Extending OWL Ontology Visualizations with Interactive Contextual Verbalization

Uldis Bojārs[1,2], Renārs Liepiņš[1], Normunds Grūzītis[1,2], Kārlis Čerāns[1,2], and Edgars Celms[1,2]

[1] Institute of Mathematics and Computer Science, University of Latvia
Raina bulvaris 29, Riga, LV-1459, Latvia
[2] Faculty of Computing, University of Latvia
Raina bulvaris 19, Riga, LV-1459, Latvia
uldis.bojars@lumii.lv, renars.liepins@lumii.lv, normunds.gruzitis@lumii.lv, karlis.cerans@lumii.lv, edgars.celms@lumii.lv

Abstract. To participate in Semantic Web projects, domain experts need to be able to understand the ontologies involved. Visual notations can provide an overview of the ontology and help users to understand the connections among entities. However, users first need to learn the visual notation before they can interpret it correctly. Controlled natural language representation would be readable right away and might be preferred in case of complex axioms, however, the structure of the ontology would remain less apparent. We propose to combine ontology visualizations with contextual ontology verbalizations of selected ontology (diagram) elements, interactively displaying controlled natural language (CNL) explanations of OWL axioms corresponding to the selected visual notation elements. Thus, the domain experts will benefit from both the high-level overview provided by the graphical notation and the detailed textual explanations of particular elements in the diagram.

Keywords: OWL · Ontology visualization · Contextual verbalization · Controlled natural language

1 Introduction

Semantic Web technologies have been successfully applied in pilot projects and are transitioning toward mainstream adoption in the industry. In order for this transition to go successfully, there are still hurdles that have to be overcome. One of them are the difficulties that domain experts have in understanding mathematical formalisms and notations that are used in ontology engineering.

Visual notations have been proposed as a way to help domain experts to work with ontologies. Indeed, when domain experts collaborate with ontology experts in designing an ontology "they very quickly move to sketching 2D images to communicate their thoughts" [8]. The use of diagrams has also been supported by an empirical study done by Warren et al. where they reported that "one-third [of participants] commented on the value of drawing a diagram" to understand what is going on in the ontology [21].

Despite the apparent success of the graphical approaches, there is still a fundamental problem with them. When a novice user wants to understand a particular ontology, he or she cannot just look at the diagram and know what it means. The user first needs to learn the syntax and semantics of the notation – its mapping to the underlying formalism. This limitation has long been noticed in software engineering [18] and, for this reason, formal models in software engineering are often translated into informal textual documentation by systems analysts, so that they can be validated by domain experts [4].

A similar idea of automatic conversion of ontologies into seemingly informal controlled natural language (CNL) texts and presenting the texts to domain experts has been investigated by multiple groups [14,20,9]. CNL is more understandable to domain experts and end-users than the alternative representations because the notation itself does not have to be learned, or the learning time is very short. Hoverer, the comparative studies of textual and graphical notations have shown that while domain experts that are new to graphical notations better understand the natural language text, they still prefer the graphical notations in the long run [13,17]. It leads to a dilemma of how to introduce domain experts to ontologies. The CNL representation shall be readable right away and might be preferred in case of complex axioms (restrictions) while the graphical notation makes the overall structure and the connections more comprehensible.

We present an approach that combines the benefits of both graphical notations and CNL verbalizations. The solution is to extend the graphical notation with interactive contextual verbalizations of the axioms that are represented by the selected graphical element. The graphical representation gives the users an overview of the ontology while the contextual verbalizations can explain what the particular graphical element means. Thus, domain experts that are novices in ontology engineering shall be able to learn and use the graphical notation rapidly and independently without special training.

Throughout this paper we refer to the OWLGrEd visual notation and ontology editing and visualization tools that are using it. The OWLGrEd notation [1] is a compact and complete UML-style notation for OWL 2 ontologies. It relies on Manchester OWL Syntax [7] for certain class expressions. This notation is implemented in the OWLGrEd ontology editor[3] and its online ontology visualization tool[4] [11]. The approach proposed in this article is demonstrated on an experimental instance of the OWLGrEd visualization tool.

In Section 2, we present the general principles of extending graphical ontology notations with contextual natural language verbalizations. In Section 3, we demonstrate how the proposed approach may be used in practice by extending ontology visualizations in the OWLGrEd notation with interactive contextual verbalizations in controlled English. In Section 4, we discuss the benefits and limitations of our approach. Section 5 discusses related work while Section 6 summarizes the article.

[3] `http://owlgred.lumii.lv`
[4] `http://owlgred.lumii.lv/online_visualization/`

2 Extending Graphical Notations with Contextual Verbalizations

This section describes the proposed approach for contextual verbalization of graphical elements in ontology diagrams, starting with a motivating example. We are focusing particularly on OWL ontologies, assuming that they have already been created and that the ontology symbols (names) are lexically motivated and consistent, i.e., in this paper we do not consider the authoring of ontologies, although the contextual verbalizations might be helpful in the authoring process as well and it would provide a motivation to follow a lexical and consistent naming convention.

2.1 Motivating Example

In most diagrammatic OWL ontology notations, object property declarations are shown either as boxes (e.g. in VOWL [12]) or as labeled links connecting the property domain and range classes (e.g. in OWLGrEd [1] or Graffoo [3]). Figure 1 illustrates a simplified ontology fragment that includes classes *Person* and *Thing*, an object property *likes* and a data property *hasAge*. This fragment is represented by using three alternative formal notations: Manchester OWL Syntax [7], VOWL and OWLGrEd. As can be seen, the visualizations are tiny and may already seem self-explanatory. Nevertheless, even in this simple case, the notation for domain experts may be far from obvious. For example, the Manchester OWL Syntax uses the terms *domain* and *range* when defining a property, and these terms may not be familiar to a domain expert. In the graphical notations, the situation is even worse because the user may not even suspect that the edges represent more than one assertion and that the assertions are far-reaching. In the case of *likes* in Figure 1, it means that everyone that likes something is *necessarily* a person.

We have encountered such problems in practice when introducing ontologies visualized using the OWLGrEd notation to users familiar with the UML notation. Initially, the users were misunderstanding the meaning of the association edges. For example, they would interpret that the edge *likes* in Figure 1 means "persons *may* like persons", which is true, however, they would also assume that other disjoint classes could have this property, which is false in OWL because multiple domain/range axioms of the same property are combined to form an intersection. Thus, even having a very simple ontology, there is a potential for misunderstanding the meaning of both the formal textual notation (e.g., Manchester OWL Syntax) and the graphical notations.

2.2 Proposed Approach

We propose to extend graphical ontology diagrams with contextual on-demand verbalizations of OWL axioms related to the selected diagram elements, with the goal to help users to better understand their ontologies and to learn the graphical notations based on their own and/or real-world examples.

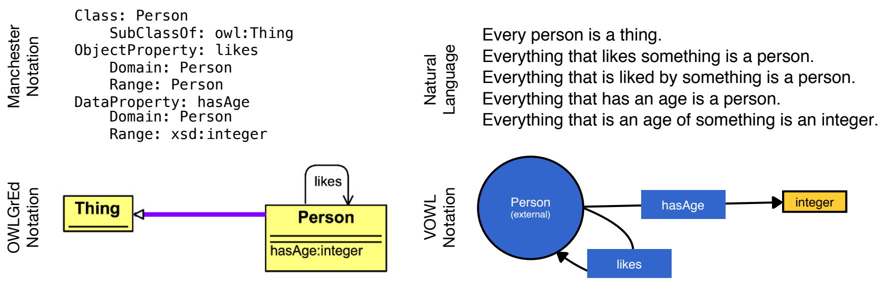

Fig. 1. A simplified ontology fragment alternatively represented by using Manchester OWL Syntax, VOWL and OWLGrEd, and an explanation in a controlled natural language

The contextual verbalization of ontology diagrams relies on the assumption that every diagram element represents a set of ontology axioms, i.e., the ontology axioms are generally presented locally in the diagram, although possibly a single ontology axiom can be related to several elements of the diagram.

The same verbalization can be applied to all the different OWL visual notations, i.e., we do not have to design a new verbalization (explanation) grammar for each new visual notation, because they all are mapped to the same underlying OWL axioms. Thus, the OWL visualizers can reuse the same OWL verbalizers to provide contextual explanations of any graphical OWL notation.

By reusing ontology verbalizers, existing ontology visualization systems can be easily extended with a verbalization service. Figure 2 illustrates how the proposed approach might work in practice:

1. *Visualizer* is the existing visualization component that transforms an OWL ontology into its graphical representation.
2. The system is extended by a *User Selection* mechanism that allows users to select the graphical element that they want to verbalize.
3. *Collector* gathers a subset of the ontology axioms that correspond to the selected graphical element.
4. The relevant axioms are passed to *Verbalizer* that produces CNL statements – a textual explanation that is shown to the user.

The actual implementation would depend on the components used and on how the output of the verbalization component can be integrated into the resulting visualization.

With the proposed approach, when domain experts encounter the example ontology in Figure 1, they would not have to guess what the elements of this graphical notation mean. Instead, they can just ask the system to explain the notation using the ontology that they are exploring. When the user clicks on the edge *likes* in Figure 1 (in either visual notation), the system would show

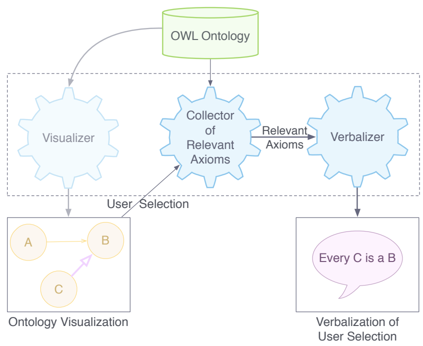

Fig. 2. Architecture of a contextual ontology verbalizer

the verbalization that unambiguously explains the complete meaning of this graphical element:

> *Everything that likes something is a person. Everything that is liked by something is a person.*

By applying the proposed approach and by using natural language to interactively explain what the graphical notation means, developers of graphical OWL editors and viewers can enable users (domain experts in particular) to avoid misinterpretations of ontology elements and their underlying axioms, resulting in a better understanding of both the ontology and the notation.

The verbalization can help users even in relatively simple cases, such as object property declarations where user's intuitive understanding of the domain and range of the property might not match what is asserted in the ontology. The verbalization of OWL axioms makes this information explicit while not requiring users to be ontology experts. The value of contextual ontology verbalization is even more apparent for elements whose semantics might be somewhat tricky even for more experienced users (e.g., *some*, *only* and cardinality constraints on properties, or OWLGrEd generalization forks with *disjoint* and *complete* constraints).

The verbalization of ontology axioms has been shown to be helpful in teaching OWL to newcomers both in practical experience reports [16] as well as in statistical evaluations [9] and thus it would be a valuable addition to ontology visualizations.

3 Proof of Concept: Interactive Verbalizations in OWLGrEd Visualizations

We illustrate the approach by extending OWLGrEd ontology visualizations (i.e. visualizations in the OWLGrEd notation used by the OWLGrEd ontology editor and its online ontology visualization tool) with on-demand contex-

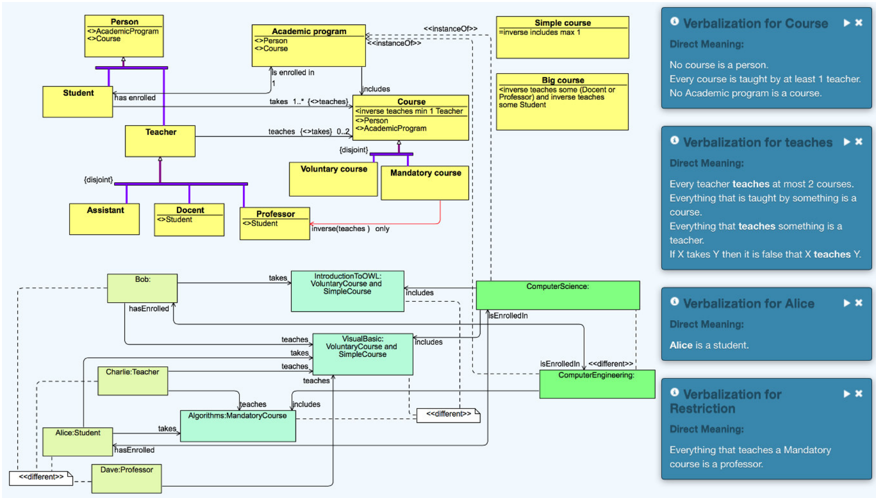

Fig. 3. The example ontology in the OWLGrEd notation with CNL verbalizations (explanations) of selected diagram elements.

tual verbalizations of the underlying OWL axioms using Attempto Controlled English (ACE) [5].

The interactive ontology verbalization layer allows users to inspect a particular element of the presented ontology diagram and to receive a verbal explanation of the ontology axioms that are related to this ontology element. By clicking a mouse pointer on an element, a pop-up widget is displayed, containing a CNL verbalization of the corresponding axioms in Attempto Controlled English. By default, the OWLGrEd visualizer minimizes the number of verbalization widgets shown simultaneously by hiding them after a certain timeout. For users to simultaneously see the verbalizations for multiple graphical elements, there is an option to "freeze" the widgets and prevent them from disappearing.

3.1 Ontology Verbalization Example

A demonstration of our approach, based on an example mini-university ontology, is available online[5]. Figure 3 shows a screenshot of the OWLGrEd visualization of this ontology containing a number of verbalizations.

These verbalizations describe the ontology elements that represent the class *Course*, the object property *teaches*, the individual *Alice* and the restriction on the class *MandatoryCourse*. Verbalizations are implicitly linked to the corresponding elements using the element labels. While it might be less convenient to identify the implicit links in a static image, the interactive nature of the combined ontology visualization and verbalization tool makes it easier for users to keep track of relations between diagram elements and their verbalizations.

[5] `http://owlgred.lumii.lv/cnl-demo`

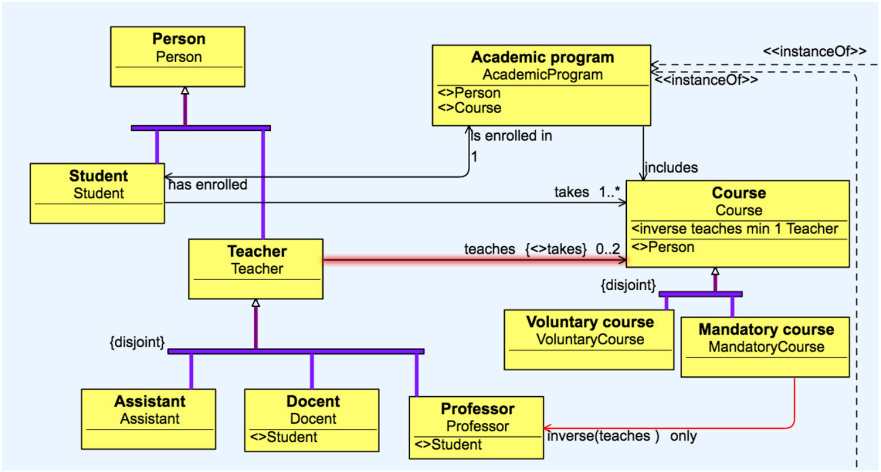

Fig. 4. A fragment of the ontology visualization example showing the *teaches* object property (highlighted).

To illustrate the verbalization functionality, let us look at the object property *teaches*, represented in the diagram by an edge connecting the class *Teacher* to the class *Course* (Figure 4). It leads to the following ACE verbalization:

(V1) *Every teacher teaches at most 2 courses.*

(V2) *Everything that is taught by something is a course.*

(V3) *Everything that teaches something is a teacher.*

(V4) *If X takes Y then it is false that X teaches Y.*

Note that the specific OWL terms, like *disjoint*, *subclass* and *inverse*, are not used in the ACE statements. The same meaning is expressed implicitly – via paraphrasing – using more general common sense constructions and terms.

In this case, the edge represents not only the domain and range axioms of the property (verbalized stentences V2, V3) but also the cardinality of its range (V1) and the restriction that *teaches* is disjoint with *takes* (expressed by the if-then statement in V4).

Further information about combining interactive contextual CNL verbalization with OWLGrEd ontology visualizations is available at [10].

4 Discussion

This section discusses the use of contextual ontology verbalization, focusing on its applicability to various graphical notations, multilinguality and the potential limitations of the approach.

4.1 Applicability to Different Visual Notations

The proposed approach is applicable to any ontology visualization where graphical elements represent one or more OWL axioms. The value of using verbalization functionality is higher for more complex notations (e.g., OWLGrEd) where graphical elements may represent multiple axioms but even in simple graphical notations, where each graphical element corresponds to one axiom, users will need to know how to read the notation. Contextual ontology verbalization addresses this need by providing textual explanations of diagram elements and the underlying OWL axioms.

A more challenging case is notations where some OWL axioms are represented as spatial relations between the elements and are not directly represented by any graphical elements (e.g., Concept Diagrams represent *subclass-of* relations as shapes that are included in one another [19]). In order to represent these axioms in ontology verbalization they need to be "attached" to one or more graphical elements that these axioms refer to. As a result, they will be included in verbalizations of relevant graphical elements. In the case of Concept Diagrams, the *subclass-of* relation, which is represented by shape inclusion, would be verbalized as part of the subclass shape.

4.2 Lexical Information and Multilinguality

In order to generate lexically and grammatically well-formed sentences, additional lexical information may need to be provided, e.g., that the property *teaches* is verbalized using the past participle form "taught" in the passive voice (*inverse-of*) constructions or that the class *MandatoryCourse* is verbalized as a multi-word unit "mandatory course".

In terms of multilinguality, contextual verbalizations can be generated for multiple languages, assuming that the translation equivalents are available for the lexical labels of ontology symbols. This would allow domain experts to explore ontologies in their native language and avoid information getting lost in translation.

An appropriate and convenient means for implementing a multilingual OWL verbalization grammar is Grammatical Framework (GF) [15] which provides a reusable resource grammar library for about 30 languages, facilitating rapid implementation of multilingual CNLs[6]. Moreover, an ACE grammar library based on the GF resource grammar library is already available for about 10 languages [2]. This allows for using English-based entity names and the OWL subset of ACE as an interlingua, following the two-level OWL-to-CNL approach suggested in [6].

We have used this GF-based approach in order to provide an experimental support for lexicalization and verbalization in OWLGrEd tools for both English and Latvian, a highly inflected Baltic language.

[6] `http://www.grammaticalframework.org/`

4.3 Limitations

Verbalization techniques that are a part of the proposed approach have the same limitations as ontology verbalization in general. In particular, verbalization may require additional lexical information to generate grammatically well-formed sentences. To some degree, by employing good ontology design practices and naming conventions as well as by annotating ontology entities with lexical labels, this limitation can be overcome. Another issue is specific kinds of axioms that are difficult or verbose to express in natural language without using terms of the underlying formalism.

5 Related Work

To the best of our knowledge, there are no publications proposing combining OWL ontology visualizations with contextual CNL verbalizations but there has been a movement towards cooperation between both fields. In ontology visualizations, notations have been adding explicit labels to each graphical element that describes what kind of axiom it represents. For example, in VOWL a line representing a *subclass-of* relation is explicitly labeled with the text "Subclass of". This practice makes the notation more understandable to users as reported in the VOWL user study where a user stated that "there was no need to use the printed [notation reference] table as the VOWL visualization was very self-explanatory" [12]. However, such labeling of graphical elements is only useful in notations where each graphical element represents one axiom. In more complex visualizations where one graphical element represents multiple axioms there would be no place for all the labels corresponding to these axioms. For example, in the OWLGrEd notation, class boxes can represent not just class definitions but also *subclass-of* and *disjoint classes* assertions. In such cases, verbalizations provide understandable explanations. Moreover, in some notations (e.g., Concept Diagrams [19]) there might be no graphical elements at all for certain kinds of axioms, as it was mentioned in Section 4.1.

In the field of textual ontology verbalizations there has been some exploration of how to make verbalizations more convenient for users. One approach that has been tried is grouping verbalizations by entities. It produces a kind of a dictionary, where records are entities (class, property, individual), and every record contains verbalizations of axioms that refer to this entity. The resulting document is significantly larger than a plain, non-grouped verbalization because many axioms may refer to multiple entities and thus will be repeated in each entity. Nevertheless, the grouped presentation was preferred by users [22]. Our approach can be considered a generalization of this approach, where a dictionary is replaced by an ontology visualization that serves as a map of the ontology.

An ad-hoc combination of verbalization and visualization approaches could be achieved using existing ontology tools such as Protégé by using separate visualization and verbalization plugins (e.g., ProtégéVOWL[7] for visualization and

[7] `http://vowl.visualdataweb.org/protegevowl.html`

ACEView[8] for verbalization). However, this would not help in understanding the graphical notation because the two views are independent, and thus a user cannot know which verbalizations correspond to which graphical elements. Our approach employs closer integration of the two ontology representations and provides contextual verbalization of axioms that directly correspond to the selected graphical element, helping users in understanding the ontology and learning the graphical notation used. The usefulness of combining ontology visualization and verbalization is also demonstrated by a course for ontology engineering that uses a CNL-based Fluent Editor and employs OWLGrEd visualizations for illustrating the meaning of CNL assertions[9].

6 Conclusions

Mathematical formalisms used in ontology engineering are hard to understand for domain experts. Usually, graphical notations are suggested as a solution to this problem. However, the graphical notations, while preferred by domain experts, still have to be learned to be genuinely helpful in understanding. Until now the only way to learn these notations was by reading the documentation.

In this article, we propose to combine ontology visualizations and CNL verbalizations in order to solve the learning problem. Using this approach the domain expert can interactively select a graphical element and receive the explanation of what the element means. The explanation is generated by passing the corresponding axioms of the element through one of the existing verbalization services. The service returns natural language sentences explaining the OWL axioms that correspond to the selected element and thus explaining what it means.

CNL explanations can help domain experts to rapidly and independently learn and use the graphical notation from the beginning, without extensive training, thus making it easier for domain experts to participate in ontology engineering thus solving one of the problems that hinder the adoption of Semantic Web technologies in the mainstream industry.

Acknowledgments. This work has been supported by the Latvian State Research program NexIT project No.1 "Technologies of ontologies, semantic web and security", the ESF project 2013/0005/1DP/1.1.1.2.0/13/APIA/VIAA/049, and the University of Latvia Faculty of Computing project "Innovative information technologies".

[8] http://attempto.ifi.uzh.ch/aceview/

[9] http://www.slideshare.net/Cognitum/introduction-to-ontology-engineering-with-fluent-editor-2014/19

References

1. Bārzdiņš, J., Bārzdiņš, G., Čerāns, K., Liepiņš, R., Sproģis, A.: UML-style graphical notation and editor for OWL 2. In: Perspectives in Business Informatics Research, pp. 102–114. Springer (2010)
2. Camilleri, J., Fuchs, N., Kaljurand, K.: ACE grammar library. Tech. Rep. Project Deliverable D11.1, MOLTO project (2012)
3. Falco, R., Gangemi, A., Peroni, S., Shotton, D., Vitali, F.: Modelling OWL ontologies with Graffoo. In: The Semantic Web: ESWC 2014 Satellite Events: ESWC 2014 Satellite Events, Anissaras, Crete, Greece, May 25-29, 2014, Revised Selected Papers. pp. 320–325. Springer (2014)
4. Frederiks, P.J., Van der Weide, T.P.: Information modeling: The process and the required competencies of its participants. Data & Knowledge Engineering 58(1), 4–20 (2006)
5. Fuchs, N.E., Kaljurand, K., Kuhn, T.: Attempto Controlled English for knowledge representation. In: Reasoning Web, pp. 104–124. Springer (2008)
6. Gruzitis, N., Barzdins, G.: Towards a more natural multilingual controlled language interface to OWL. In: Proceedings of the 9th International Conference on Computational Semantics. pp. 1006–1013 (2011)
7. Horridge, M., Patel-Schneider, P.F.: OWL 2 Web Ontology Language Manchester Syntax. W3C Working Group Note (2009)
8. Howse, J., Stapleton, G., Taylor, K., Chapman, P.: Visualizing ontologies: A case study. In: The Semantic Web – ISWC 2011, pp. 257–272. Springer (2011)
9. Kuhn, T.: The understandability of OWL statements in controlled English. Semantic Web 4(1), 101–115 (2013)
10. Liepiņš, R., Bojārs, U., Grūzītis, N., Čerāns, K., Celms, E.: Towards self-explanatory ontology visualization with contextual verbalization. In: Databases and Information Systems: 12th International Baltic Conference, DB&IS 2016, Riga, Latvia, July 4-6, 2016, Proceedings. pp. 3–17. Springer (2016)
11. Liepins, R., Grasmanis, M., Bojars, U.: OWLGrEd ontology visualizer. In: ISWC Developers Workshop 2014. CEUR (2015)
12. Lohmann, S., Negru, S., Haag, F., Ertl, T.: VOWL 2: User-oriented visualization of ontologies. In: Knowledge Engineering and Knowledge Management, pp. 266–281. Springer (2014)
13. Ottensooser, A., Fekete, A., Reijers, H.A., Mendling, J., Menictas, C.: Making sense of business process descriptions: An experimental comparison of graphical and textual notations. Journal of Systems and Software 85(3), 596–606 (2012)
14. Power, R., Third, A.: Expressing OWL axioms by English sentences: dubious in theory, feasible in practice. In: Proceedings of the 23rd International Conference on Computational Linguistics: Posters. pp. 1006–1013. Association for Computational Linguistics (2010)
15. Ranta, A.: Grammatical Framework, a type-theoretical grammar formalism. Journal of Functional Programming 14(2), 145–189 (2004)
16. Rector, A., Drummond, N., Horridge, M., Rogers, J., Knublauch, H., Stevens, R., Wang, H., Wroe, C.: OWL pizzas: Practical experience of teaching OWL-DL: Common errors & common patterns. In: Engineering Knowledge in the Age of the Semantic Web, pp. 63–81. Springer (2004)
17. Sharafi, Z., Marchetto, A., Susi, A., Antoniol, G., Guéhéneuc, Y.G.: An empirical study on the efficiency of graphical vs. textual representations in requirements comprehension. In: 21st International Conference on Program Comprehension. pp. 33–42. IEEE (2013)

18. Siau, K.: Informational and computational equivalence in comparing information modeling methods. Journal of Database Management 15(1), 73–86 (2004)
19. Stapleton, G., Howse, J., Bonnington, A., Burton, J.: A vision for diagrammatic ontology engineering. In: Visualizations and User Interfaces for Knowledge Engineering and Linked Data Analytics. CEUR (2014)
20. Stevens, R., Malone, J., Williams, S., Power, R., Third, A.: Automating generation of textual class definitions from OWL to English. J. Biomedical Semantics 2(S-2), S5 (2011)
21. Warren, P., Mulholland, P., Collins, T., Motta, E.: The usability of Description Logics. In: The Semantic Web: Trends and Challenges, pp. 550–564. Springer (2014)
22. Williams, S., Third, A., Power, R.: Levels of organisation in ontology verbalisation. In: Proceedings of the 13th European Workshop on Natural Language Generation. pp. 158–163. Association for Computational Linguistics (2011)