A similarity based approach to omission finding in ontologies

Tahani Alsubait, Bijan Parsia, and Uli Sattler

School of Computer Science, The University of Manchester, United Kingdom
{alsubait,bparsia,sattler}@cs.man.ac.uk

Abstract. With the growing interest in using ontologies in semantically-enabled applications, the interest in enhancing the quality of such ontologies has grown as well. Standard reasoning services focus on certain obvious dimensions of quality, e.g., to detect inconsistencies and incoherence. In addition, bespoke tools have been presented to address the completeness dimension of quality, e.g., missing entailments. These tools are usually focused on very restricted subsets of all the possible missing entailments, i.e., only atomic subsumptions. We present a new protocol to detect both existing invalid entailments and missing valid entailments. We also present a case study to evaluate the usefulness of the presented protocol for ontology validation purposes.

1 Introduction

With the growing interest in using ontologies in semantically-enabled applications, the interest in enhancing the quality of such ontologies has grown as well. Ontologies can grow large in terms of size and complexity, making it challenging to maintain their quality and accuracy. Typically, the ontology development life cycle involves an ontology validation stage in which both ontology developers and domain experts come together to review the ontology and make sure it is free of errors. The most hard-to-spot errors in ontologies are the ones that do not make the ontology inconsistent or incoherent, though cause either undesirable or missing entailments. This is similar to the so-called "logical errors" in programming languages which cause the program to produce undesired output but do not cause compilation errors or abnormal termination. However, as is the case with programming languages, standard debugging tools cannot help in identifying such logical errors. This is why there is a need to develop tools and techniques for this purpose.

Indeed, there are many possible ways to find errors in ontologies. Direct ontology inspection can be effective but has the obvious disadvantage of being infeasible for large ontologies. In addition, direct inspection might be more effective for finding soundness problems (i.e., invalid entailments) rather than completeness problems (i.e., missing entailments) [9]. Other approaches have been proposed to address completeness problems. For example, Formal Concept Analysis (FCA) has been used for such a purpose [5]. FCA, in this context, is used to compute a concept lattice, i.e. a subsumption hierarchy of all conjunctions of concept

names occurring in an ontology, and the negations of these concept names. This lattice is then used to present successive questions to a domain expert to identify possible missing terminological or assertional axioms. A general observation about this approach is that they focus on finding missing atomic subsumptions. That is, they address *only* the completeness dimension (i.e., ignore soundness) and within the completeness dimension, they *only* consider subsumption relations between concept names (i.e., ignore complex subsumptions). Similarly, the approach presented by Lambrix et al. [8] is aimed at completing is_a hierarchies.

In this paper, we are suggesting a new protocol for finding omissions in ontologies. The protocol depends on asking a domain expert a set of multiple choice questions (MCQs) with high similarity between the correct and wrong answers.[1] Restricting the answer set to only those answers that are very similar to the correct answer can be useful in restricting the search space (in a principled way) when attempting to detect omissions. These omissions can be either missing *atomic subsumptions* or missing *complex subsumptions*. Using similarity to elicit knowledge from domain experts has already been used in well known elicitation techniques. For example, the triadic elicitation technique involves presenting 3 concepts to domain experts who are asked to identify the two similar concepts and explain why the third is considered different. Similarly, we present some statements that are entailed to be invalid by the ontology, yet they are very similar to a valid entailment and ask the experts to verify whether they are indeed invalid entailments or possibly missing valid entailments.

The questions presented to the expert should take the form of a multiple-response question[2] where the expert is asked to select all (and only) the correct answers. We re-use the question generation (QG) application described in [2] to generate questions that has exactly one answer entailed by the ontology to be correct. For the purpose of using these questions to validate the ontology, we select (for each question) a varied number, ranging for example from 1 to 10, of answers that are entailed to be wrong answers. The similarity between the key and distractors is set to be above a threshold. To measure the similarity between (possibly) complex concepts, we use the similarity measures presented in [4, 3]. To examine whether using a threshold of a high value has an impact on the number and type of the identified errors, we experiment with two different thresholds as we will describe in detail in Section 3. In general, since the wrong answers are selected to be similar to the correct answer, we question whether the ontology should entail that they are correct answers as well, i.e., a missing entailment.

As an example, consider the Java ontology that has been used in [1] as a knowledge source for generating educationally-useful MCQs. A detailed description of the ontology is presented in [1]. During the development of the Java ontology, we have witnessed the usefulness of looking at MCQs generated from this ontology for validating it on the fly. Some important "errors" in the ontology were easily identified by looking at the MCQs generated from it, in particular,

[1] In MCQ terminology, a correct answer is referred to as a key and a wrong answer is referred to as a distractor.

[2] In a multiple-response question, more than one answer can be correct.

MCQs with errors. Some errors were syntactic (e.g., typing mistakes) while others were logical (e.g., a wrong entailment identified by looking at an invalid key or a missing entailment identified by looking at an invalid distractor). Logical errors are generally harder to spot and considered more interesting when debugging an ontology. We briefly present some specific examples from the Java ontology in Table 1 and Table 2.

Table 1: Missing entailment example

Stem:	A feature of Virtual Machine Code is:
Key: Distractors:	(A) Portability (B) Write once Run Anywhere (C) Platform Independence (D) Reusability
Explanation of error:	the distractors are correct answers (i.e., all the answers are features of Virtual Machine Code)
Reasons for the (missing) entailment:	Those features have been asserted (in the ontology) to be features of Java Programming but not features of Virtual Machine Code. However, due to the similarity between the features (answers A, B, C, and D) they have all appeared in the answer list of this MCQ.

Table 2: Undesired entailment example

Stem:	Swing stands for:
Key: Distractors:	(A) Application Programming Interface (B) Abstract Windowing Toolkit (C) Java Foundation Classes
Explanation of error:	the key is not a correct answer (i.e., Swing does not stand for Application Programming Interface)
Reasons for the (undesired) entailment:	Swing $\sqsubseteq$ API API $\sqsubseteq \exists$ standsFor.ApplicationProgrammingInterface Therefore, the ontology entails that: Swing $\sqsubseteq \exists$ standsFor.ApplicationProgrammingInterface

Clearly, some logical errors in the Java ontology have resulted in producing the errors that appear in these MCQs. Identifying the errors in these MCQs by a Java expert has helped in finding and correcting some omissions in the

Java ontology. These examples show that looking at questions generated from an ontology can be fruitful for identifying some omissions in the ontology. In particular, it helped to identify invalid keys and distractors, i.e., answers that were thought to be correct while they are in fact wrong or vice versa.

In this paper, we present a case study to further explore the applicability of QG methods for ontology validation purposes. Rather than validating an ontology under development, we study the case of validating a previously built ontology in an attempt to suggest ways to improve it. We present some specific examples for possible errors in the SNOMED CT ontology as identified by some domain experts. In addition, QG methods can support ontology comprehension purposes which can be a goal in itself or it can be done prior to validating an ontology that has been built by a different ontology developer. We briefly tackle this in the study presented in this paper.

2 Implementing a prototype QG-based application for ontology validation

To evaluate the usefulness of the suggested QG-based approach for ontology validation purposes, we have implemented a prototype web-based application that (1) presents a selected set of multiple-response questions generated from an ontology to a domain expert (see Figure 1) and (2) based on the expert's answers, the application suggests some possible wrong and/or missing entailments in the ontology (see Figure 2). As we already described in the introduction, the questions in fact are generated such that they have only one answer which is entailed by the ontology to be correct. However, experts answering these questions are asked to pick all the answers they believe to be correct. Experts are also asked to indicate whether they are confident about their answers, per question. They can also leave a comment for a detailed explanation.

When the answers provided by an expert are different from the ones entailed by the ontology, the expert is asked to confirm their answers, as shown in Figure 3. The aim of this extra verification step is to encourage deeper engagement.

3 A case study

3.1 Goals

The main goal of this case study is to evaluate the usefulness of the suggested QG-based approach for ontology validation purposes. To address this goal, we try to answer the following question: Can a domain expert identify some omissions in an ontology by looking at MCQs generated from that ontology? We focus on a specific class of MCQs in which each wrong answer is similar to the correct answer (but entailed by the ontology to be a wrong answer). We expect that looking at such questions can reveal some omissions or missing statements (in the ontology) that might be difficult to spot without looking at the questions. This is because these wrong answers are similar to the correct answer and therefore

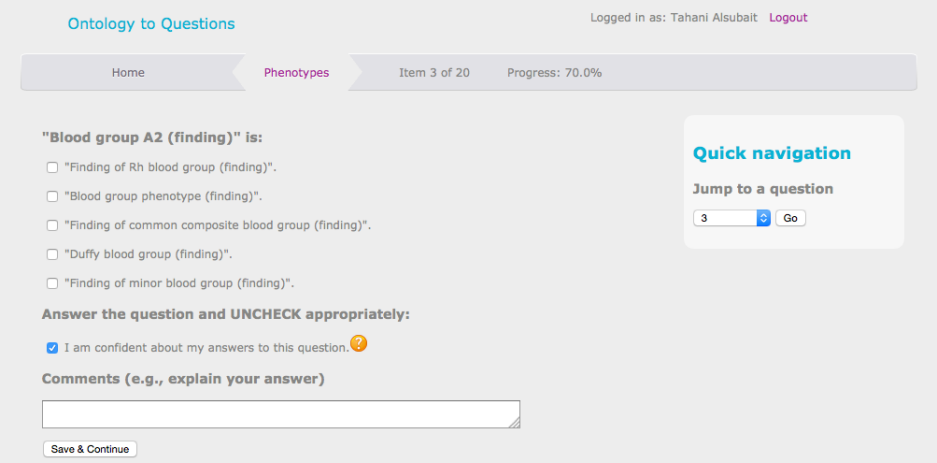

Fig. 1: QG-based support for ontology validation

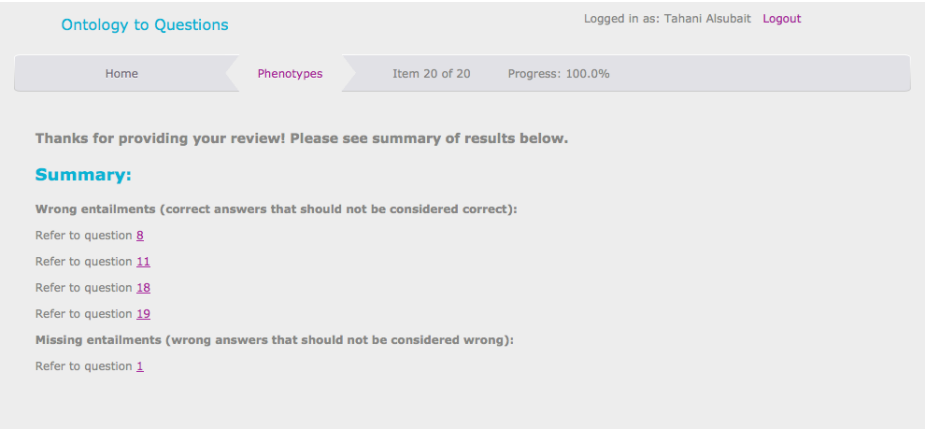

Fig. 2: Summary of suggestions to improve the ontology

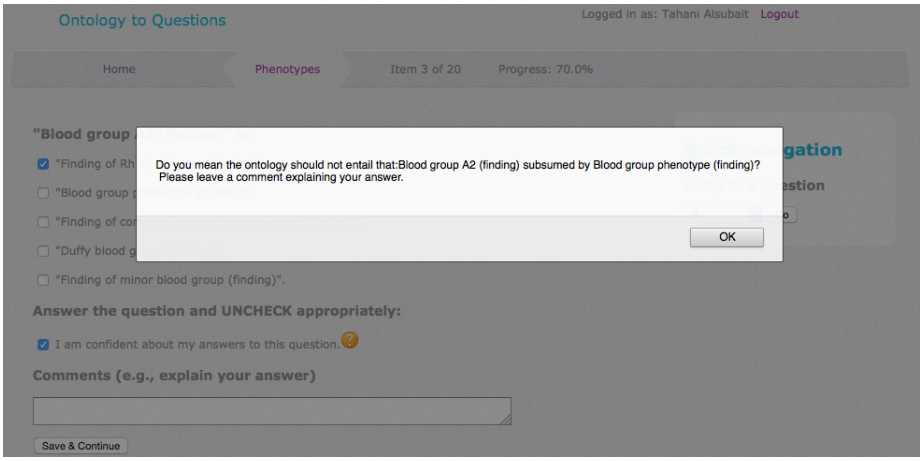

Fig. 3: Extra verification step

raise the question of whether they have been considered as wrong answers due to having any missing statements in the ontology or due to actual constraints in the domain. The missing statements that are intended to be detected can be either *atomic* or *complex* subsumptions. Missing or invalid atomic subsumptions highlight problems in the inferred class hierarchy of the ontology. Since this hierarchy is frequently looked at by ontology developers, we expect, in general, that there are more missing/invalid complex subsumptions rather than atomic subsumptions in a given ontology. We examine this hypothesis in the current study by looking at two sets of questions, Set A1 and Set A2. The questions in the two sets are constructed:

1. in Set A1: based on atomic subsumptions.
2. in Set A2: based on complex subsumptions.

Another goal of this study is to explore the impact of varying the similarity degree between the key and distractors on the overall usefulness of the generated questions for validation purposes. To examine this factor, we generate and compare two sets of MCQs, Set B1 and Set B2 which are described below. We try to answer the following question: Is looking at MCQs from Set B1 more useful for ontology validation purposes than looking at MCQs from Set B2? The MCQs in the two sets are generated such that the similarity between the wrong answers and the correct answer is:

1. in Set B1: above a threshold Δ_{max}.
2. in Set B2: below a threshold Δ_{max} but above a second threshold Δ_{min}.

The two sets A1 and A2 are not disjoint from sets B1 and B2. To examine all possibilities, we generate four disjoint sets of questions such that the questions:

1. in Set 1: are selected from Set A1 and Set B1.

2. in Set 2: are selected from Set A1 and Set B2.
3. in Set 3: are selected from Set A2 and Set B1.
4. in Set 4: are selected from Set A2 and Set B2.

3.2 Materials and methods

Ontology selection The current study requires the availability of a domain expert to answer a set of MCQs generated from a domain ontology. Due to the availability of an expert in BioInformatics, we have asked that expert to select some parts of an ontology which he thinks might be suitable for the purpose of this study. Due to the expert's interest in SNOMED CT in general and genetic findings in particular and his assumptions that the ontology is not detailed enough in this part, we have selected a (small) part of genetic findings that covers phenotypes (e.g., Blood groups). All the subclasses (197 classes) of the class *Phenotype* have been used as a seed signature to extract a $\bot$-module [10]. In addition, the object property *RoleGroup* has been added to the seed signature. This property is used to group certain properties together [12] and is necessary for extracting the module. The resulting module has a total of 246 classes and 6 object properties.

Generating questions Two sets of questions have been generated from the extracted module using the prototype QG application described in [2]. This prototype generates two different sets of questions, namely difficult and easy questions. The difficult questions are generated such that the similarity between the key and distractors is above the average similarity between all siblings in the ontology (or in the current study, the extracted module). The easy questions are generated such that the similarity between the key and distractors is above two thirds of the average similarity between all siblings in the module (but less than the average similarity between all siblings). For the current study, we consider difficult questions to be questions of Set B1 and easy questions to be questions of Set B2. After computing the average similarity between all siblings in the module, the thresholds Δ_{max} and Δ_{min} have been set to 0.88 and 0.587, respectively. The generated questions take the form "What is X?" where X is a class name and the answers are either class names or class expressions. This kind of questions is suitable for finding missing/invalid entailments that we are interested in. Among the generated questions, 223 questions have class-name-based answers, referred to as Set A1 questions, and 24 questions have class-expression-based answers, referred to as Set A2 questions. Among the class-expression-based questions, only 5 questions are suitable for Set B1 (i.e., the similarity between the key and distractors is above the threshold Δ_{max}). These 5 questions are referred to as Set 3 as defined in Section 3.1. Each question has exactly one key but the number of distractors was variable. If the number of generated distractors for a given question is more than 10, we randomly select 10 distractors out of the available ones. We have restricted the number of distractors to be below or equal to 10 to make the question answering phase manageable.

Answering questions Two domain experts have been asked to answer a total of 20 questions (5 questions from each of the four sets Set 1, Set 2, Set 3 and Set 4). The first expert is a bioinformatician and the second expert is a physician. The 20 questions were selected randomly from the set of generated questions in the previous step. Three samples of those questions are presented in Section 3.3. The questions were presented to the domain experts via the web-interface described in Section 2, see Figure 1. The first 10 questions are from Set A1 and the second 10 questions are from Set A2. We chose to present questions from Set A1 first, for deeper engagement, because they are expected to take less time to answer compared to questions from Set A2. Within Sets A1 and A2, questions from Sets B1 and B2 are randomly ordered. Also, a think-aloud technique was used to get a deeper insight into the advantages and limitations of the approach. The experts were allowed to use any external source to help them in answering the questions. After answering all the questions, the experts were asked to answer three last questions about their overall experience in answering the questions. These questions, which are shown in Figure 4, are:

1. Did any question help you to find any bugs in the ontology? Please explain.
2. Did any question help you to think about aspects of the ontology you had not considered before? Please explain.
3. Please provide any comments that could help us to improve this tool. Provide examples if possible.

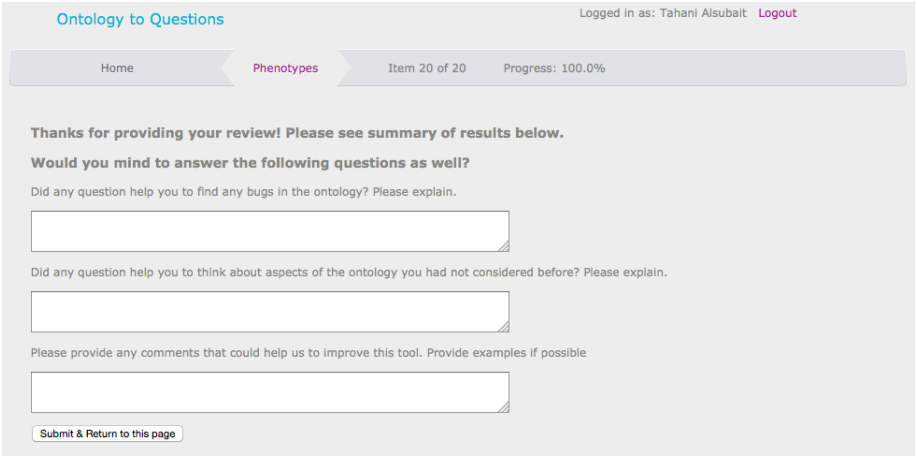

Fig. 4: Using QG-methods to validate ontologies

3.3 Results and discussion

For 9 out of the 10 questions in Set B1, the first expert's answers were correct, i.e., equivalent to what is entailed by the ontology. The only question for which

this expert's answers were different from the ones entailed by the ontology is the question presented in Table 3. This question is the only question which contains an answer that contains an existential restriction; all the other answers contain either class names or conjunctions of class names. The expert has identified both a missing entailment (invalid wrong answer) and a wrong entailment (invalid correct answer). In particular, the expert indicated that the ontology should entail that a finding of common composite blood group is subsumed by a finding of blood group and phenotype finding. He also indicated that the ontology should not entail that a finding of common composite blood group is subsumed by a finding of blood group and interprets (attribute) ABO and Rho(D) typing (procedure). The expert indicated that he was not confident about his answers to this question and explained that by reporting that he was not familiar with the terminology used by the ontology to describe the concepts presented in this question, e.g., interprets (attribute). In consistent with the first expert's answers, the second expert answered all the questions in Set B1 correctly; hence she did not identify any possible omissions in this part of the ontology.

Table 3: A first example for a question generated from SNOMED CT

Stem:	"Finding of common composite blood group" is:
Key:	(A) "Finding of blood group" and Interprets "ABO and Rho(D) typing"
Distractors:	(B) "Finding of blood group" and "Phenotype finding"

For 8 out of the 10 questions in Set B2, the first and second experts' answers were equivalent to what is entailed by the ontology. The two questions for which the two experts' answers were different from the ones entailed by the ontology are the questions presented in Table 4 and Table 5. In both questions, the answers are conjunctions of class names. Again, in both questions, the experts have identified a missing entailment (by selecting one of the distractors) and a wrong entailment (by not selecting the expected key). Both experts have agreed on the wrong answer that they chose to select as an answer. The two experts have indicated that they are not confident about their answers to these two questions. The first expert explained why he was not confident about his answers to the question presented in Table 4 by pointing out that one of the terms used in the question, i.e., inherited, seems irrelevant since all blood groups are inherited. For this question, the experts indicated that the ontology should entail that inherited weak D phenotype is subsumed by blood group phenotype and finding of minor blood group. Similarly, for the question presented in Table 5, the experts indicated that the ontology should entail that trans weak D phenotype is subsumed by blood group phenotype and finding of minor blood group.

In total, the first expert indicated that he was confident when answering only 7 questions out of the 20 questions. The second expert was confident when answering 13 questions out of the 20 questions. The first expert explained that by pointing out that although the terminology used in the ontology might seem

Table 4: A second example for a question generated from SNOMED CT

Stem:	"Inherited weak D phenotype" is:
Key:	(A) "Blood group phenotype" and "Finding of Rh blood group"
Distractors:	(B) "Blood group phenotype" and "Finding of ABO blood group"
	(C) 'Blood group phenotype" and "Duffy blood group"
	(D) "Blood group B" and "Blood group Para-Bombay"
	(E) 'Blood group phenotype" and "Finding of minor blood group"

Table 5: A third example for a question generated from SNOMED CT

Stem:	"Trans weak D phenotype" is:
Key:	(A) "Blood group phenotype" and "Finding of Rh blood group"
Distractors:	(B) "Blood group phenotype" and "Duffy blood group"
	(C) "Blood group B" and "Blood group Para-Bombay"
	(D) "Blood group phenotype" and "Finding of minor blood group"
	(E) "Blood group phenotype" and "Finding of ABO blood group"

to be natural to an ontology developer, it does not seem to be natural for a subject matter expert. Consistent with this, the second expert reported that the language of questions made it difficult to interpret what the question was asking. The first expert also reported that the questions seem to be of varying difficulty. For example, he pointed out that answering most of the questions from Set A1 was straightforward. These questions use only class names as answers. In contrast, he reported that questions two questions from the same set, which also use only class names as answers, were harder to answer. He explained that by pointing out that the answers were very similar and hence he found it difficult to decide which answer is the correct answer. The answers to these questions were: Blood laboratory and Blood bank which are indeed similar (yet refer to different departments). The first expert further explains that he selected what he thought was the best answer, rather than the only correct answer. Consistent with this, the second expert reported that, for the exact two questions, she picked what she thought was the best answer. The experts did not identify any missing entailments in these two questions, i.e., they did not indicate that a wrong answer should be a correct answer. However, their explanation supports the hypothesis we are testing in this study, i.e., looking at MCQs with distractors that are similar to the key can be helpful in identifying missing entailments.

As described earlier, the similarity between the key and distractors in questions from Set B1 is higher than the similarity between the key and distractors in questions from Set B2. Although one would expect that questions in Set B1 would reveal more omissions in the ontology compared to questions in Set B2 (because the wrong answers are more similar to the correct answers), this was not the case. Questions in Set B1 have identified 2 (possible) omissions while questions in Set B2 have identified 4 (possible) omissions. This can be explained

by the fact that errors can occur in *different* parts of the ontology. For example, questions in Set B1 would identify missing subsumees that are very close to their (potential) subsumer, e.g., in the inferred class hierarchy. In contrast, questions in Set B2 would identify missing subsumees that are not very close to their potential subsumer. In general, looking at this (rather small) set of questions was helpful in spotting some omissions in the ontology and suggesting improvements. Consistent with our expectations, the results also show that the method may be generally more helpful in identifying invalid/missing entailments involving complex subsumptions, i.e., Set A2, rather than atomic subsumptions, i.e., Set A1.

The aim of the second and third question presented to the experts after answering the questions was to evaluate the usefulness of the presented MCQs to support ontology comprehension purposes. According to the answers provided by the experts, the questions were not very helpful in identifying new aspects of the ontology they had not considered before. The first expert pointed out that this is due to having (1) questions that seem to be unnatural to a subject matter expert (due to describing concepts in an uncommon way) and (2) changes in the difficulty level of the questions (partly due to the first point). He further explains by pointing out that these two points might limit the usefulness of this form of MCQs for supporting students who want to learn about the subject. The second expert, who is a physician, did not respond to this question as she was not familiar with the ontology.

3.4 Related work

Baader et al. [5] presented a FCA-based approach for completing Description Logics-based knowledge bases. Their approach is aimed at extending both the terminological and the assertional part of the knowledge base, i.e., the TBox and the ABox, respectively. A *Protégé* plugin implementing this approach is presented in [11].

Bertolino et al. [6] have investigated the use of QG-based methods for validation purposes. Their method aims at validating models in general and can be applied to ontologies as well. A set of True/False questions generated from an (altered) model are presented to a group of domain experts. The responses gathered from domain experts are used to validate the model. The method proposed by Bertolino et al. is different from our method in that they suggest to alter the model by deliberately introducing some errors in it before the QG step. Their method is also suitable for finding invalid entailments but not missing entailments. Although they have reported that their method have helped the recruited experts to think about new aspects of the domain which they have not considered before, the method does not guarantee that this applies to the unaltered (error-free) parts of the domain only.

Another related work is the approach presented by Dragisic et al. [7] that takes already found missing entailments as input and suggest logical solutions to repair the ontology by possibly adding missing axioms. Their approach is extended in [8] by attempting to repair ontologies without given missing entail-

ments. This approach is different from our approach in that it only considers missing atomic subsumptions.

4 Summary and future directions

We have suggested a new protocol for finding omissions in OWL ontologies. We have also presented a case study for evaluating the usefulness of the suggested protocol for ontology validation purposes. Although the results seem to be promising, they are far from significant. Further efforts are needed to improve and evaluate the presented strategy. In particular, more user studies are needed. As a future work, we plan to implement a *Protégé* plugin to allow ontology developers to benefit from the suggested protocol.

References

1. T. Alsubait, B. Parsia, and U. Sattler. Generating multiple choice questions from ontologies: How far can we go? In *Proceedings of the First International Workshop on Educational Knowledge Management (EKM 2014)*, 2014.
2. T. Alsubait, B. Parsia, and U. Sattler. Generating multiple choice questions from ontologies: Lessons learnt. In *The 11th OWL: Experiences and Directions Workshop (OWLED2014)*, 2014.
3. T. Alsubait, B. Parsia, and U. Sattler. Measuring similarity in ontologies: A new family of measures. In *Proceedings of the 19th International Conference on Knowledge Engineering and Knowledge Management (EKAW 2014)*, 2014.
4. T. Alsubait, B. Parsia, and U. Sattler. Measuring similarity in ontologies: How bad is a cheap measure? In *27th International Workshop on Description Logics (DL-2014)*, 2014.
5. F. Baader, B. Ganter, B. Sertkaya, and U. Sattler. Completing description logic knowledge bases using formal concept analysis. In *Proceedings of IJCAI 2007*, 2007.
6. A. Bertolino, G. DeAngelis, A. DiSandro, and A. Sabetta. Is my model right? let me ask the expert. *Journal of Systems and Software*, 84(7):1089–1099, 2011.
7. Z. Dragisic, P. Lambrix, and F. Wei-Kleiner. Completing the is-a structure of biomedical ontologies. In *10th International Conference on Data Integration in the Life Sciences*, 2014.
8. P. Lambrix, F. Wei-Kleiner, and Z. Dragisic. Completing the is-a structure in light-weight ontologies. *Journal of Biomedical Semantics*, 6(12), 2015.
9. J. Rogers. *Development of a methodology and an ontological schema for medical terminology.* PhD thesis, Department of Computer Science, 2004.
10. U. Sattler, T. Schneider, and M. Zakharyaschev. Which kind of module should I extract? In *Proceedings of the 22nd International Workshop on Description Logics (DL-09)*, 2009.
11. B. Sertkaya. A Protégé plugin for completing OWL ontologies. In *The Semantic Web: Research and Applications*, pages 898–902. Springer Berlin Heidelberg, 2009.
12. K. Spackman, R. Dionne, E. Mays, and J. Weis. Role grouping as an extension to the description logic of ontylog, motivated by concept modeling in snomed. In *Proceedings of the AMIA Symposium: American Medical Informatics Association*, pages 712–712, 2002.