness values ranged from 0.13 to 1.7. The value of the relationship richness metric was 0 for some ontologies, and the highest value was 0.9. The attribute-class ratio metric values for all ontologies were 0. The equivalence ratio for most ontologies was 0, with the highest value being 0.58. The proportion of the axiom class was low in almost all cases, the highest at 302.37. The inverse relationship ratio for most ontologies of the metric was 0, with the highest value being 1. The lowest value of the class relationship ratio was 0.41 and the highest value was 1.47.

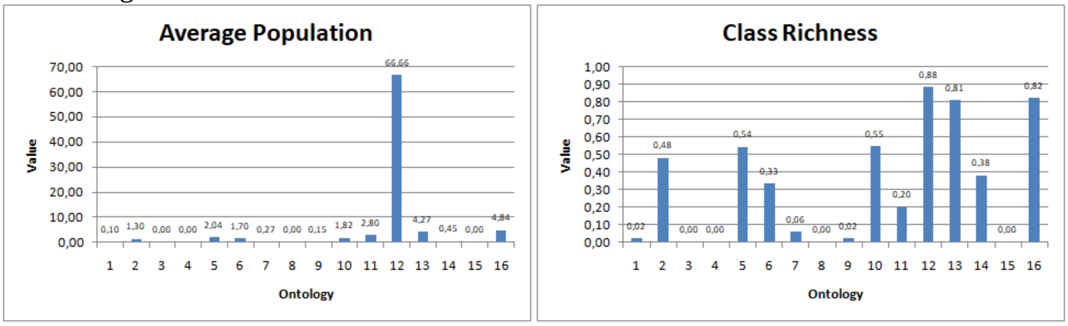
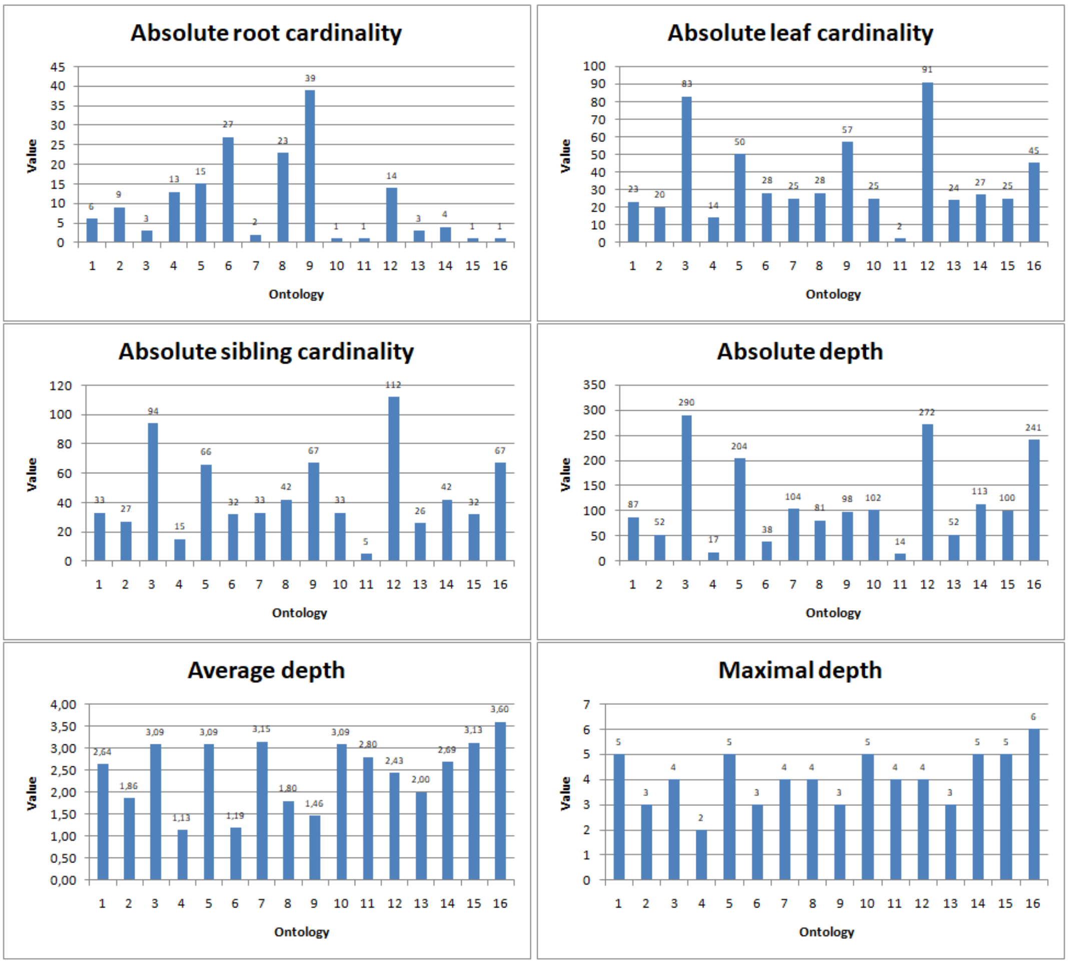


Figure 8: The knowledgebase metrics

The average number of individuals was close to 0 or 0 in many cases, with the highest value being 66.66. Class richness values range from 0 to 0.88 for each ontology.



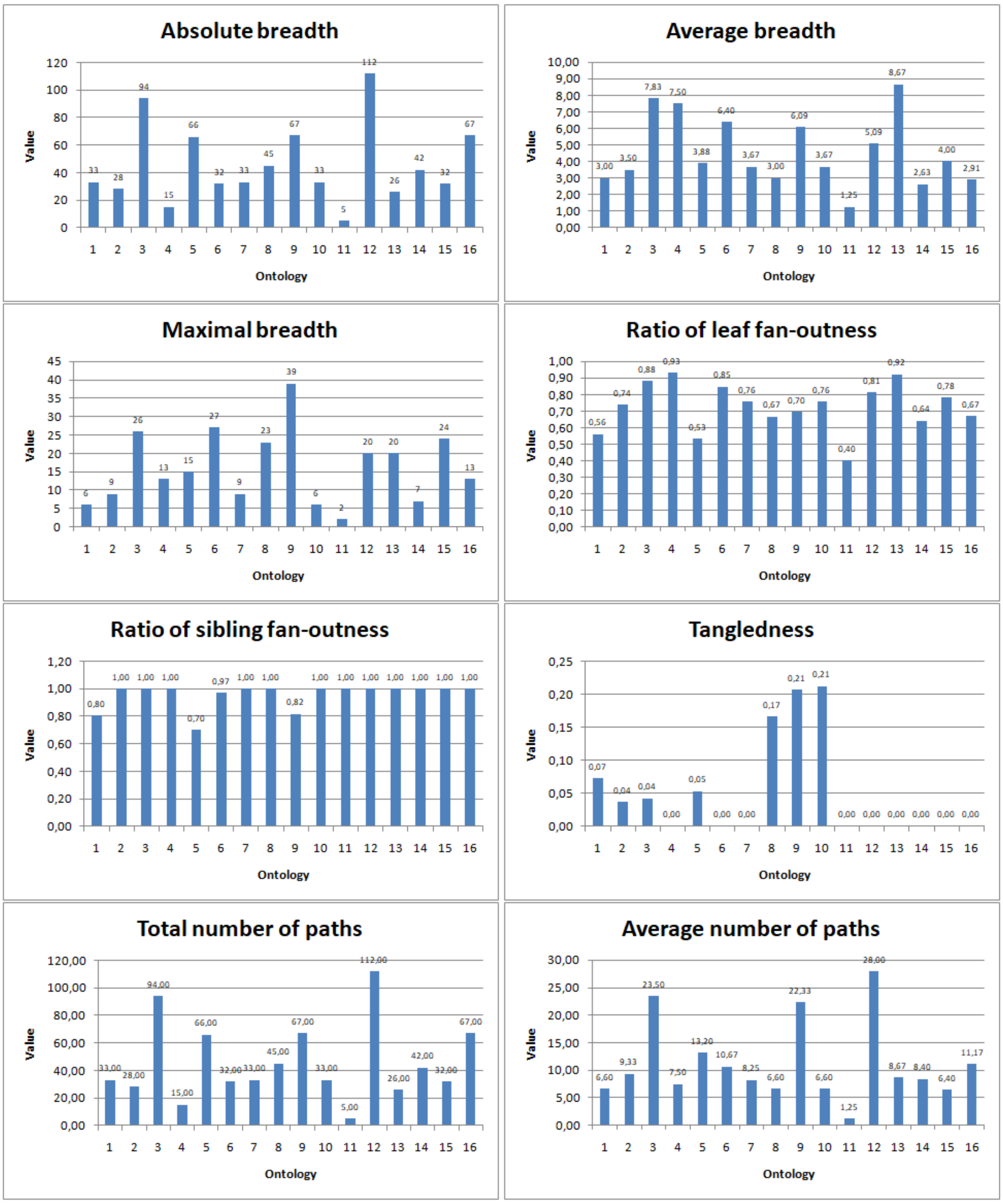


Figure 9: The graph metrics

Absolute root cardinality values were in many cases less than 10, with the lowest value being 1 and the highest being 39. Absolute leaf cardinality values ranged from 2 to 91. Most of the values were between 20 and 30. Absolute sibling cardinality values ranged from 5 to 112. Absolute depth metrics range from 14 to 290. Average depths range from 1.13 to 3.6. Maximum depths are low, ranging from 2 to 6. Absolute breadth values range from 5 to 112. The average breadth values are between 1.25 and 8.67. Maximum breadth values are between 2 and 39. Ratio of leaf fan-outness values range from 0.4 to 0.9. Ratio of sibling fan-outness values range from 0.7 to 1, with 1 on most ontologies. Tangledness values for most ontologies were 0, with a highest value of 0.21. The values for the number of paths metric are different for each ontology. The lowest value was 5 and the highest was 112. The average number of trips also varied, with the lowest being 1.25 and the highest being 28.

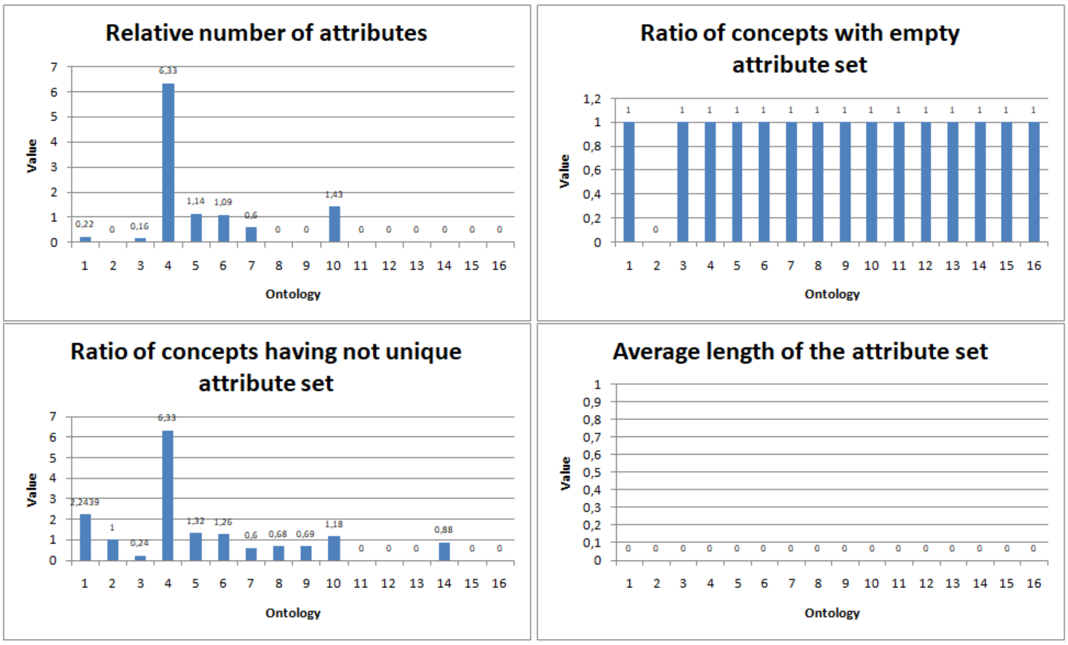


Figure 10: The attribute-oriented metrics

The relative number of attributes ranged from 0 to 6.33. If this number is too large, most attributes are not used. This is true for ontology 4. The proportion of concepts that had an empty set of attributes was almost always 1. In one case, it was 0. This means that if this value is not 0, then the ontology is not valid. The ratio of concepts that do not have a unique set of attributes, in quite a few cases, the value is not 0, which indicates that the ontology is invalid. Especially for ontology 4, where it had a value of 6.33. The average length of the set of attributes was 0 in each case, which indicates that the attributes are relevant (would not be relevant for a large value).

5. Conclusion

The paper analyzes the role of attributes in providing a high level of interoperability and reusability of the constructed ontology models. To help designers to use an appropriate set of attributes in ontology modeling, the paper introduces novel quality metrics on attribute aspects of the ontology. The following state-of the art metrics are introduced: Base metrics, Class axioms, Object property axioms, Data property axioms, Individual axioms, Annotation axioms, Shema metrics, Knowledgebase metrics, Class metrics, Graph metrics. In addition, own Attribute-oriented metrics are also presented. The metrics were evaluated in detail, through a total of 16 ontology systems that we downloaded from github. During the evaluation of the sample systems, we noticed that most of the systems do not describe the topic in detail, they need further expansion. However, some systems contain a remarkably large number of some basic elements (e.g., individuals, classes). Also during the evaluation of attribute-oriented metrics, we noticed that some ontologies are not valid.

Acknowledgement. „Supported by the ÚNKP-20-3 New National Excellence Program of the Ministry for Innovation and Technology from the source of the National Research, Development and Innovation Fund."

References

[1] D. Fensel, Ontology-based knowledge management. Computer, 35(11), (2002), 56-59. https://doi.org/10.1109/MC.2002.1046975

[2] N. Sugiura, Y. Shigeta, N. Fukuta, N. Izumi, & T. Yamaguchi, Towards on-the-fly ontology construction–focusing on ontology quality improvement. In European Semantic Web Symposium (2004), 1-15 https://doi.org/10.1007/978-3-540-25956-5_1

[3] B. Stvilia, A model for ontology quality evaluation. First Monday, (2007). https://doi.org/10.5210/fm.v12i12.2043

[4] S. Tartir, I. B. Arpinar & A. P. Sheth, Ontological evaluation and validation. In Theory and applications of ontology: Computer applications (2010), 115-130 https://doi.org/10.1007/978-90-481-8847-5_5

[5] A. Lozano-Tello & A. Gómez-Pérez, Ontometric: A method to choose the appropriate ontology. Journal of Database Management (JDM), 15(2), (2004), 1-18. https://doi.org/10.4018/jdm.2004040101

[6] ONTOMETRICS: https://ontometrics.informatik.uni-rostock.de/ontologymetrics/

[7] J. García, F. Jose'García-Peñalvo & R. Therón, A survey on ontology metrics. In World Summit on Knowledge Society, (2010), 22-27 https://doi.org/10.1007/978-3-642-16318-0_4

[8] D. Vrandečić & Y. Sure, How to design better ontology metrics. In European Semantic Web Conference, (2007), 311-325 https://doi.org/10.1007/978-3-540-72667-8_23

[9] H. Yao, A. M. Orme, & L. Etzkorn, Cohesion metrics for ontology design and application. Journal of Computer science, 1(1), (2005), 107-113.

[10] S. Tartir, I. B. Arpinar, M. Moore, A. P. Sheth & B. Aleman-Meza, OntoQA: Metric-based ontology quality analysis, (2005).

[11] Agárdi A, Kovács L. Property-Based Quality Measures in Ontology Modeling. Applied Sciences. 2022; 12(23):12475. https://doi.org/10.3390/app122312475