Selecting Ontology Entailments for Presentation to Users

Artemis Parvizi, Chris Mellish, Kees van Deemter, Yuan Ren and Jeff Z. Pan
Department of Computing Science, University of Aberdeen, Aberdeen, U.K.

Keywords: Semantic Web, Entailment Selection, System Feedback, Human Computer Interaction.

Abstract: Presenting entailments of axioms in a formal ontology is a non-trivial task. This position paper argues that the problem of selecting entailments for presentation to users is not adequately acknowledged or addressed in the literature or in implemented systems. We analyse the problem and consider some different approaches that can help to address the problem.

1 CONTENT SELECTION WITH OWL

Many tools aim to assist users in creating or modifying ontologies. Most of these tools will at some point show users certain *entailments* of the axioms they have defined; for instance, the new entailments that follow given the addition of a new axiom. On the face of it, this might seem a simple matter: a reasoner is applied to the defined axioms and generates a set of entailments (perhaps filtering out those that were already entailed before the last axiom was added) and these are then presented to the user. This simple situation could work if the user was developing an ontology using only RDFS. But it is well known that, with any sufficiently expressive logic and when there can be arbitrary background knowledge, the number of entailments of a set of axioms is *infinite*. Unfortunately most families of OWL fall into this class of expressive logics.

For instance, in a variant of the famous pizza ontology, an ontology author might be interested in those concepts that subsume *PineappleHamPizza* (here abbreviated to *PH*) after addition of an axiom defining this concept. If the user is unlucky, the list of entailments might be presented to her as a list whose first few items are as follows (mercifully limited to entailments of the form $PH \sqsubseteq X$):

$$PH \sqsubseteq (\neg VegetarianPizza \sqcap NamedPizza) \quad (1)$$
$$PH \sqsubseteq (\forall hasBase.ThinAndCrispyBase) \quad (2)$$
$$PH \sqsubseteq (\forall hasBase.(\exists isIngredientOf.CheeseyPizza \sqcap SpicyPizza)) \quad (3)$$
$$PH \sqsubseteq (\exists hasBase.(\exists isIngredientOf.SpicyPizzaEquiv) \sqcap ThinAndCrispyBase) \quad (4)$$
$$PH \sqsubseteq (\exists hasIngredient.(\forall isToppingOf.SpicyPizza) \sqcap CheeseTopping)) \quad (5)$$
$$PH \sqsubseteq (\exists hasIngredient.(\forall hasSpiciness.Hot)) \quad (6)$$
$$PH \sqsubseteq (\exists hasTopping.(\exists isToppingOf.NamedPizza \sqcap SpicyPizzaEquiv)) \quad (7)$$
$$PH \sqsubseteq (\leq 4.hasTopping) \quad (8)$$
$$PH \sqsubseteq (Pizza \sqcap NamedPizza) \quad (9)$$
$$PH \sqsubseteq (Pizza \sqcap Pizza \sqcap NamedPizza) \quad (10)$$
$$PH \sqsubseteq (\neg CheeseTopping) \quad (11)$$
$$PH \sqsubseteq (\leq 245.hasTopping) \quad (12)$$

The set of all the entailments of an ontology can be infinite for two different reasons, giving rise to two different selection problems:

1. There are infinitely many logically equivalent versions of any given axiom, especially if one takes into account the background knowledge of the ontology at hand. For example, in the above, $Pizza \sqcap NamedPizza$ (axiom 9) is equivalent with $Pizza \sqcap Pizza \sqcap NamedPizza$ (axiom 10), therefore any axiom that uses one can use the other instead. Which one(s) to present?
2. In the worst case, there are infinitely many *logically distinct* entailments, which state infinitely many different things about the world. For instance, if $PH \sqsubseteq (\exists hasBase.X)$ then also $PH \sqsubseteq (\exists HasBase.(X \sqcup Y))$ for every possible class Y. Which of these will interest the user?

The first of these problems is known in the Computational Linguistics literature as the "problem of logical equivalence" (Appelt, 1987).

The fact that the number of entailments is infinite means that an interface supporting ontology development in OWL has to make a *selection* from the infi-

Parvizi A., Mellish C., van Deemter K., Ren Y. and Z. Pan J..
Selecting Ontology Entailments for Presentation to Users.
DOI: 10.5220/0005136203820387
In *Proceedings of the International Conference on Knowledge Engineering and Ontology Development* (KEOD-2014), pages 382-387
ISBN: 978-989-758-049-9

nite set[1] More precisely, given that no reasoner will deliver the infinite set, the interface needs to select by making focussed requests to the reasoner and then perhaps selecting from the results. The challenge, in our view, is to select those axioms that the user is most interested in, and which give the user the best understanding of the total set of entailments; additionally, the selected set should be presented in a way that helps the user understand the structure of the set.

How do existing interfaces address this problem? The most widely used ontology authoring environment, Protégé [2] provides a box where the user can check "Show Inferences". The interface then makes certain entailments visible (or findable), and others of course not, without further comment. As another example, the ROO environment (Denaux et al., 2012) provides valuable feedback to a user when they add a new axiom. One form of its output can be something like "This input implies 6 new relevant facts ...", but without an explanation of what notion of "relevance" is at work. With both of these systems, a technically sophisticated person can find out and understand the criteria for selection, but the criteria are not justified and it is unclear whether they match the needs of the user. Worse still, users may end up misunderstanding the nature of logic by getting the impression that entailments are necessarily finite and limited in scope.

2 POSSIBLE APPROACHES TO ENTAILMENT SELECTION

A number of principles can be used to select entailments, but it is surprisingly difficult to find a clear-cut way of categorising them. In the following, we divide these principles into four categories, based on the type of information they use. These categories can of course be combined. The most frequently used methods use just syntax.

2.1 Approaches using Syntax

Here axioms are selected based on the explicit structure used to state them. For instance, a syntactic approach might select axioms of the form $A \sqsubseteq B$ where A is a named concept. The example in section 1 is of this kind, with PH in the position of A.

Traditional ontology authoring environments, such as Protégé , ClickOnA (Wolters and Nolle, 2013) and Swoop (Kalyanpur et al., 2006), allow the user to select different aspects of the ontology to be viewed by offering different options in the user interface. In terms of axiom presentation, the user is selecting the material based on the syntactic form of the relevant axioms. A common facility is to allow the user to browse around the axioms of the form $X \sqsubseteq Y$, with X and Y being named classes, by displaying a hierarchy of classes. These facilities rely on standard reasoner functionalities, provided for instance through the OWL API (Horridge and Bechhofer, 2011), such as to get named superclasses and subclasses of named classes, named domains and ranges of properties, named equivalent classes, the object property values of a named instance and indeed to test an arbitrary specific axiom for being entailed.

Surprisingly Controlled Natural Language-based (CNL) systems, whose main objective is to simplify the process of exploration and authoring OWL ontologies, do not seem to address entailment selection. For example, ACE View (Fuchs et al., 2008) (a Protégé extension) presents an "entailments view" showing class assertions, property assertions and subclass axioms where the involved individuals, properties, and classes are always *named*. OntoVerbal (Liang et al., 2011) (a Protégé plugin) concentrates on axiom realisation, grouping and aggregation, and does not present any approach to tackle axiom selection. The RoundTrip Ontology Authoring (ROA) (Davis et al., 2008) environment only presents axioms that have been explicitly defined, selecting in turn top level classes, subclasses, instances, class properties and their respective domain and ranges, and instance properties.

In the Rabbit to OWL Ontology authoring (ROO) editor (Denaux et al., 2012), the authors acknowledge the issue of infinitely many entailments and decide to use a subset that contains the finite *set of relevant implications*. Relevant axioms are those of the form $A \sqsubseteq B$, $\top \sqsubseteq B$, $A \sqsubseteq \bot$ and $A(a)$ such that A and B are concept expressions that appear in some axiom in $O \cup \{a\}$ and a is a named individual in $O \cup \{a\}$. This is probably the widest set of entailments presented by an ontology authoring environment.

For reducing the set of generated subsumers, Mellish and Pan (Mellish and Pan, 2008) restrict the entailments to be in the sublanguage $\mathcal{ALEN}$ where disjunctions are not allowed and negations are restricted to only atomic concepts. To make this subsumers set finite, the authors restrict the generated axioms to be in a specific normal form and limit the syntactic complexity of this.

[1] A similar problem arises (e.g. with a less expressive logic) when the number of entailments is large but finite.

[2] http://protege.stanford.edu/

2.2 Approaches using Logic

A logic-based approach takes into account how axioms behave logically, i.e. makes appeal to notions like entailment and logical derivation in selecting axioms. In its purest form, such an approach can only make choices between axioms which are logically distinct. For instance, one principle might be not to select any axiom that is a tautology. As another example, a logical approach might select axioms of the form $X \sqsubseteq Y$ which are *most specific* for Y in the ontology O, in which there is no Z such that $O \models X \sqsubseteq Z$, $O \models Z \sqsubseteq Y$ and $O \not\models Y \sqsubseteq Z$; for instance, if the ontology has Boy $\sqsubseteq$ Man $\sqsubseteq$ Person then Boy $\sqsubseteq$ Person should not be presented but Boy $\sqsubseteq$ Man may be.

Mellish and Pan (Mellish and Pan, 2008) adopt something like the above principle of specificity. They also state various logical principles based on the Gricean maxims (Grice, 1970) of cooperative conversation for further reducing the set of subsumers being generated for a class.

A further possibility is to use logic in a different way, by letting the derivation of entailments play a role. For example, the user may be permitted to ask to only see entailments equivalent to axioms whose derivation hinges on the transitivity of a particular role; entailments whose derivation does not use axioms containing the universal quantifier; or entailments whose derivation requires more than n reasoning steps. Further, it might make sense to prefer entailments that have longer derivations (because they are less trivial) or entailments that have shorter derivations (because they are more understandable)[3].

2.3 Approaches using Discourse Structure

When entailments are being selected for presentation as part of a conversation, e.g. in the context of an ontology authoring environment, there may be structure in the discourse that can be exploited to help determine what entailments might be most relevant. A discourse has a minimal structure coming from the sequence of utterances from the participants, but there are more sophisticated notions of discourse structure that might be used (Lascarides and Asher, 2007). When a user presents a set of axioms in ontology authoring, these axioms should not be considered as individually independent sentences; there will be a coherence and fluency that is helpful for the ontology authors. For instance, when the user is in the middle of describing subclasses of a given class, entailments that mention those subclasses may be especially relevant and interesting.

Although we are not aware of any existing approaches using discourse structure for entailment selection, there are hints of useful approaches in other work. In particular, work on ordering and grouping of textual material generated from ontologies (see Section 3) is relevant. This is because any principle that indicates that axioms x and y should be grouped together (e.g. axioms "about" the same concept should be grouped together) can be converted into a principle for axiom selection - given that x is in the recent discourse context, axiom y should be selected (e.g. select axioms "about" the concepts currently being discussed).

2.4 Approaches using Pragmatics

A pragmatics-based approach uses explicit principles about what will be communicatively successful. In other words, it takes serious account of the fact that users will be human beings and will have goals and intentions.

One approach to modelling user's goals in ontology authoring is the use of *Competency Questions* (QCs). A CQ is a natural language question that the ontology author expects the ontology to be able to answer (Uschold et al., 1996). In (Ren et al., 2014), the authors have introduced the notion of *Competency Question-driven Ontology Authoring*. By collecting and analysing various CQs from two different domains, the authors proposed a set of patterns for modelling natural language CQs as OWL patterns. The authors categorised these patterns and extracted a set of presuppositions for each pattern. For example, a CQ "Which animals eat grass?" displays the pattern [WHICH [CE1] [OPE] [CE2]?], where CE1 and CE2 are concepts and OPE is a 2-place relation. Questions of this pattern suggest that the user has certain expectations, (known as *presuppositions* in the linguistics literature (Levinson, 1983)), which may or may not be entailed by the ontology. For example, the question suggests that animals should be able to eat grass. If an the ontology entails that $animal \sqsubseteq \forall eat.\neg Grass$ then one of the presuppositions of the CQ is violated.

We believe a user is likely to be especially interested in the presuppositions of his or her CQs, because if one or more of these are violated then the CQ is unanswerable. This insight can be exploited in the selection of entailments.

Some of the suggestions that we have made in the above for guiding the *selection* of entailments have their roots – or at least bear strong similarities – to ex-

[3] Richard Power, personal communication

isting ideas and methods in theoretical and computational linguistics. The same will be true for our (more tentative) ideas about *ordering* and *grouping* entailments.

3 ORDERING AND GROUPING

If we take the shape of the axioms as given (e.g., we do not translate logic axioms into a more readable format, as in e.g. (Power, 2014)) then, apart from selection, the other issues to be addressed are ordering and grouping of entailments. These affect the presentation of finite as well as infinite sets of axioms[4] and are relevant regardless of whether axioms are presented graphically or in a list.

Interestingly, the task of entailment *selection* interacts with the other tasks of *ordering* and *grouping*. One of the reasons why the list of axioms presented in Section 1 is difficult to digest is that the axioms were listed in a somewhat arbitrary order and without any further structuring, that is, as a simple enumeration. There is no point in selecting a set of axioms that cannot be coherently grouped and ordered. Similarly, if we are expecting to order or group axioms in a particular way then this may help us to select axioms in the first place. The following work addresses the problems of ordering and grouping and presents solutions that may have relevance for selection.

In OntoVerbal, a distinction between direct and indirect axioms is made. This basically means producing a syntactic characterisation of what concept an axiom is mainly "about" (the "topic" of the axiom). So $A \sqsubseteq B$ is directly about A but only indirectly about B. Directness and also syntactic complexity are used as the basis for grouping together related axioms. The hypothesis is that by grouping and presenting axioms in a rhetorically coherent manner (following Rhetorical Structure Theory) with a single topic, not only is the coherence of the text maintained, but also the user is aided in recognising topic change.

NaturalOWL (Androutsopoulos et al., 2013) is another OWL verbaliser. When describing an OWL class or individual (target), it first presents all the statements that directly describe the target (primary targets); second, the system describes the desirable statements that are indirectly related to the target (second-level targets). Second-level targets are only generated when the target is an individual. Consistent with the linguistic literature (Halliday and Hasan, 1976), an attempt was made to do this in a "coherent" fashion, using text planning; a *local coherence* measure, based on Centering Theory (CT) (Grosz et al., 1995)) is applied; an additional mechanism prevents information repetition in the generated text. NaturalOWL also uses the complexity of the generated sentences to decide how triples are mapped to English sentences. A text might be overly complicated, which requires breaking down, or the sentences might be too simple and in need of aggregating. Aggregation relies on a set of thresholds for the number of sentences that can be aggregated.

[4] See e.g. http://www.w3.org/TR/2002/WD-rdfsyntax-grammar-20020325/, where *stripping* and *abbreviation* are employed to avoid repeated subjects or objects in adjacent RDF triples.

4 OUR EXAMPLE REVISITED

The example of 1 might serve to show some of the consequences that our ideas about selection, ordering and grouping might have. These remarks will be highly tentative and incomplete.

Logical Approaches. The rule suggested in Section 2.2 for choosing the *most specific* subsumer would remove an axiom such as $PH \sqsubseteq \neg CheeseTopping$ (axiom 11) from the entailments listed in Section 1, because the more general statement $PH \sqsubseteq \neg Topping$ holds. This principle would also cause replacing the axiom $PH \sqsubseteq (\leq 245.hasTopping)$ (axiom 12) by a stronger axiom such as $PH \sqsubseteq (\leq 4.hasTopping)$ (axiom 8). Many other entailments can be removed using similar principles.

Furthermore, consider the axiom $PH \sqsubseteq (\exists hasBase.(\exists isIngredientOf.SpicyPizzaEquiv) \sqcap ThinAndCrispyBase)$ (axiom 4), which implies $PH \sqsubseteq (\exists hasBase.ThinAndCrispyBase)$. Given that *hasBase* is functional, this makes the axiom $PH \sqsubseteq (\forall hasBase.ThinAndCrispyBase)$ (axiom 2) logically redundant, so if the functionality of roles is presented elsewhere then the latter can be removed.

Pragmatic Approaches. Suppose, for simplicity, the user has expressed only one Competency Question, namely "What base does a pizza have?". Then it is presupposed that *some pizzas are allowed to have bases*. If the ontology contains an axiom $\exists hasBase.\top \sqsubseteq Pizza \sqcap Topping$, then this implies that nothing can have a base (because Pizza and Topping are disjoint in the ontology). This makes it impossible for the ontology to meaningfully answer the competency question. If the user deletes the above faulty axiom, the user is likely to be interested in the consequence that now the ontology supports the idea that pizzas can have bases. Since this CQ presupposes the satisfiability of $Pizza \sqcap \exists hasBase.Base$, the fact that this concept is now satisfiable is also worth pointing out.

Discourse based Approaches. Suppose the system's analysis of Competency Questions has suggested focussing on entailments that make use of the *hasBase* role, but the user has recently performed a long series of authoring actions centring on the *hasIngredient* role. This suggests broadening the selection of axioms, focussing on entailments that make use of either the *hasBase* role or the *hasIngredient* role; For example, axioms 3, 5, and 6 may be of interest; although we can argue against the inclusion of axiom 3, if we have already mentioned $PH \sqsubseteq (\exists hasBase.ThinAndCrispyBase)$. In the (unlikely) absence of other information this might mean omitting all other entailments.

If and when genuine (i.e., non-redundant) information is omitted, the system should indicate what principles were used to guide the selection process, so as to avoid misunderstandings. For example, the system could say "this presentation focusses on the *hasBase* and *hasIngredient* roles. To view other entailed axioms, do ...". Explanations of this kind would fit neatly within the ideas suggested in the following section.

Syntactic Approaches. Ultimately, syntactic approaches might be more suitable for ordering and grouping than for selection, which is better handled on more principled grounds. Axioms could be grouped by dividing them into broad syntactic classes, for example, (1) any and all axioms of the form $PH \sqsubseteq (\exists hasIngredient.\varphi)$, (2) axioms of the form $PH \sqsubseteq (\forall hasIngredient.\varphi)$, (3) axioms of the form $PH \sqsubseteq (\exists hasBase.\varphi)$, (4) axioms of the form $PH \sqsubseteq (\forall hasBase.\varphi)$, (5) any and all other axioms. This is just one example of a possible grouping; users might be offered a choice between different types of grouping. The original example might result after the user has asked to see subsumers of the concept *PineappleHamPizza*.

5 INTEGRATION INTO AN AUTHORING ENVIRONMENT

(Power, 2014) argued that an ontology authoring system should 1. only generate informative statements, 2. detect redundancies (deciding whether to fix or report them), 3. identify and report contradictions, and finally 4. report implications.

Perhaps the most important aspect of this policy, in our view, is informativeness. Reducing the set of entailments and increasing the informativeness of this set can be achieved by a combination of syntactic, logical, and discourse-oriented approaches. While syntactic structures provide a starting point for limiting the set of entailments, only a logical framework can detect and address redundancy. Finally, the user's focus of attention may be tracked through a sequence of the user's authoring operations. If the system is able to grasp what the user's focus of attention is, this information can inform the selection task as well.

We have suggested extending the content selection policy by introducing and modelling users' requirements through competency questions. In an ontology authoring interface that allows the users to *submit* and *monitor* competency questions and their associated authoring tests (Ren et al., 2014), the system will have a better understanding of the users' requirements. Thus, in such an interface, when the system is faced with a large set of entailments, a natural way forward would be to select entailments and non-entailments based on such requirements. These requirements can also affect the grouping and ordering process.

Figure 1 presents a way of displaying CQs and their associated authoring tests in an ontology authoring interface. In this figure, the CQ "What cake has which cake filling?" is shown on top and is expanded with 5 different authoring tests. The CQ can be answered meaningfully only when all the authoring tests are passed. As shown in the figure, a green/red dot is used to indicate the pass/fail status of the authoring tests so that the users know what is preventing the CQ from being answered. In the example shown in Figure 1, users can find out that the *hasFilling* property has not been introduced into the ontology so that *Cake* and *CakeFilling* cannot be related.

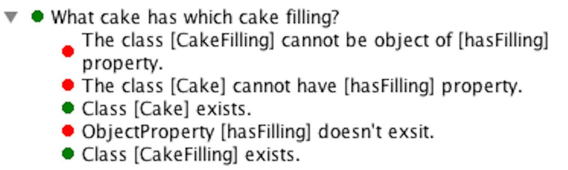

Figure 1: An example of the Competency Question tree with its associated authoring tests.

Suppose someone is using the interface to author an ontology and, at some stage, she is faced with two CQs that are as yet unfulfilled (i.e., marked red in the interface), and she is currently working on the first of these two. Suppose this is the CQ "What base does a pizza have?" that was discussed in Section 4, under **Pragmatic approaches**. Suppose, at some point, she manages to fulfil this CQ. At this point, particular axioms become worth presenting (as explained in Section 4). If this set of axioms is of a manageable size then axioms that are relevant in connection with the the last remaining axiom might also be presented (because they are relevant to the next authoring task). Presentation of axioms should also be subject to the logical, syntactical, and discourse-based considerations that we discussed elsewhere.

It seems to us that content selection, grouping and ordering should be developed in a way that more decisions are made by the adaptable authoring environment rather than reasoner side. Discourse-based approaches would have the best chance of coming to fruition if ontology editing used an intelligent dialogue-oriented authoring environment. Unlike previous dialogue-oriented interfaces, such as ROO, we envisage an ongoing dialogue with the user that can provide valuable insight into user's objectives, and in which occasional disambiguation dialogues could be initiated by the system.

Predicting *all* the logical consequences of an authoring action is a difficult task, even for the most experienced user. It might therefore be useful to offer users the option of asking *hypothetical questions*. We envisage that this could work as follows: If in doubt, the user would be able to ask what the consequences of a given authoring action would be. Upon this question, the system should present the entailments of this action in a transparent way (as discussed in this position paper), to allow him or her to understand the consequences of the action as clearly as possible. Finally, the user should be given the choice between committing or reverting the changes. We hypothesise that these new features would help to prevent errors in ontology editing and increase users' confidence in the quality of the ontology that they are creating.

6 SUMMARY

This paper has argued that the problem of entailment selection is largely neglected in software tools that (perhaps implicitly) claim to present entailments from OWL ontologies. On the other hand, there are a number of good ideas in current work for how the problem might be addressed in a more substantial way. It is interesting that many of the ideas about how to select entailments come from Linguistics or the Philosophy of Language, rather than from formal logic (Gricean maxims, presuppositions, discourse structure). That is, a solution to the problem needs to look at the nature of the *communication* within which the selection takes place. Finally, we discussed the added benefits of other restructuring mechanism, such as grouping and ordering.

ACKNOWLEDGEMENTS

This research has been funded by the EPSRC project: *WhatIf: Answering "What if..." questions for Ontology Authoring*. EPSRC reference EP/J014176/1.

REFERENCES

Androutsopoulos, I., Lampouras, G., and Galanis, D. (2013). Generating natural language descriptions from owl ontologies: the naturalowl system. *Journal of AI Research*, 48:671–715.

Appelt, D. E. (1987). Bidirectional grammars and the design of natural language generation systems. In *workshop on Theoretical issues in natural language processing*, pages 206–212.

Davis, B., Iqbal, A. A., Funk, A., Tablan, V., Bontcheva, K., Cunningham, H., and Handschuh, S. (2008). Roundtrip ontology authoring. Springer.

Denaux, R., Thakker, D., Dimitrova, V., and Cohn, A. G. (2012). Interactive semantic feedback for intuitive ontology authoring. In *FOIS*, pages 160–173.

Fuchs, N. E., Kaljurand, K., and Kuhn, T. (2008). Attempto controlled english for knowledge representation. In *Reasoning Web*, pages 104–124. Springer.

Grice, H. P. (1970). *Logic and conversation*. Harvard Univ.

Grosz, B. J., Weinstein, S., and Joshi, A. K. (1995). Centering: A framework for modeling the local coherence of discourse. *Computational linguistics*, 21(2):203–225.

Halliday, M. A. and Hasan, R. (1976). *Cohesion in English*. London: Longman.

Horridge, M. and Bechhofer, S. (2011). The owl api: A java api for owl ontologies. *Semantic Web*, 2(1):11–21.

Kalyanpur, A., Parsia, B., Sirin, E., Grau, B. C., and Hendler, J. (2006). Swoop: A web ontology editing browser. *Journal of Web Semantics*, 4(2):144–153.

Lascarides, A. and Asher, N. (2007). Segmented discourse representation theory: Dynamic semantics with discourse structure. In *Computing meaning*, pages 87–124. Springer.

Levinson, S. C. (1983). *Pragmatics*. Cambridge University Press.

Liang, S. F., Stevens, R., Scott, D., and Rector, A. (2011). Automatic verbalisation of snomed classes using ontoverbal. In *Artificial Intelligence in Medicine*, pages 338–342. Springer.

Mellish, C. and Pan, J. Z. (2008). Natural language directed inference from ontologies. *Artificial Intelligence*, 172(10):1285–1315.

Power, R. (2014). A formal dialogue model for ontology authoring. In *the 50th Anniversary Convention of the AISB*.

Ren, Y., Parvizi, A., Mellish, C., Pan, J. Z., van Deemter, K., and Stevens, R. (2014). Towards competency question-driven ontology authoring. In *The Semantic Web: Trends and Challenges*, pages 752–767. Springer.

Uschold, M., Gruninger, M., et al. (1996). Ontologies: Principles, methods and applications. *Knowledge engineering review*, 11(2):93–136.

Wolters, M. and Nolle, A. (2013). Clickona: An editor for dl-litea based ontology design. In *26th International Workshop on Description Logics. Ulm, Germany*.