On the capabilities and limitations of OWL regarding typecasting and ontology design pattern views

Adila A. Krisnadhi, Pascal Hitzler, and Krzysztof Janowicz

[1] Wright State University, OH, USA
[2] University of California, Santa Barbara, USA

Abstract. In ontology engineering, particularly when dealing with heterogeneous domains and their subfields, legacy data, various data models, existing standards, code lists, and so forth, there is a frequently recurring need to express certain types of axioms related to linking diverse representational choices. Some of these point to limitations of the Web Ontology Language (OWL), others require best practice guides for the community. Here, we introduce some of these linking axioms and point to open research problems of potential interest to both application developers and researchers working on logical foundations of OWL.

1 Introduction and Motivation

During our ontology engineering work with subject matter experts from a wide range of domains including the broader geo-sciences [13], geography, industrial ecology, the digital humanities, libraries and the publishing industry, particle physics, and so forth, we became aware of the frequently recurring need to express certain types of axioms related to linking diverse representational choices. Often, these axioms can be easily expressed using first-order predicate logic, but the description logic underlying the Web Ontology Language OWL [8] does not – or not obviously – enable us to express these axioms.

In this paper, we will motivate and describe the types of axioms we encountered, as well as the capabilities and limitations of OWL in representing them, to the extent we are aware of them. The goal of the paper is to introduce these limitations as open problems on which researchers interested in improving and extending OWL and its underlying logics could work. Some of the issues discussed herein were already alluded to in [17], however they were not discussed in the context of formal semantics of OWL and the underlying description logics-based formalisms.

While all the axioms we will describe are about expressing relationships between different representational choices, we will structure our discussion by separating them into two types of axioms. The first kind, discussed in Section 2 are concerned with *typecasting.* The second kind, discussed in Section 3 are concerned with *view* expansion and contraction in the context of Ontology Design

Patterns [7,13,18]. Both kinds are relevant to ontology modeling and in ontology alignment with complex mappings. In Section 4 we conclude with a list of research questions compiled from throughout the paper.

2 Typecasting in OWL

In this section we discuss three kinds of typecasting that we frequently encountered in our work on ontological modeling. Casting between types is the implicitly or explicitly process by with one (data) type is converted into another type, e.g., widening an *int* to a *long*. In object-oriented modeling, this includes accessing objects that instantiate certain types as objects of their common ancestor type. To give a simple example, a Point Of Interest (POI) class may define a method to return the spatial footprint of the place as geographic (point) coordinates. All classes that extend the POI class, say Restaurant and Hospital, can be queried for their footprint by iterating over a collection of POI.

Here, we use the term typecasting to refer to multiple representational choices to define classes and properties in the context of description logics and translations between those representational choices.

2.1 Typecasting Individual to Class and Back: Explicit Versus Implicit Typing of Instances

The first case is concerned with the representational choice between the explicit typing of individuals (via rdf:type) versus the identification of the type of an individual by reference to a classname, given as an individual. In other words, we do typecasting from a class to an individual and vice versa.

Problem Description Schematically, the two representational choices are depicted in Figure 1. Generally speaking, the explicit use of rdf:type seems to be more in the spirit of OWL, however, case 1b occurs naturally e.g., when referring to an external controlled vocabulary.

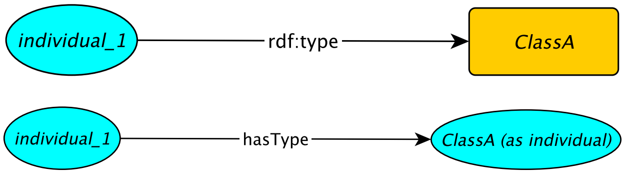

Fig. 1. Case 1a (top) and case 1b (bottom)

Consider, for example, the case of measurement types. A concrete measurement of a particular characteristic of a feature of interest, e.g., a lake, can be of type NitrateConcentration (which in turn is a Concentration measurement).

```
ex:measurement1  rdf:type  geo:NitrateConcentration .
```

We assume here that the namespace geo refers to an appropriate ontology that contains measurement types. At the same time, however, it may be appropriate for this ontology to incorporate an existing wide-spread controlled vocabulary for the identification of measurement types used in certain fields of the geosciences. Such controlled vocabularies often come in the form of code lists or may describe measurement types as individuals because one may want to give additional information about them, say

```
geo:NO3-_concentration  geo:potentialSource geo:urban_runoff .
```

With `geo:NO3-_concentration` being an individual, one would therefore appropriately identify the measurement type for `:measurement1` by specifying

```
ex:measurement1 geo:hasMeasurementType geo:NO3-_concentration .
```

Logical Aspects This typecasting case can be handled easily in OWL. Axiom (1) maps case 1a to 1b, while axiom (2) maps case 1b to 1a.

$$\text{ClassName} \sqsubseteq \exists \text{hasType}.\{\text{classname}\} \tag{1}$$

$$\exists \text{hasType}.\{\text{classname}\} \sqsubseteq \text{ClassName} \tag{2}$$

For the nitrate concentration examples, the mappings can be expressed by:

$$\texttt{geo:NitrateConcentration} \equiv \exists \texttt{geo:hasMeasurementType}.\{\texttt{geo:NO3\_concentration}\}.$$

It shall be noted that this case is closely related to punning between classes and individuals as it is allowed in OWL 2 DL. However, punning as such refers to the use of one identifier to denote both a class name and an individual name. As can be seen from our discussion, it is in fact possible to map, in OWL 2 DL, between the two perspectives.

2.2 Rolification: Typecasting from Classes to Properties

Rolification [11,19] refers to the typecasting of classes into properties. This is a key technique for representation of (Datalog) rules in OWL. There are of course rules that can be readily expressed using OWL axioms, e.g., guarded range restrictions such as $\text{Event}(x) \wedge \text{occursAt}(x, y) \rightarrow \text{Place}(y)$, which is equivalent to $\exists \text{occursAt}^{-}.\text{Event} \sqsubseteq \text{Place}$. However, this is not the case for other rules such as:

$$\text{Fix}(x) \wedge \text{hasLocation}(x, y) \wedge \text{hasSpatialFootprint}(z, y) \wedge \text{Port}(z) \rightarrow \text{atPort}(x, z)$$

The above rule can be expressed in OWL by *rolifying* the two class names, i.e. by introducing new roles (properties) R_{Fix} and R_{Port} and asserting three axioms

$$\text{Fix} \equiv \exists R_{\text{Fix}}.\text{Self}, \qquad \text{Port} \equiv \exists R_{\text{Port}}.\text{Self},$$
$$R_{\text{Fix}} \circ \text{hasLocation} \circ \text{hasSpatialFootprint} \circ R_{\text{Port}} \sqsubseteq \text{atPort}$$

Another perspective on the use of rolification is that it allows us to express *typed role chains* of the form

$$R_1 \circ \ldots \circ R_n \sqsubseteq R,$$

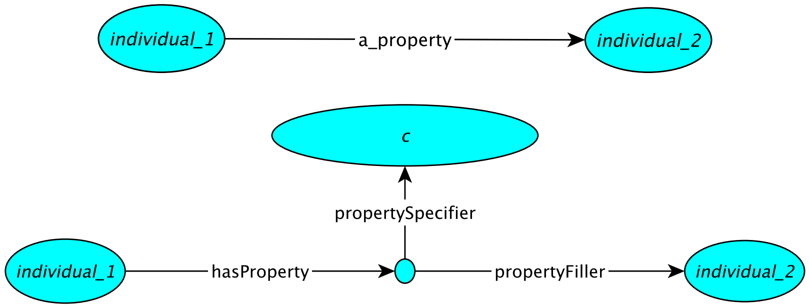

Fig. 2. Case 2a (top) and case 2b (bottom)

where each R_i is either a property name or a class name, the latter of which is to be typecast into a property using rolification.

While the rolification axiom $A \equiv \exists R_A.\text{Self}$ itself is of course expressible in OWL DL, its primary use cases, i.e., general conversion of rules, always involve property chains. OWL DL, however, imposes a so-called *regularity restriction* on the use of property chains [9], which may be violated by the introduced ones. The origin of the regularity restriction is that without it, reasoning over the logic would be undecidable. While this means that the restriction cannot be lifted in its entirety without rendering the logic undesirable, it would be helpful to soften it, i.e., to describe types of cases which violate regularity, but which retain decidability.

2.3 Reification: Typecasting Properties into Classes

The third case we want to discuss concerns the representational choice between using a new property name versus typing and reification.

Problem Description. Schematically, the two representational choices are depicted in Figure 2, but this requires further explanation. Case 2a simply indicates the use of a property, e.g., returning to the scientific cruises and ports example, a statement like

```
geo:cruise123   geo:hasLeadScientist   geo:PeterWiebe .
```

while case 2b would use a reified representation, using a blank node, e.g., using RDF Turtle syntax [2], this is in the form of

```
geo:cruise123  geo:hasParticipant   [
               geo:playsRole  geo:leadScientist   ;
               geo:isPlayedBy  geo:PeterWiebe ] .
```

The latter type of representation is in fact very common, e.g. as part of the so-called Agent Role ontology design pattern,[3] and has in some form even be

---

[3] `http://ontologydesignpatterns.org/wiki/Submissions:AgentRole`

adopted by schema.org under the term *role*.[4] The advantage of the second, more verbose representation is that additional information can be added to the blank node, e.g., the funding agency, affiliation, and so forth. Note that the typecasting discussed in Section 2.1 can also be applied to the individual acting as the property specifier in case 2b above.

Logical Aspects. Mapping from case 2b to case 2a is possible in OWL, with a caveat, using a rule like

$$\begin{aligned}\text{hasProperty}(x,y) \land \text{propertySpecifier}(y,c) \land \text{propertyFiller}(y,z)\\ \rightarrow \text{a\_property}(x,z),\end{aligned}$$

which can be expressed in OWL using the two axioms

$$\begin{aligned}\exists\text{propertySpecifier}.\{c\} &\sqsubseteq \exists\text{propertySpecifier\_c}.\text{Self}\\ \text{hasProperty} \circ \text{propertySpecifier\_a} \circ \text{propertyFiller} &\sqsubseteq \text{a\_property}\end{aligned}$$

– note that this translation uses rolification of the complex class

$$\exists\text{propertySpecifier}.\{c\},$$

as discussed above in Section 2.2, for typecasting classes into properties.

The mentioned caveat is thus the same as the one mentioned in Section 2.2 above, namely that OWL 2 DL imposes restrictions on the use of property chains, and depending on the other axioms in the ontology the typecasting axioms may lead to a violation of these so-called *regularity restrictions*, which effectively would render the resulting ontology to be in OWL 2 Full but not in OWL 2 DL. However, approximate work-arounds are possible, using so-called *nominal schemas* [11,14], which we will discuss further below.

Typecasting in the other direction, from case 2a to case 2b, cannot be handled in OWL, but can be handled using rules with existential head – well-known in database as tuple-generating dependencies (TGDs) [3] – e.g., as follows.

$$\begin{aligned}\text{a\_property}(x,z) \rightarrow \exists y(\text{hasProperty}(x,y) \land \text{propertySpecifier}(y,a)\\ \land \text{propertyFiller}(y,z))\end{aligned}$$

3 Ontology Design Pattern Views Contraction and Expansion

By a *view* we mean a set of shortcuts through an ontology or an ontology design pattern. To illustrate the concept, which is discussed also in [12,18], we adapt an example from the GeoLink oceanography ontology [13].

Refering to Figure 3, the red arrows indicate shortcuts, and we will discuss the case of the isTraversedBy shortcut. Of course the picture is only a visualization

---

[4] `https://schema.org/Role`

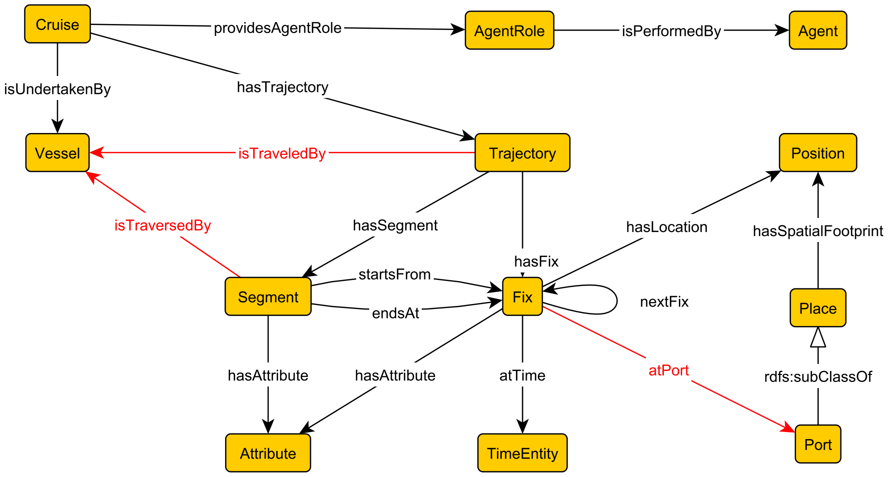

Fig. 3. Part of the GeoLink oceanography ontology to illustrate views.

of a part of the ontology, which consists of a set of OWL axioms which we do not list here.

In the fabric of the ontology, the isTraversedBy shortcut is in fact redundant, i.e. it can be inferred using the rule

$$\begin{aligned}\text{Vessel}(x) \wedge \text{isUndertakenBy}^-(x,y) \wedge \text{Cruise}(y) \wedge \\ \wedge\ \text{hasTrajectory}(y,z) \wedge \text{Trajectory}(z) \wedge \text{hasSegment}(z,w) \wedge \\ \wedge\ \text{Segment}(w) \to \text{isTraversedBy}(w,x).\end{aligned} \tag{3}$$

Since the application of the rule results in a simpler representation of the relationship between a trajectory segment and the vessel traversing it, we refer to this type of rule also as a *contraction*.

The reverse of a contraction is an *expansion*. In our experience, this case occurs when, e.g., a data provider may have only information about trajectories (and their segments) which oceanographic vessels have taken. In order to populate the ontology with this data, it is required to *expand* the data by inserting an additional individual (or a blank node) as the cruise connecting the trajectory and the vessel.

3.1 Contraction

A generic depiction of the view idea is presented in Figure 4. The grey ellipse shall indicate a labeled graph which in turn can be represented as a conjunction of unary and binary predicates involving ClassA and ClassB, like

$$\text{ClassA}(x) \wedge \text{ClassB}(y) \wedge C_1(x_1) \wedge \cdots \wedge C_n(x_n) \wedge R_1(y_1,y_2) \wedge \cdots \wedge R_k(y_k,y_{k+1}),$$

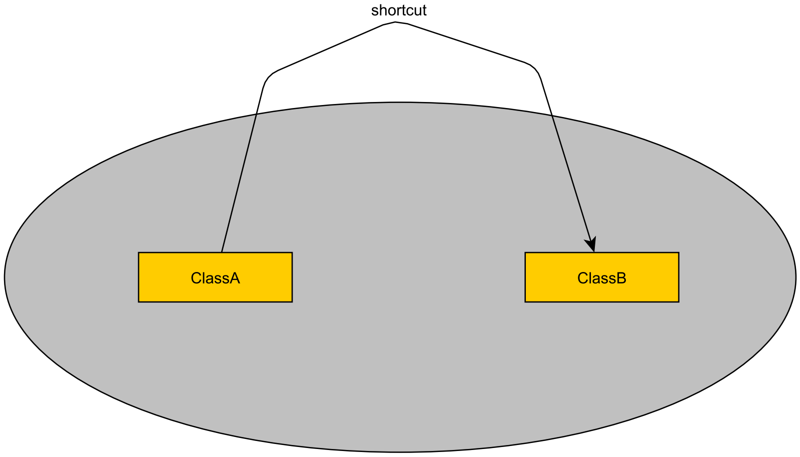

Fig. 4. Depiction of a generic shortcut.

where x, y, the x_i and the y_j are any variables.

Contraction (i.e., a shortcut between the classes ClassA and ClassB) can then be expressed using the rule

$$\text{ClassA}(x) \wedge \text{ClassB}(y) \wedge C_1(x_1) \wedge \cdots \wedge C_n(x_n) \wedge R_1(y_1, y_2) \wedge \cdots \wedge R_k(y_k, y_{k+1}) \\ \rightarrow \text{shortcut}(x, y). \tag{4}$$

Note that the simpler typecasting case discussed in Section 2.3 can in fact be understood as a very simple case of contraction and expansion.

The rule expressing a shortcut (i.e., view contraction) cannot in general be represented in OWL, and this is well-known. In particular, if the graph representing the rule body is cyclic, this is not possible in many cases. Discussing this in detail is out of scope for this paper, but a detailed account of this can e.g. be found in [11,14].

However, let us work the earlier example from rule (3), which is not cyclic. In this case we can convert the rule into OWL using rolification, which results in the following set of axioms.

$$\begin{aligned}
\text{Vessel} &\equiv \exists R_{\text{Vessel}}.\text{Self} \\
\text{Cruise} &\equiv \exists R_{\text{Cruise}}.\text{Self} \\
\text{Trajectory} &\equiv \exists R_{\text{Trajectory}}.\text{Self} \\
\text{Segment} &\equiv \exists R_{\text{Segment}}.\text{Self} \\
R_{\text{Segment}} &\circ \text{hasSegment}^- \circ R_{\text{Trajectory}} \circ \\
&\circ \text{hasTrajectory}^- \circ R_{\text{Cruise}} \circ \text{isUndertakenBy} \sqsubseteq \text{isTraversedBy}
\end{aligned}$$

The problem is again, of course, that the introduction of additional role chains may render the ontology to be outside OWL DL due to possible violations of regularity restrictions.

The cases where rules are cyclic pose particular challenges. We illustrate this by an example taken from [11]. The rule defines a shortcut indicating a review assignment with a conflict of interest.

$$\begin{aligned}&\text{hasReviewAssignment}(v,x) \wedge \text{hasAuthor}(x,y) \wedge \text{atVenue} \wedge\\ &\wedge \text{hasSubmittedPaper}(v,u) \wedge \text{hasAuthor}(u,y) \wedge \text{atVenue}(u,z)\\ &\qquad\qquad\qquad\qquad\to \text{hasConflictingAssignedPaper}(v,x)\end{aligned}$$

Approximate (sound but incomplete) translations of such cyclic rules into OWL are possible using DL-safe rules [10,16]. Better approximations (i.e., with not as much loss in terms of logical consequences) are possible using so-called *nominal schemas* [11,14]. While in the meantime some results have been obtained regarding efficient reasoning with nominal schemas [6,20], the topic does still require in depth exploration to obtain sufficient coverage for modeling purposes.

Assuming familiarity with rule to OWL conversion techniques as discussed e.g. in [11], we identify several research questions which address such conversion issues. Some of them have in fact already been exposed by our earlier discussions.

(1) Translation of rules usually requires rolification and the use of role chains, i.e. softening regularity restrictions would be extremely helpful.
(2) Approximate translation of rules (approximate in order to avoid regularity issues) currently requires the use of nominal schemas, for which efficient reasoning algorithms, as well as suitable modeling and reasoning tools, require further investigation.
(3) Translated rules often fall into the OWL EL fragment with additional use of inverse roles. While OWL EL requires regularity, the regularity requirement is not required for decidability of the logic. However, in the presence of inverse roles, together with a non-regular set of role chains, the logic becomes undecidable [1]. Softening the regularity requirement for OWL EL with addtional inverse roles would make it possible to translate more shortcut rules.
(4) Likewise, OWL EL (with the regularity restriction) together with inverse roles is no longer tractable. Research into conditions under which tractability is retained would be helpful in practice – see e.g. for [5] for some work related to this issue.

Another issue arising out of shortcuts is if ClassB, in our generic example, is actually a datatype, i.e., the infered role 'shortcut' shall be a datatype property. Structurally, representation of the corresponding rule should follow the same method, however the resulting OWL axioms will then usually involve a role chain with a datatype property as the final, right-most role. However, the OWL standard currently does not allow this. We conjecture that allowing this would probably be a minor extension of the standard, but this still requires looking at.

3.2 Expansion

Expansion is the reverse of contraction, i.e. expanding from a shortcut into the graph, as in our generic example. It can be understood as a generalized version of the direction from Section 2.3 where a blank node is introduced, i.e. as a type of role introduction instead of using an elementary property.

Of course, simply reversing the implication arrow in rule (4) is insufficient, as quantification of the variables needs to be addressed. The appropriate axiomatization, in fact, is the following.

$$\begin{aligned}\text{shortcut}(x,y) \rightarrow \text{ClassA}(x) \land \text{ClassB}(y) \land \exists x_1 \ldots \exists x_n \exists y_1 \ldots \exists y_n (C_1(x_1) \land \ldots \\ \cdots \land C_n(x_n) \land R_1(y_1, y_2) \land \cdots \land R_k(y_k, y_{k+1}))\end{aligned}$$

Similar to the case above in Section 2.3, existential rules appear to be a suitable paradigm, in principle. However the potentially rather complex rule heads deserve considerable investigation, in particular if it is to be integrated with ontology reasoning.

A specific case which may also deserve studying is when the rule head may be translatable into a right-hand-side role chain, i.e. an axiom of the form

$$R \sqsubseteq R_1 \circ \cdots \circ R_n,$$

possible after some rolification. Right-hand-side role chains have been studied in the literature and in the general case they lead to undecidability, particularly when left-hand side role chains are present. Decidability by generalizing regularity restriction where shown by Mosurovic, et al [15]. On the other hand, the rule above can also be categorized into guarded TGDs [4] for which query answering is decidable. Note, however, that adding existential rules to OWL in general may cause the violation of guardedness condition, hence may not guarantee decidability.

4 Conclusions

We have seen that modeling issues arising in practice give rise to logical axioms which are currently not expressible within the OWL DL standard, and this prompts research questions which may ultimately lead to a suitable coverage in a later version of the standard. To provide an overview, we list the reseach questions raised by our discussion.

1. Relaxing RBox regularity constraints to make use of rolification easier, for several of the aspects mentioned above.
2. Relaxing RBox regularity constraints in the specific case of OWL EL with additional inverse roles would allow for the expression of more view contractions. Aspects to be considered would be both, decidability and tractability.
3. Develop more efficient reasoning algorithms and implementations for nominal schemas, as they are one way to circumvent the regularity issues arising from rolification.

4. Investigate reasoning aspects regarding role chains which end in datatype literals, including the issue of right-hand-side role chains.
5. Investigate right-hand-side role chains as a possible extension of OWL DL.
6. Investigate the integration of existential rules with OWL DL, in particular for complex rule heads.

Acknowledgements. This work was supported by the National Science Foundation under award 1017225 *III: Small: TROn – Tractable Reasoning with Ontologies* and award 1440202 *EarthCube Building Blocks: Collaborative Proposal: GeoLink – Leveraging Semantics and Linked Data for Data Sharing and Discovery in the Geosciences.*

References

1. Baader, F., Brandt, S., Lutz, C.: Pushing the EL envelope. In: Kaelbling, L.P., Saffiotti, A. (eds.) IJCAI-05, Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence, Edinburgh, Scotland, UK, July 30-August 5, 2005. pp. 364–369 (2005)
2. Beckett, D., Berners-Lee, T., Prud'hommeaux, E., Carothers, G.: RDF 1.1. Turtle – Terse RDF Triple Language. W3C Recommendation (25 February 2014), available at http://www.w3.org/TR/turtle/
3. Beeri, C., Vardi, M.Y.: A proof procedure for data dependencies. Journal of the ACM 31(4), 718–741 (1984)
4. Calì, A., Gottlob, G., Kifer, M.: Taming the infinite chase: Query answering under expressive relational constraints. J. Artif. Intell. Res. (JAIR) 48, 115–174 (2013), `http://dx.doi.org/10.1613/jair.3873`
5. Carral, D., Feier, C., Grau, B.C., Hitzler, P., Horrocks, I.: *EL*-ifying ontologies. In: Demri, S., Kapur, D., Weidenbach, C. (eds.) Automated Reasoning – 7th International Joint Conference, IJCAR 2014, Held as Part of the Vienna Summer of Logic, VSL 2014, Vienna, Austria, July 19-22, 2014. Proceedings. Lecture Notes in Computer Science, vol. 8562, pp. 464–479. Springer (2014)
6. Carral, D., Wang, C., Hitzler, P.: Towards an efficient algorithm to reason over description logics extended with nominal schemas. In: Faber, W., Lembo, D. (eds.) Web Reasoning and Rule Systems – 7th International Conference, RR 2013, Mannheim, Germany, July 27-29, 2013. Proceedings. Lecture Notes in Computer Science, vol. 7994, pp. 65–79. Springer (2013)
7. Gangemi, A.: Ontology design patterns for semantic web content. In: Gil, Y., Motta, E., Benjamins, V.R., Musen, M.A. (eds.) The Semantic Web – ISWC 2005, 4th International Semantic Web Conference, ISWC 2005, Galway, Ireland, November 6-10, 2005, Proceedings. Lecture Notes in Computer Science, vol. 3729, pp. 262–276. Springer (2005)
8. Hitzler, P., Krötzsch, M., Parsia, B., Patel-Schneider, P.F., Rudolph, S. (eds.): OWL 2 Web Ontology Language: Primer. W3C Proposed Recommendation 22 September 2009 (2009), available from http://www.w3.org/TR/owl2-primer/
9. Hitzler, P., Krötzsch, M., Rudolph, S.: Foundations of Semantic Web Technologies. CRC Press/Chapman & Hall (2010)
10. Hitzler, P., Parsia, B.: Ontologies and rules. In: Staab, S., Studer, R. (eds.) Handbook on Ontologies, pp. 111–132. Springer (2009)

11. Krisnadhi, A., Maier, F., Hitzler, P.: OWL and rules. In: Polleres, A., d'Amato, C., Arenas, M., Handschuh, S., Kroner, P., Ossowski, S., Patel-Schneider, P.F. (eds.) Reasoning Web. Semantic Technologies for the Web of Data – 7th International Summer School 2011, Galway, Ireland, August 23-27, 2011, Tutorial Lectures. Lecture Notes in Computer Science, vol. 6848, pp. 382–415. Springer (2011)
12. Krisnadhi, A.A., Hu, Y., Janowicz, K., Hitzler, P., Arko, R., Carbotte, S., Chandler, C., Cheatham, M., Fils, D., Finin, T., Ji, P., Jones, M., Karima, N., Lehnert, K., Mickle, A., Narock, T., O'Brien, M., Raymond, L., Shepherd, A., Schildhauer, M., Wiebe, P.: The GeoLink framework for pattern-based Linked Data integration. In: Proceedings ISWC2015 poster session (2015), to appear
13. Krisnadhi, A.A., Hu, Y., Janowicz, K., Hitzler, P., Arko, R., Carbotte, S., Chandler, C., Cheatham, M., Fils, D., Finin, T., Ji, P., Jones, M., Karima, N., Mickle, A., Narock, T., O'Brien, M., Raymond, L., Shepherd, A., Schildhauer, M., Wiebe, P.: The GeoLink modular oceanography ontology. In: Proceedings ISWC2015. Lecture Notes in Computer Science, Springer (2015), to appear
14. Krötzsch, M., Maier, F., Krisnadhi, A., Hitzler, P.: A better uncle for OWL: nominal schemas for integrating rules and ontologies. In: Srinivasan, S., Ramamritham, K., Kumar, A., Ravindra, M.P., Bertino, E., Kumar, R. (eds.) Proceedings of the 20th International Conference on World Wide Web, WWW 2011, Hyderabad, India, March 28 – April 1, 2011. pp. 645–654. ACM (2011)
15. Mosurovic, N., Krdzavac, H., Graves, M., Zakharyaschev, M.: A decidable extension of SROIQ with complex role chains and unions. Journal of Artificial Intelligence Research 47, 809–851 (2013)
16. Motik, B., Sattler, U., Studer, R.: Query answering for OWL-DL with rules. Journal of Web Semantics 3(1), 41–60 (2005)
17. Noy, N. (ed.): Representing Classes As Property Values on the Semantic Web. W3C Working Group Note (5 April 2005), `http://www.w3.org/TR/swbp-classes-as-values/`
18. Rodrguez-Doncel, V., Krisnadhi, A.A., Hitzler, P., Cheatham, M., Karima, N., Amini, R.: Pattern-based Linked Data publication: The Linked Chess Dataset case. In: Proceedings COLD2015 (2015), to appear
19. Rudolph, S., Krötzsch, M., Hitzler, P.: All elephants are bigger than all mice. In: Baader, F., Lutz, C., Motik, B. (eds.) Proceedings of the 21st International Workshop on Description Logics (DL2008), Dresden, Germany, May 13-16, 2008. CEUR Workshop Proceedings, vol. 353. CEUR-WS.org (2008)
20. Steigmiller, A., Glimm, B., Liebig, T.: Reasoning with nominal schemas through absorption. Journal of Automated Reasoning 53(4), 351–405 (2014)