# Evaluating the Fitness of a Domain Ontology to Formalized Stakeholder Requirements

Alexander Vasileyko [0000-0002-8894-326X] and Vadim Ermolayev [0000-0002-5159-254X]

Department of Computer Science, Zaporizhzhia National University, Zhukovskogo st. 66, Zaporizhzhia, Ukraine
vasileyko.alex@gmail.com, vadim@ermolayev.com

**Abstract.** This position paper presents the Ph.D. project proposal, by the first author, aimed at developing the methodological and formal approaches, and also software tools for evaluating the fitness of a domain ontology to the formalized stakeholder requirements in a domain. The paper describes the objectives and presents the vision of the solution to be developed as a three-step process including analysis, mapping, and fitness evaluation. The paper also presents the initial steps in finding the relevant techniques that may help attack the outlined problems. The project plans to develop novel approaches and techniques, in particular, for formalized requirements analysis, extending a mapping language, fitness computation and visualization. These approaches will be implemented in a software solution that will help knowledge engineers evaluate their results more efficiently and effectively and also prioritize their ontology refinement work based on objective fitness measures. The software will be experimentally validated as outlined in the paper.

**Keywords:** Ontology evaluation, Ontology refinement, Ontology alignment, Ontology mapping, Ontology fitness, Domain knowledge stakeholder, Vote, OntoElect, Gravitation Framework.

## 1 Introduction

This paper presents a position and vision towards a Ph.D. project aimed at developing the methodological approaches and software tools for evaluating the fitness of a domain ontology to the formalized stakeholder requirements in a domain. As a theoretical and methodological basis, this project uses OntoElect [1], more particularly its third phase of ontology evaluation against the formalized stakeholder requirements. These requirements are presented as ontology fragments in a form of a UML model and OWL + SWRL code coming out from the conceptualization process [2].

An ontology evaluation process helps evaluate ontology fitness to the stakeholder requirements, comprising its coverage of and accuracy regarding the desired interpretation of a domain. In ontology development process, ontology engineers need to possess a way to evaluate their output based on the requirements by the knowledge stakeholders. This evaluation can be regarded as unbiased if based on the use of quantitative objective measure(s). The measures could be easier and more rigorously introduced if the requirements are presented formally – such that their formal straightforward comparison to the ontology is enabled. Fitness [3] is regarded as one of the most appropriate objective integral ontology quality measures in this project. Having fitness measured in an ontology evaluation process, will allow to analyze the flaws in the ontology. Moreover, there are several problems which this evaluation process can help reveal and hopefully solve.

**Usability and quality.** Once the usability and quality of the ontology is assessed, it helps understand the degree of how much the stakeholders in the domain and the ontology engineer can trust this particular ontology.

**Reusability.** Due to a continually increasing numbers of ontologies, ontology engineering process becomes more centric to reusing existing ontology fragments, modules, or entire ontologies. A mature fragment or ontology with high fitness to the stakeholders' requirements in the domain may be more readily re-used.

In practical (engineering) terms, it is planned that the project develops and deploys the suite of instrumental software tools that improve domain ontology evaluation process and decrease the effort to be spent by a knowledge engineer in evaluating their ontology under development.

For proving the validity of our concept and early development results, the project plans to undertake experimental evaluation using the W3C OWL-Time ontology as it is the most widely used time ontology [5]. The requirements to the Syndicated Ontology of Time (SOT) [6] will be used as available as the background knowledge in the domain.

The remainder of the paper is structured as follows. Section 2 describes our motivation for this PhD project. Section 3 analyzes the related work. Section 4 outlines the OntoElect methodology, with a particular detail regarding its requirements evaluation phase. The planned research workflow for the planned ontology evaluation approach is presented in Section 5. Section 6 gives the plans for the future work and concludes the paper.

## 2 Motivation

It is hard to overstate the importance of evaluation in ontology development. Both the knowledge stakeholders in the domain in which the ontology is developed and deployed, and the ontology engineers are interested to have a quality ontology that fits the (majority of the) acquired requirements. Hence, both parties need an objective way to evaluate the ontology and also control its development and refinement in a pro-active manner. Having an instrument for such a control makes: (i) the stakeholders confident that their requirements are met and knowledgeable to which degree the requirements are met; (ii) the ontology engineers equipped with the arguments for proving the utility of their result and promoting its reuse.

Ontology evaluation, however, is perhaps the most immature part in ontology engineering. Therefore, the effort is currently increasing to make ontology evaluation a real engineering part of ontology development – see also the related work in Section 3. Measurable and unbiased ontology evaluation allows enhancing the development process by providing rigorous feedback regarding the requirements. It helps check ontology correctness and completeness, and then presents the set of aspects to further guide the refinement of the ontology. Moreover, it promotes knowledge sharing and reuse, which significantly reduces development time and cost. Unfortunately, ontology evaluation, like any evaluation or validation step in development, requires substantial effort. Therefore, it is also demanded to use automated or semi-automated techniques to speed up ontology evaluation process, make it less complex, less laborious, more rigorous and objective, and more complete.

This Ph.D. project aims to refine the methodology and develop the suite of software tools for evaluating the fitness, to the stakeholder requirements, of the ontology (or several competing ontologies) describing an arbitrary domain. Fitness is planned to be objectively measured using an extended mapping language for relating the fragments of an ontology and corresponding formalized requirements and scoring their similarity to each other. It is also planned that the results of these measurements will be presented in a visual form for which the approach of [7] will be used.

## 3 Related Work and Research Objectives

Research in ontology evaluation still remains timely and demanded. Relevant activities resulted in the provision of the rules, guidelines, and different suggested ways to perform ontology evaluation. Among many other important works, the following methodologies, methods, and tools offer important insights, bits of background knowledge and technology to the presented Ph.D. project.

**OntoElect** is an ontology refinement methodology. It facilitates, in an unbiased and measured way, finding out what needs to be improved in the domain ontology to better meet the requirements of the knowledge stakeholders in a domain. OntoElect may also be used to cross-evaluate different ontologies, describing the same domain, by comparing their fitness to stakeholder requirements [1].

**Klagenfurt Conceptual Predesign Model (KCPM)** represented requirements in a lightweight formalized form by concentrating on the structural, functional, and behavioral terminology of an application domain [8].

**METHONTOLOGY** is one of the most comprehensive ontology engineering methodologies for building ontologies from scratch, reusing other ontologies, and re-engineering them. This framework allows ontology construction at the conceptual level, and comprises evaluation, conceptualization, management, configuration, integration, and implementation [9].

**NeOn** methodology provides the methodological guidelines for the formal evaluation and building stand-alone ontologies as well as ontology networks. The methodology supports the collaborative features of ontology development, as well as the dynamic evolution of ontology networks [10].

**OntoClean** method describes how to clean the concept taxonomies and makes explicit ontological commitments assumed in the definitions of the ontology terms [11].

**Denny Vrandečić's Framework** provides six methods for ontology evaluation, namely schema validation, pattern discovery using SPARQL, normalization, metric stability, representational misfit, unit testing [12].

**OOPS! (OntOlogy Pitfall Scanner)** is a web-based (semi)automatic ontology diagnosis system for detecting possible pitfalls that could lead to modeling errors. This tool helps ontology engineers in ontology development process and divides the process into diagnosis and repair steps. Currently, OOPS! provides mechanisms to detect some pitfalls automatically, thus helps developers in the diagnosis activity [13].

Ontology evaluation is a broad and developing field that still has sufficient space to make efforts in resolving the problems that are not yet fully solved. It is worth noting that each of the works mentioned above covers a particular facet of the field, where effort still needs to be applied. However, an inspiring fact regarding the abovementioned approaches is that, in their constellation, these allow presenting an ontology evaluation task as the one answering three important questions [1]: (i) is the evaluated ontology correct? (ii) is it complete; and (iii) does it have sufficient quality? It looks straightforward that the answers to these questions have to be sought by comparing the ontology describing a domain to the requirements in this domain.

This PhD project takes in the insights and background knowledge from these predecessors. To push the state of the art in ontology evaluation forward, it has the ambition to develop, in frame of the OntoElect Evaluation Phase [1], a fully instrumented engineering approach to answer the important questions. In particular:

- It plans to use ontology fitness to domain stakeholder requirements as an integral usability measure of an ontology
- It plans to present the requirements to an ontology as OWL DL[1] and SWRL[2] fragments
- It plans to develop a formal mapping approach to compare formalized requirements to OWL DL (+SWRL) ontologies
- It plans to visualize the fitness of an ontology to the set of domain requirements using the gravitation approach [7]
- It will develop the suite of instrumental software tools to support the ontology evaluation workflow and make ontology less laborious by partial automation
- For experimental evaluation and validation of the proof of concept, it will elaborate the use cases in the domains of Time Representation and Reasoning and Knowledge Management

## 4 Ontology Evaluation Workflow

Evaluation phase is essential in ontology development process to identify the fitness of the ontology to the requirements acquired from domain knowledge stakeholders. To find out how much an ontology fits to the requirements, it is necessary to reveal its similarities and dissimilarities to the requirements. This information will also play an important role in further ontology refinement process. Furthermore, based on this

---

1 OWL DL: https://www.w3.org/TR/owl-ref/

2 SWRL: https://www.w3.org/Submission/SWRL/

knowledge, ontology engineers and domain knowledge stakeholders will have more confidence in the quality of the resulting artifact and also the development process.

This project focuses on the evaluation of domain ontologies such that: (i) there is sufficient evidence of the formally represented requirements of the domain knowledge stakeholders; and (ii) the requirements are specified in OWL+SWRL. Hence, it is assumed that the requirements are available as OWL + SWRL fragments. These ontological fragments will be analyzed and used in the evaluation process.

This workflow supposes that ontology engineers perform ontology evaluation and validation process using the instrumental software. It is envisioned that the workflow involves the following phases: (i) requirements analysis; (ii) mapping the requirements to the ontology; and (iii) fitness evaluation. The details of these phases are presented as follows. The practical result of this research should be the suite of evaluation and validation software tools that help ontology engineers perform evaluation (semi)automatically.

### 4.1 Analysis

The objective of this phase is to analyze the stakeholders' requirements, for the particular domain ontology, not as individual unrelated fragments, but in their entirety. This analysis of correctness is required to verify the consistency of the specification of different types of properties that span across several requirements – in order to assure that these were given in an unambiguous and harmonized way. The focus of this step is finding the inconsistencies in domain properties or contradictions in object properties connecting different requirements. Thus, the input of this phase is going to be the set of the ontological fragments, representing the requirements, and the concept taxonomy with the anchor concepts of the requirements organized in a subsumption + meronymy hierarchy. The inputs will be provided after their conceptualization and formalization [2] both as UML models OWL + SWRL code fragments. It is assumed that the concept taxonomy has already been validated by the ontology engineer. So, only the set of requirements (as a graph potentially connected by subsumption, meronymy, and other properties) will be analyzed for inconsistencies. The inconsistencies will be sought in properties that represent the same semantics but are specified differently in different requirements (e.g., one property suggests that time is continuous with stamps represented by (super)reals, but the other states that time is discrete).

It is also planned that inconsistencies will be evaluated not only at a schema level but also between the instances of the ontology, if those are made available. It is supposed that the requirements may contain a few instances as ground facts to support their matter. Finally, the properties of some instances may contradict also the schema – which has also to be evaluated. Using the same example as above, one may potentially notice that in several provided instances of a time point the time stamp values of time are given as integers, however the schema states that this property is of a real type.

In different fragments, properties may be specified with different restrictions. These restrictions may be given as OWL restrictions but also as SWRL rules included in the fragment codes. A noteworthy example of such a restriction kind that may span across several requirements is instance-based disjointness. Indeed, your `cat`, as an individual that: (i) cannot be allowed as the instance of both a `TerresticalAnimal` and `AquaticAnimal` (at least within the same `period of time`); and (ii) may be considered by you, as a stakeholder of the knowledge about your `cat`, as belonging to both categories – as it `likes to swim`. Instance-based evidence is straightforwardly very helpful also in this case.

Semantically the same property may be specified using different syntactic labels in different requirements. This kind of inconsistency needs also to be detected and corrected at the analysis step. NLP-based techniques, in particular string similarity measures (e.g. [14]) may be used in these cases both for properties and instances.

Finally, it is worth mentioning that a fully automated approach may not reliably work for such sorts of analysis as the relevant techniques still cannot offer an appropriately high level of quality to be fully trusted. Therefore, in this project a semi-automated approach will be pursued. A tool will detect potential inconsistencies and present these to the knowledge engineer for the verification of validity. The knowledge engineer will use the tool to overlook the detected problems and edit the ontological fragments, either in their UML model or OWL + SWRL representation.

### 4.2 Mapping and Transformation

Ontology mapping in the context of this project can be described as a process taking valid (corrected) and approved formalized stakeholders' requirements (ontology fragments) and a domain OWL DL(+SWRL) ontology as the input and returning their detailed mappings, at the instance, property, and concept levels.

The objective of the mapping step is to detect and measure the similarity and dissimilarity between the requirements, and relevant domain ontology contexts. Let's consider the further mapping steps for the implementation. Also, it would be more efficient to describe the workflow by concentrating on the single requirement as an example.

1. **Finding the context**. Both the requirement and domain ontology are regarded as labeled oriented graphs. The graph of a formalized requirement is the representation of its elements, which are the core concept (anchor), its properties, and instances (see Fig. 1(a)). So, the path distance from its anchor to the periphery may quite rarely exceed 1 edge. The graph of an ontology is much bigger as it includes all its concepts and properties (see Fig. 1(b)). The task for this step is to find the context within the ontology graph that best matches to the requirement graph. To define the relevant context, we are going to use several techniques: (i) string similarity measures; (ii) comparing OWL/SWLR formalized fragments; (iii) a topological approach. If found, it would be further interpreted as the one implementing this requirement, probably in part. If not found, the evidence will be noted that the requirement is perhaps not implemented in the ontology. The technique for this sort of matching is superimposing the requirements graph onto the structural contexts of ontology concepts, one after one, and measuring their similarity [15] in a balanced way – using a carefully chosen set of strings and structural similarity measures. As a result, the set of mappings is

built that relate the elements of the requirement to the elements of the found structural context within the ontology.

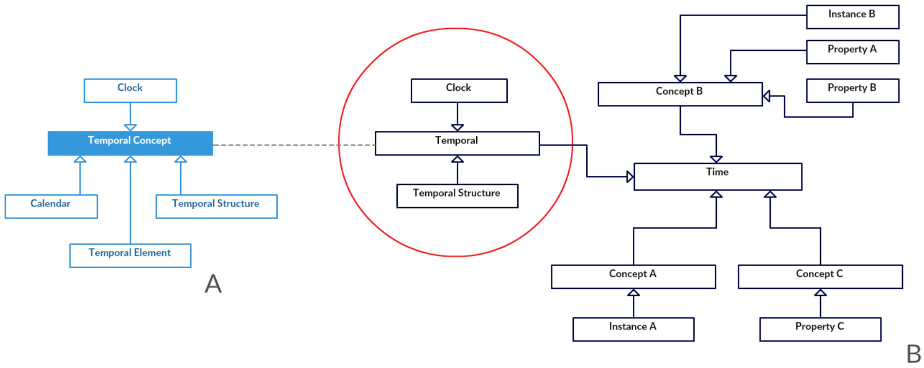


**Fig. 1**: The graphs of a requirement (A) and ontology (B)

2. **Scoring the Context**. Each of the mappings found for the structural context is an equivalence mapping which is satisfied partially – i.e. to the extent given by the measured similarity degree. So, if for example property *ao* (in the ontology) is 60 per cent similar to the property *ar* (in the requirement) is could be noted that *ao* meets *ar* with the ratio 0.6 and does not satisfy it with the ratio of 0.4. If no mapping was found, then the (structural context of) the ontology does not meet the requirement given by the requirement element at all – so dis-similarity ratio will be set to 1.0. Hence, provided that every element in the requirement has its significance score [1], the absolute scores of dis-satisfaction and satisfaction could be computed for all the elements. The score of the context will be formed as two sums: of dis-satisfaction and satisfaction scores for the elements of the requirement. Please refer to Fig. 2 for an example of scoring the similarity of the Instant context in the W3C OWL-Time ontology[3] to the TimeInstant requirement [1]. Taking into account, that it is challenging to compare the similarity of two contexts, we are going to compare the properties of these contexts. The rule is that each property has datatype and measure characteristics. Thus, if both contexts have the properties before and after with datatype integer and measure seconds, we can consider that similarity between the contexts will be increased. And vice a verse, if some of the property missed or has differed characteristics in one of the contexts the similarity will be decreased.

Currently we have noted two problems that need to be solved in order to elaborate and implement this two-step mapping activity.  The first is that a refined and extended equivalence mapping language that accounts for (dis-)similarity and (dis-)satisfaction values needs to be proposed and implemented in the instrumental software. The second is that a balanced set of similarity measures has to be selected to provide reliably complete mappings. To do that a set of patterns needs to be carefully designed and the measures have to be evaluated experimentally to be finally selected.

[3] W3C OWL-Time: https://www.w3.org/TR/owl-time/

Finally, it has to be mentioned that a fully automated approach, again, may not reliably work for mappings and, in particular for similarity measurements. The available techniques still cannot offer an appropriately high level of quality to be fully trusted. Therefore, in this project a semi-automated approach will be pursued. A tool will detect potential mappings and offer its estimates of the similarity measurements. These estimates may be computed following the approach of [15] or a similar technique. These will be presented to the knowledge engineer for the verification of validity. The knowledge engineer will use the tool to overlook the proposed mappings and edit these if deemed necessary.

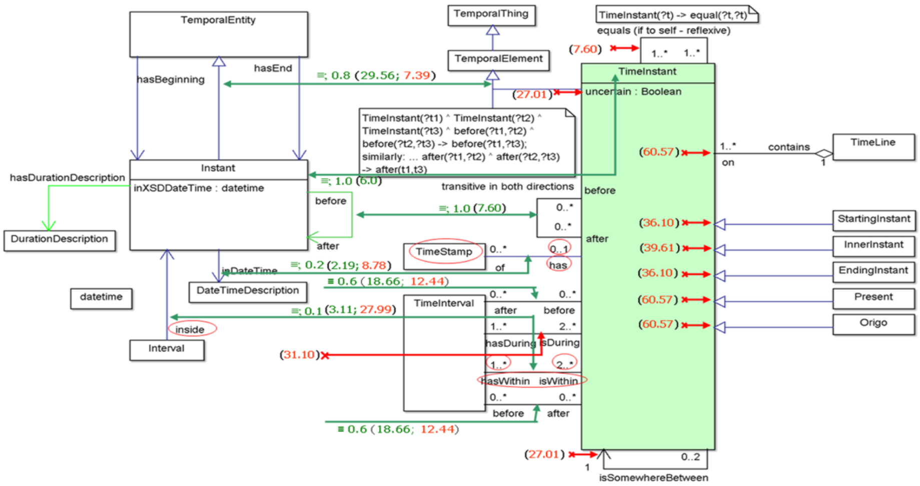


**Fig. 2.** The scoring of the satisfaction of the TimeInstant requirement by the OWL-Time ontology, adopted from [1].

### 4.3 Fitness Evaluation and Visualization

The final step of the evaluation workflow is built around using the scores, generated at the mapping step, for computing and presenting the integral fitness of the evaluated ontology to the given requirements. This fitness is computed as the sum of satisfaction / dis-satisfaction scores. Partial sums of these scores for different ontology parts may also help reveal which of the parts need more effort to make the ontology better fit to the requirements.

An example of the manually calculated satisfaction / dis-satisfaction scores of the OWL-Time ontology regarding the four most significant SOT requirements [5] are given in Table 1. It may be seen in Table 1 that the imperfections of the ontology in the context of a TimeInterval cause the most significant losses in fitness. So, it might be reasonable to focus on this part of the ontology in the next iteration of its refinement.

For better perception by the domain knowledge stakeholders and ontology engineers, the fitness of the ontology to the requirements can be visualized. For visualization the proposal of the Gravitation Framework [7] is regarded as quite appropriate. In this framework, the elementary scores of satisfaction / dis-satisfaction, measured at the mapping step and normalized in the interval of [0, 1], are regarded as the components of a "domain gravitation" field. The equilibrium state is reached by an ontology at a distance l from the center of gravitation when the superposition of elementary satisfaction and dis-satisfaction "forces" goes to zero. This distance in fact visualizes how far is the ontology from perfectly meeting the requirements.

**Table 1.** Example. The fitness of OWL-Time to the 4 most significant TIME requirements. Adopted from [1]

| **Key Element** | **Fully Implemented** | **Partially Implemented** | **Missing Features** |
|---|---|---|---|
| | **Cumulative Satisfaction / Dis-satisfaction count** | | |
| TimeInterval context (**1231.04**) | **351.59** | **97.05 / 103.15** | **679.25** |
| TimeInstant context (**509.96**) | **13.60** | **53.52 / 56.60** | **386.24** |
| TimeLine (**57.29**) | --- | --- | **57.29** |
| Clock (**16.25**) | --- | --- | **16.25** |
| Total: | **515.76** | | **1298.78** |
| **Fitness**: | | **0.3971** | |

## 5 Planned Evaluation

The envisioned approach, together with the instrumental software tools, for evaluating how well an ontology, in arbitrary domain, fits the requirements needs to be experimentally evaluated and validated. A way to validate the solution is to offer it to knowledge engineers for a trial. Further, their impression of the usability and performance of the solution is compared to their normal mode of work – without the solution.

In the use case for evaluation, it is planned to exploit the requirements collected in our working repository of SOT. SOT is developed using OntoElect as the ontology engineering methodology. The repository belongs to our group and therefore is fully available as background knowledge. In evaluation, it is planned to compare the outputs of the developed tools to the same outputs developed by human knowledge engineers. The ontologies for which their fitness will be measured are those reviewed in [5].

For user validation, the knowledge engineers will be offered to answer questionnaires about the effort spent in this activity and their subjective assessments of the usability and usefulness of the tools.

After the evaluation and validation of the solution in the SOT use case, another use case in a different domain will be elaborated. One of the potential candidate domains is Knowledge Management. For a part of this domain the preliminary work on extracting features has already been done [16].

## 6 Some Conclusions

This position paper presented the proposal of the Ph.D. project, by its first author, that develops the methodological approaches and software tools for evaluating the fitness of a domain ontology to the formalized stakeholder requirements in a domain. The approach taken by the project is domain independent.

The paper described project objectives and presented the vision of the solution as a three step process including analysis, mapping, and fitness evaluation. The paper also described our initial steps in finding the relevant techniques that may help attack the outlined problems. Some of the problems have only partial solutions, as described in the related work. Therefore, the project plans to develop novel approaches and techniques, in particular, for formalized requirements analysis, extending a mapping language, fitness computation and visualization. These approaches will be implemented in a software solution that will help knowledge engineers evaluate their results more efficiently and effectively and also prioritize their ontology refinement work based on objective fitness measures. The software will be experimentally validated as presented in the paper.

In particular, in our future work we will be looking for the answers to several important questions related to the three steps of our envisioned process. The first question is about the identification of the contexts in an ontology that are relevant to a particular requirement. The second question is about a reliable way to estimate the scores of satisfaction/dissatisfaction in the equivalence mappings. Finally, we have to elaborate the evaluation approach for the developed technique and tools and the use cases, including industrially strong ontologies.

## References


1. Ermolayev, V.,: OntoElecting requirements for domain ontologies: the case of time domain. EMISA Int J of Conceptual Modeling, 13(Sp.I.), 86–109 (2018)
2. Moiseyenko, S., Ermolayev, V.: Conceptualizing and formalizing requirements for ontology engineering. In: Antoniou, G., Zholtkevych, G. (eds.) Proc. ICTERI 2018 PhD Symposium, CEUR-WS, online (2018) – to appear
3. Tatarintseva, O., Ermolayev, V., Keller, B., Matzke, W.-E.: Quantifying ontology fitness in OntoElect using saturation- and vote-based metrics. In: Ermolayev, V., et al. (eds.) Revised Selected Papers of ICTERI 2013, CCIS, vol. 412, pp. 136–162 (2013)
4. Hobbs, J. R., Pan, F.: Time ontology in OWL. W3C working draft (2006) https://www.w3.org/TR/2006/WD-owl-time-20060927/
5. Ermolayev, V., Batsakis, S., Keberle, N., Tatarintseva, O., Antoniou, G.: Ontologies of time: review and trends. IJCSA 11(3), 57–115 (2014)
6. Ermolayev, V.: Toward a syndicated ontology of time for the semantic web. AIFB Oberseminar, Karlsruhe Institute of Technology, 02-Feb (2016)
7. Ermolayev, V.: The law of gravitation for ontologies and domains of discourse. Computer Science Journal of Moldova, 23(2), 209–236 (2015)
8. Fliedl, G., Kop, C., Mayr, H.C.: From textual scenarios to a conceptual schema. In: Data & Knowledge Engineering 55(1), 20–37 (2005)

9. Gómez-Pérez, A., Fernández-López, M., Corcho, O.: Ontological engineering. 1st ed., Springer-Verlag, London (2003)
10. Suárez-Figueroa, M. C., Gómez-Pérez, A., Motta, E., Gangemi, A.: Ontology engineering in a networked world. Springer-Verlag, Berlin, Heidelberg (2012)
11. Guarino, N., Welty, C.A.: An overview of OntoClean. In: Staab, S., Studer, R. (eds.) Handbook on Ontologies. Springer, Berlin, Heidelberg (2004)
12. Vrandečić, D.: Ontology evaluation. PhD thesis, Fakultät für Wirtschaftswissenschaften, KIT, Karlsruhe, Germany (2010)
13. Poveda-Villalón, M.: Ontology evaluation: a pitfall-based approach to ontology diagnosis. PhD Thesis, Universidad Politécnica de Madrid (2016)
14. Chugunenko, A., Kosa, V., Popov, R., Chaves-Fraga, D., Ermolayev, V.: Refining terminological saturation using string similarity measures. In: Ermolayev, V. et al. (eds.) Proc. ICTERI 2018, CEUR-WS, online (2018) – to appear
15. Ermolayev, V., et al.: A strategy for automated meaning negotiation in distributed information retrieval. In: Gil, Y., et al. (eds.) ISWC 2005. LNCS, vol. 3729, pp. 201–215 (2005)
16. Kosa, V., Chaves-fraga, D., Naumenko, D., Yuschenko, E., Badenes-Olmedo, C., Ermolayev, V, Birukou, A.: Cross-evaluation of automated term extraction tools by measuring terminological saturation. In: Bassiliades, N., et al. (eds.) ICTERI 2017, Revised Selected Papers, CCIS, vol. 826, pp. 135–163, Springer, Cham (2018)