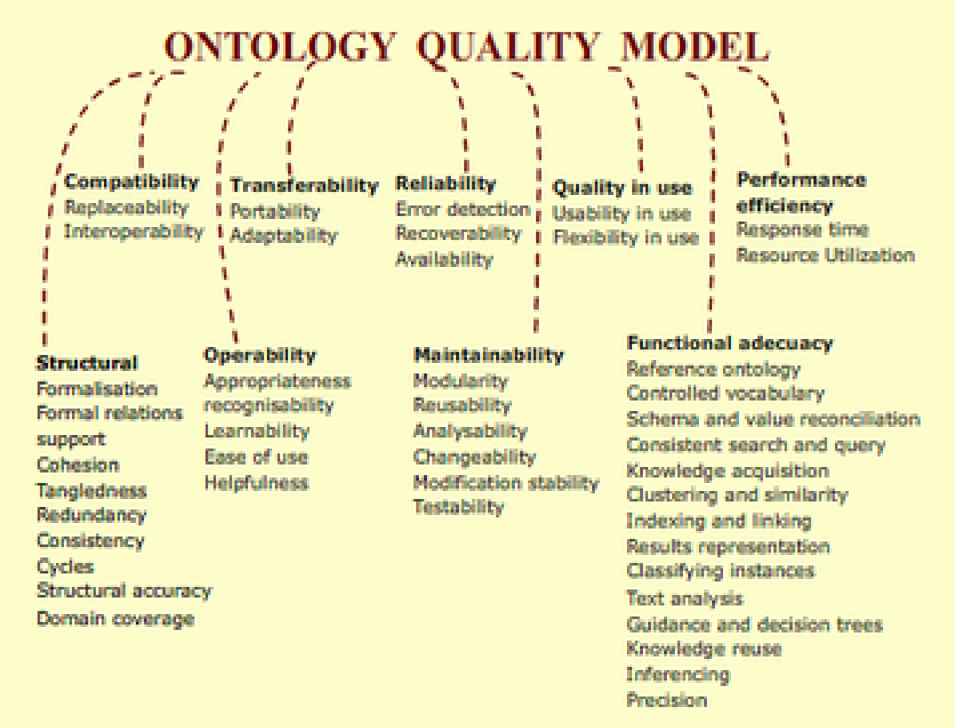
The Ontology Quality Evaluation Framework (OQuaRE)

OQUARE[1] is a framework for evaluating the quality of ontologies based on the standard ISO/IEC 25000:2005 for Software product Quality Requirements and Evaluation known as SQuaRE [2]. OQuaRE aims to define all the elements required for ontology evaluation: evaluation support, evaluation process and metrics. The current version of our framework includes, so far, the quality model and the quality metrics, because they are the basic modules for the technical evaluation of the quality of the ontology, and which are explained next.

The OQuaRE Quality Model

This model reuses and adapts the following SQuaRE characteristics to ontologies: functional adequacy, reliability, operability, maintainability, compatibility, transferability, performance efficiency and quality in use. Moreover, according to requirements, principles and characteristics of ontologies and to the state of the art of ontology evaluation [3], structural features of ontologies are important to evaluate their quality, but they are not considered in the standard. Consequently, we added such characteristic to our framework. In order to determine the quality sub-characteristics, we combined the ones suggested in SQuaRE with the ones suggested by state-of-the-art methods from the ontology evaluation community. the functional adequacy sub-characteristics are the intended uses for ontologies identified in [4],

MB Additions, [comments[



STRUCTURAL

This category is the only one in this framework that is not specified as such in SQuaRE, but it is important when evaluating ontologies, since it accounts for ontology quality factors such as consistency, formalisation, redundancy or tangledness

Sub-characteristic	Definition	Description	Metrics	Conc/ Op
Formalisation	Capability of the ontology to support reasoning in principle Use of upper level partitions with disjoints to ensure semantics of classes are respected.	An ontology should be expressed in a common formal language E.j. for bio-ontologies in agree with the ontology principle FP 002 format [5]: OBO Format,	P23. Using incorrectly ontology elements	C O

		OWL or OWL2 concrete syntax, RDF/XML, OWL2-XML, OWL2-Manchester Syntax, Common Logic concrete syntax, CLIF, Conceptual Graphs, etc		
Formalisation	Capability of the ontology to support reasoning in an application, in practice Making sure the ontology meets computational criteria.	Application of DL-safe rules (if in a DL language OWL or OWL2) and other mathematical measures to ensure the suitability of the ontology for a reasoning based application.. Other tests in real life e.g. run tests on sample data, can establish computability without necessarily being able to prove computability. Level of expressivity v formalization is a balance.	(measures exist) http://oa.upm.es/5453/1/OntologyTest.pdf	O
Formal relations support	Capability of the ontology to represent relations supported for formal theories different to the formal support for taxonomy	It accounts for the types of formal relations supported by the ontology, different to "is a" relations This covers how much information is conveyed by the ontology.	RROnto	C O
Structural accuracy	Degree of the correctness of the terms used in the ontology Requires human review. Domain experts. Be able to indicate whether this is an issue	correctness of taxonomic links, use of upper level with disjoint categories, consideration of "all some rule" in case of existential restrictions, value restrictions only if disjoint partitions available, domain and range restrictions of object properties, sufficient metadata and annotation properties, free text definitions where necessary	Experts, P5, P14, P15, P16, P17.P19, P23. P25	C

Consistency	Degree of the consistency of the ontology	consistent naming conventions, Logical consistency, structural consistency, consistent distinction class - instance E.g. Capitals, plurals, Standards, etc	P22	C O
Tangledness	This measures the distribution of multiple parent categories, so that it is related to the existence of multiple inheritance.	[Note this is a good thing in a conceptual ontology and a bad thing in an application ontology]	TMOnto	
Cycles	The existence of cycles through a particular semantic relation is usually a sign of bad design, since they may lead to inconsistencies.		P6	C O
Cohesion	An ontology has a high cohesion if the classes are strongly related.	Density of relationships among classes. Disconnected parts of the ontology (islands)	LCOMOnto P4 Coh-QA	C
Domain coverage	The degree to which The ontology cover the specified domain	It is evaluated by experts. Also cross reference to domain technical standards for message and data, to assess coverage (ontology should have semantics of all the terms in those tech standards) Intersection of terms in the ontology and the domain technical standards. Annotations in the ontology identifying mapping to domain technical standards. Term origin: provenance of the term Mapping (future): real time mapping of the whole of the terms in target industry standards (regression test).		C

Redundancy	Existence of properties that can be inferred from those already in the model; Existence of properties that add no meaning	Some redundancy types are: Inferred information more than once from the relations, classes and instances found in ontology. The same/different formal definition of classes, properties or instances referring to different/same classes, properties or instances [6] For the former, these may be beneficial in conceptual ontologies, where they can add labels, localized definitions, and mapping to data / message tech models; in operational ontologies they are not needed. For the latter, these should not be in either style of ontology and should be detected and removed.	P2, P32	O

Functional adequacy

R = Reuse of other ontologies (for FIBO to refer to)

The capability of the ontologies to provide concrete functions.

Sub-characteristic	**Definition**	**Description**	**Metrics**	**Conc/ Op**

Reference ontology	Degree in which the ontology can be used as a reference resource for the particular domain the ontology is built for.	Machines can exploit better the reference knowledge offered by an ontology which have a more explicit structure. Relates to the other 'C' metrics especially domain coverage, consistency, redundancy etc. Structural characteristics all apply.		C R
Controlled vocabulary	Capability of the ontology to avoid heterogeneity of the terms. Segregate ontology issues (synonyms, no heteronyms) from terminological standard 'layer' in future SBVR support, which deals with vocabulary matters.	The labels of the ontological entities are used to avoid heterogeneity, which would complicate data processing and analysis. Ontologies provide terminology management, unifying the terms used for referring to the included knowledge entities which should have an unambiguous and non-redundant definition. ontological concepts are described with different terminologies, different meanings are assigned to the same word in different contexts and different taxonomies are examples of synonymy, polysemy and structural heterogeneity [7].	ANOnto P1, P2 P9	C
Schema and value reconciliation.	Degree in which ontology provide a common data model that can be applied to reconciliation and integration. conceptual ontology: this is a primary use case Operational: federated semantic querying applications	Integration is building a new ontology reusing other available ontologies. = reusability (in Maintainability). Mappings? Formalization. Different ontologies with subjective features and particular perspective on the world, cannot be compared without reconciliation and integration, which are necessary to interchange, migration and standardization of information and knowledge of such ontologies [See e.g. Partridge, Smith on use of upper ontology partitions - all Track C sessions]	RROnto, AROnto, Formal degree, consistency degree	C O*
Consistent search and query	The degree which the formal model and structure of the ontology provide a semantic context to evaluate which are the data wanted by the users, allowing better querying and searching methods	Structure of ontology, to make more consistent search queries. Annotation richness - does not contribute to semantic search. May be relevant in other forms of search (keywords)?	ANOnto,RROnto, AROnto,INROnto, Formal degree P8 P11 P12 P4 P30 P31 P32 P13 (P2)	O

Knowledge acquisition – representation.	Capability of the Ontology to represent the knowledge acquired. (ability to support a knowledge base of individuals)	Knowledge acquisition is the gathering, storage, and encoding of existing information. Individuals: OntoQA OOPS! - ability to support KB adequately also relationship richness etc. e.g NOMOnto	ANOnto, RROnto,NOMOnto P9, P14 - 18, P23 Test: sample data KB	C
Clustering	Degree in which the annotations of data with respect to ontology terms can be used for clustering such data against the aspects of the ontology.	Clustering can be defined as the process of organizing objects into groups whose members are more similar to each other than to individuals in other groups. SME Review. Detection of classes which have similar properties to other classes elsewhere. Also ontology modularity (maintainability) and partitioning. Plugins for this in NeOn ontology editor (Eclipse based editor). http://neon-toolkit.org/wiki/Ontology_Module_Partition for the plugin - classes per cluster, use plugin as a check. There is a paper that analyzes the labels of every class;how used in subclasses. ISO 11179	ANOnto,P2 Import to NeOn ecosystem and run plugins	C

Similarity	a. Capability of the component of the ontology to be compared for similarity b. Similarity of one ontology to another	a. There exist different similarity measures: Taxonomy similarity, Relation similarity, Attribute similarity, semantic similarity. [8] Good thing v bad thing e.g. non identification of similar classes (P12, P30) v ability to compare (Good) Availability of SKOS annotations defining the broader, narrower meanings between elements. SKOS http://skos.um.es/unescothes/ http://databases.unesco.org/thesaurus/ b. Run OntoQA on two ontologies to compare output for similarity between ontologies (numeric output); run OOPS! to find similarities expressed as similar pitfalls? Would need to inspect after. Some Ps would give clues e.g. inverses etc. Semantic similarity: OOPS! tests on synonymy etc. may be relevant. Use ontology matching tool to measure semantic similarity. Mappings e.g. Equivalent Class usage. If mappings are complex (tangled) this would show that ontologies on the same subject may be modeling it differently. If mappings more direct, then they are similar. (see refs)	RROnto, AROnto, P12 P30 SKOS usage Visual inspection b. See refs - Euzenat; Gracia	C C
Indexing and linking	Degree in which the classes defined in the ontology can act as indexes for quick information retrieval	See search and query. This covers the search aspect of this.	RROnto, AROnto,INROnto P21 P30 P31 P27	C O

Results representation	Capability of the ontology to analize complex results such as microarrays experiments	Application examples include SPARQL queries across complex networks of instruments and business entities, returning query results in tabular form. Are there quality measures for the ontology to support this? e.g. completeness and consistency of properties? Mix of datatype property v object property.	CROnto, AROnto, INROnto, NACOnto NOCOnto (No of Ancestor classes,child classes)	O
Classifying instances	Degree in which ontology Instances can be recognized as member of a certain class	Need to be able to populate a KB with individuals that are instances of the correct class in FIBO. Therefore annotation; Also misuse of classes, properties etc. (Pitfalls); Misuse of class / instance relations. Also punning.Domain and range measure also apply.	OntoQA ANOnto P20; P13 P14, 15, 16 P17 P11 P19 Tests on example instance data	C O
Text analysis	Capability of the structure of the ontology to helps detecting associations between words or concepts and classifying word types.	Vocabulary issues - SBVR and/ or SKOS layers on top of FIBO, not FIBO itself.	Formal degree ANOnto P1, P2 P9, P30 P31 P27	C
Guidance	Capability of the ontology to guide the specification of domain theories.	Ontology by capturing knowledge about a domain and encapsulating constraints about class membership provides guidance in the specification of domain theories and support decision making processes. Necessary and sufficient properties; use of OWL restrictions is important here	AROnto,INROnto P14 P15 P16 (by inspection)	C O

Decision trees	Capability of the ontology to be used building Decision trees.	Decision trees are used to represent the logical structures of classification rules for domain specific empirical data. See also Classifying Instances above. Classification and Rules: This relates partly to the ability to add business rules to operational ontology (e.g. SWRL, RF, R) on top of what's in the Conceptual Ontology content. This is for classifications that are based on relationships among other classes. Are there quality requirements to support this? For SWRL, have to be in OWL; have to have adequate information to define the rules i.e. the predicates for the rules are well defined ontological classes in the ontology	AROnto,INROnto, TMOnto RROnto CROnto NOMOnto P16; all except annotation related	O
Knowledge reuse	The degree to which The ontology knowledge can be used to build other ontologies (knowledge in the ontology) not knowledge in a KB (individuals)..	This entry: structural considerations in the ability to reuse knowledge. See Maintainability/ Reusability for other reuse aspects.	ANOnto,AROnto, INROnto, Formal degree, Consistency degree, NOMOnto, LCOMOnto P-all	C
Inference	The degree to which The formal model of the ontology can be used by reasoners to make implicit knowledge explicit.	Inference expands the knowledge base with additional information using the existing data, metadata, and rules. Measures of the expressivity will cover this	formal degree,RROnto, CROnto AROnto P4,P5, P6, P11-P18, P27-P31, P33	O
Precision	The degree to which The ontology provides the right or specified results with the needed degree of accuracy	Two aspects of precision: how fine / deep and how accurate. a. measures relating to how correct e.g. misuse of things b. Depth of class and relationship hierarchies; The business coverage as described above in 'Domain Coverage' section.	DITOnto, INROnto, P23 P28 P29 P5 P27	C O

Maintainability

The capability of ontologies to be modified for changes in environments, in requirements or in functional specifications.

Sub-characteristic	Definition	Description	Metrics	BCO/OO
Modularity	The degree to which the ontology is composed of discrete components such that a change to one component has minimal impact on other components.	Balance of size of modules, number / complexity of OWL imports,and segregation of concerns (by partition, by classification facet, by business application / use case / 3rdness "context" class relations?...)	WMCOnto, CBOnto	
Reusability	The degree to which an asset (part of) the ontology can be used in more than one ontology, or in building other assets	It can be measured by percent of classes that could be reused. Important for foundational ontologies e.g. FIBO Foundations. Important in modular ontology standards structure generally - aids extensibility of the business domain semantics. Availability for others to use - must be published Reusability internally - need not be. Examples of use See also learnability and related items. Maintainability also (reusability). Versioning etc. also important (maintainability)	WMCOnto, CBOnto DITOnto,NOCOnto, RFCOnto,NOMOnto Availability Examples	
Analysability	The degree to which The ontology can be diagnosed for deficiencies or causes of failures (inconsistences), or for the parts to be modified to be identified		LCOMOnto, WMCOnto, CBOnto DITOnto, RFCOnto,NOMOnto,	

Changeability	The degree to which The Ontology enables a specified modification to be implemented. The ease with which an ontology can be modified	Some kinds of changes in the ontology are: Add or remove classes, axioms, logical axioms, annotations, explicitly stated axioms or annotations and inferred axioms that are entailed by ontology. A measure of this is the extent to which concepts are abstracted from the specific to the most atomic (archetypical) with suitable levels in between. Not needed in an application ontology but vital in a conceptual ontology.	WMCOnto, CBOnto DITOnto,NOCOnto, RFCOnto,NOMOnto LCOMOnto, NOMOnto,	
Modification stability	The degree to which The ontology can avoid unexpected effects from modifications of the knowledge (terms, classes, properties, etc..).	Ontology changes could modify ontology specification or conceptualization and having negative effect over ontology. If classes are suitable abstracted then all changes should be additive; if not they will not be.	WMCOnto, CBOnto, RFCOnto, COMOnto, NOCOnto,	
Testability	The degree to which the ontology modified can be validated.	As well as the metrics given here, there is the possibility to create a set of standard SPARQL queries and test individuals for regression testing. Also use of ACE with a human in the loop, to validate implications of changes.	WMCOnto, CBOnto, RFCOnto, DITOnto, NOMOnto, LCOMOnto	

Compatibility

he ability of two or more software components to exchange information and/or to perform their required functions while sharing the same hardware or software environment

Sub-characteristic	Definition	Description	Metrics
Replaceability	The degree to which The ontology can be used in place of another specified Ontology for the same purpose in the same environment.		WMCOnto, CBOnto DITOnto,NOCOnto, NOMOnto
Interoperability	The degree to which the ontology can be cooperatively operable combining its knowledge with one or more other ontologies.	Ontology matching consists of matching a concept from one ontology to another.	

Transferability

The degree to which the software product can be transferred from one environment to another

Sub-characteristic	Definition	Description	Metrics
Portability	The degree in which an Ontology or one part of the ontology can be transferred from one hardware or software environment to another	The degree to which The ontology can be translated between different formal languages, or how easily the code can be moved to another language	
Adaptability	The degree to which The ontology can be adapted for different specified environments (languages, expresivity levels) without applying actions or means other than those provided for this purpose for the Ontology considered.		WMCOnto, CBOnto DITOnto, RFCOnto,

Operability

Effort needed for use, and on the individual assessment of such use, by a stated or implied set of users.

Sub-characteristic	Definition	Description	Metrics	
Appropriateness recognisability	The degree to which the Ontology enables users to recognise whether it is appropriate for their needs.	The ability to recognise the appropriateness of the functions from initial impressions of the ontology and/or any associated documentation such as Manuals, guides, comments.		C
Informativeness	Capability of the ontology to be informative. Defines how well the ontology content is communicated (particularly to modelers) so that future changes are made with an understanding of what is there now.		ANOnto AROnto	C
Learnability	The degree to which the ontology enables users to learn its application. Split this out to cover aspects of how well the information in the ontology may be understood.	Effectiveness of the user documentation and/or help system. Use of labels (simple ontology); use of SKOS annotations for definitions, editorial notes etc. Availability of business facing presentation (boxes-and-lines diagram; spreadsheet; CNL); whether the constructs in the ontology are amenable to these presentations (or have equivalent relations which are); e.g. OWL Restrictions are not learnable to all but a professional modeler, i.e. no business domain expert.	LCOMOnto, WMCOnto, CBOnto, RFCOnto, NOMOnto, NOCOnto, AROnto ANOnto **P8. Missing annotations**	
Ease of use	The degree to which the ontology makes it easy for users to operate and control it.	[how do you operate an ontology] Ease of formulation of queries?		

Helpfulness	The degree to which the Ontology provides help when users need assistance. FIBO: In future, for different business visualizations, this row would cover measures of the ontology related to those specific viz techniques.	The ontology provide clear error messages, manuals and guides for help the users, including help comprehensive, effective and easy to find. [isn't this an application requirement e.g. a reasoning based application, semantic querying application, rather than the ontology itself?]		

Reliability

Capability of an ontology to maintain its level of performance under stated conditions for a given period of time

Sub-characteristic	Definition	Description	Metrics
Error detection	The degree to which The Ontology enables users to detect faults.	Some of the faults are: Inconsistency, incompleteness and redundancy.	
Recoverability	The degree to which the Ontology can re-establish a specified level of performance and recover the data directly affected in the case of a failure.	[Surely an application not an ontology metric?]	LCOMOnto,WMCOnto, NOMOnto, DITOnto, R
Availability	The degree to which an ontology or part of it is operational and available when required for use with different applications	[Surely an application not an ontology metric?]	

Performance Efficiency

Relationship between the level of performance of the software and the amount of resources used, under stated conditions, taking into account elements such as the time response, or memory consumption.

Sub-characteristic	Definition	Description	Metrics
Response time	The degree to which The Ontology provides appropriate response and processing times from and throughput rates when performing its function (Queries and reasoning), under stated conditions.	Mathematical measures of ontology reasoning time frames / DL-safe rules, finite v non finite computational time etc. Applies to operational ontologies only (application constraint)	
Resource Utilization	The degree to which the application uses appropriate amounts and types of resources when The Ontology performs its function under stated conditions.	analogous to the above, for ontologies.	

Quality in use

Quality in a particular context of use. Quality in use is the degree to which a product used by specific users meets their needs to achieve specific goals

Sub-characteristic	Definition	Description	Metrics

Usability in use	Effectiveness in use: the degree to which specified users can achieve specified goals with accuracy and completeness in a specified context of use. Efficiency in use: The degree to which specified users expend appropriate amounts of resources in relation to the effectiveness achieved in a specified context of use. Satisfaction in use: The degree to which users are satisfied in a especified context of use. Satisfaction is further subdivided into sub-subcharacteristics: Likability (cognitive satisfaction), Pleasure (emotional satisfaction), Comfort (physical satisfaction), Trust.		
Flexibility in use	Context conformity in use: The degree to which usability in use meets requirements in all the intended contexts of use. Context extendibility in use: The degree of usability in use in contexts beyond those initially intended	Business domain views and support thereof?	

Quality metrics

OQuaRE permits the definition of the quality model in terms of quality characteristics. In this way, this standard suggests a series of quality characterstics that should be used for measuring quality. Each quality characteristic has a set of quality subcharacteristics associated and each subcharacteristic has a set of primitives measures associated. For the the definition of the metrics (primitives), the following notation has been adopted:

C1; C2; …; Cn: Classes of the ontology. RC1; RC2; …; RCk: Relationships of the class Ci. PC1; PC2; …; PCz: Properties of the class Ci. IC1; IC2; …; ICm: Individuals of the class Ci. SupC1; SupC2; …; SupCm: Direct superclasses of a given class C. Thing: Root class of the ontology.

Some of the metrics like Coupling Between Objects (CBO), Depth of Inheritance Tree (DIT), Number Of Children (NOC), Response For a Class (RFC), Weighted Method Count (WMC), (Chidamber and Kemerer, 1994)[9] and Number Of local Methods (NOM) by (Li and Henry, 1993)[10] were selected from Software Engineering and, in particular, Object-oriented Programming (OOP) and adapted to ontologies. Despite ontologies and object oriented design having different properties, there are a series of shared notions as the existence of classes, individuals and properties that can be exploited to adapt OOP metrics to ontologies. And reused other metrics developed by the ontology engineering community, especially for the structural properties from, for instance, Yao, Orme, and Etzkorn (2005)[11] or Tartir and Arpinar (2007)[12] and Gangemi, Catenacci, Ciaramita, and Lehmann [13]

LCOMOnto - Lack of Cohesion in Methods

Semantic and conceptual relatedness of classes. It can be used to measure the separation of responsibilities and independence of components of ontologies LCOMOnto=∑path(|C(leaf)i|)/m , where path|C(leaf)i| is the length of the path from the leaf class i to Thing, and m is the total number of paths in the ontology

WMCOnto - Weighted Method Count

Mean number of properties and relationships per class WMCOnto(∑|PCi|+∑|RCi|)⁄∑|Ci| , where Ci is the i-th class in the ontology

DITOnto - Depth of subsumption hierarchy

Length of the largest path from Thing to a leaf class DITOnto=Max (∑D|Ci|), where Ci are the classes and D|Ci| is the length of the path from the i-th leaf class of the ontology to Thing

NACOnto - Number of Ancestor Classes

Mean number of ancestor classes per leaf class. It is the number of direct superclasses per leaf class NACOnto=∑|SupC(Leaf)i|/∑|C(leaf)i)|

NOCOnto - Number of Children

Mean number of direct subclasses. It is the number of relationships divided by the number of classes minus the relationships of Thing $NOCOnto=\sum|RC_i|/(\sum|C_i|-|RThing|)$

CBOOnto - Coupling between Objects

Number of related classes. It is the average number of the direct parents per class minus the relationships of Thing $CBOOnto=\sum|SupC_i|/(\sum|C_i|-|RThing|)$

RFCOnto - Response for a class

Number of properties that can be directly accessed from the class $RFCOnto=(\sum|PC_i|+\sum|SupC_i|/(\sum|C_i|$

NOMOnto - Number of properties

Number of properties per class $NOMOnto=\sum|PC_i|/\sum|C_i|$

RROnto - Properties Richness

Number of properties defined in the ontology divided by the number of relationships and properties $RROnto=\sum|PC_i|/\sum(|RC_i|+\sum|C_i|)$

AROnto - Attribute Richness

Mean number of attributes per class $AROnto=\sum|AttC_i|/\sum|C_i|$

INROnto - Relationships per class

Mean number of relationships per class $INROnto=\sum|RC_i|/\sum|C_i|$

CROnto - Class Richness

Mean number of instances per class $CROnto=\sum|IC_i|/\sum|C_i|$; where IC_i, is the set of individuals of the C_i

ANOnto - Annotation Richness

Mean number of annotations per class $ANOnto=\sum|AC_i|/\sum|C_i|$; where C_i is the i-th class in the ontology

TMOnto - Tangledness

Mean number of parents per class, of properties and relationships per class TMOnto=∑| RCi| /∑|Ci|-∑|C(DP)i|; where Ci is the i-th class in the ontology and C(DP)i is thei-th class in the ontology with more than one direct parent

References

1. ↑ A. Duque-Ramos, J.T. Fernández-Breis, R. Stevens, N. Aussenac-Gilles, Oquare: a square-based approach for evaluating the quality of ontologies Journal of Research and Practice in Information Technology 43(2011) 159-73.
2. ↑ ISO25000 2005. ISO/IEC 25000 2005, Software engineering - Software product quality requirements and evaluation (SQuaRE) - guide to square (ISO/IEC 25000). Geneva, Switzerland: International Organization for Standardization.
3. ↑ GANGEMI, A., CATENACCI, C., CIARAMITA, M. and LEHMANN, J. (2006): Modelling ontology evaluation and validation. Semantic Web: Research and Applications, Proceedings, 4011: 140 –154.
4. ↑ STEVENS, R. and LORD, P. (2009): Application of ontologies in bioinformatics. In: BERNUS, P., BŁAŻEWICS, J., SCHMIDT, G., SHAW, M., STAAB, S. and STUDER, R. (eds.) Handbook on Ontologies. Springer Berlin Heidelberg.
5. ↑ http://www.obofoundry.org/wiki/index.php/FP_002_format
6. ↑ FAHAD M., QADIR, M. (2008). A Framework for Ontology Evaluation. In Proceedings International Conference on Conceptual Structures (ICCS'08), Toulouse, France, July, page 711.
7. ↑ KIU, C.-C., & LEE, C.-S. (2006). Ontology Mapping and Merging through OntoDNA for Learning Object Reusability. Educational Technology & Society, 9 (3), 27-42
8. ↑ MAEDCHE A., ZACHARIAS V. (2002). Clustering Ontology-based Metadata in the Semantic Web. In Proceedings 6th European Conference on Principles of Data Mining and Knowledge Discovery, Springer-Verlag, London, Uk, 348-360
9. ↑ CHIDAMBER, S.R. and KEMERER, C.F. (1994): metric suite for object oriented design. IEEE Transactions on SoftwareEngineering, 467–493.
10. ↑ LI, W. and HENRY, S. (1993): Object-oriented metrics that predict maintainability. Journal of Systems and Software, 23:111–122.
11. ↑ YAO, H., ORME, A. and ETZKORN, L. (2005): Cohesion metrics for ontology design and application. Journal of ComputerScience, 1.
12. ↑ TARTIR, S. and ARPINAR, I.B. (2007): Ontology evaluation and ranking using OntoQA. ICSC 2007: International Conference on Semantic Computing, Proceedings, 185–192.
13. ↑ GANGEMI, A., CATENACCI, C., CIARAMITA, M. and LEHMANN, J. (2006): Modelling ontology evaluation and validation. Semantic Web: Research and Applications, Proceedings, 4011: 140 –154.