# Characterising Competency Questions for Ontologies

C. Maria Keet[1,*], Zubeida Casmod Khan[2]

[1]Department of Computer Science, University of Cape Town, South Africa
[2]Council for Scientific and Industrial Research, Pretoria, South Africa

## Abstract

Competency Questions (CQs) are widely used in ontology development to guide the scoping and validation stages, among others. However, very limited guidance exists for formulating CQs and assessing whether they are good CQs, leading to issues such as ambiguity and unusable formulations. To solve this, one requires insight into the nature of CQs for ontologies and their constituent parts, as well as which ones are not CQs for ontologies. We aim to contribute to such theoretical foundations in this paper, which is informed by analysing questions, their uses, and the myriad of ontology development tasks. This resulted in a first conceptual model for competency questions, which comprises five main types of CQs: Scoping (SCQ), Validation (VCQ), Foundational (FCQ), Relationship (RCQ), and Metaproperty (MpCQ) questions. This model enhances the clarity of CQs and therewith aims to improve on the effectiveness of CQs in ontology development, thanks to their respective identifiable distinct constituent elements. We illustrate and evaluate them with a use case and demonstrate where which type can be used in ontology development tasks. To foster use and research, we enhanced our preliminary annotated repository of CQs (ROCQS), that incorporates an existing CQ dataset, new CQs and CQ templates that further demonstrate distinctions among types of CQs, and annotated problematic questions.

## Keywords

Competency Question, Ontology Engineering, Ontology Development, Foundational Ontology, Ontology methodology



## 1. Introduction

Competency questions (CQs) are used within a variety of domains, such as education to design curricula using Bloom's Taxonomy, performance evaluation to assess an employee's performance, and assessing fitness of interrogation in a trial [1, 2]. Within ontology development, CQs are used throughout the processes to guide the development, such as in the NeOn methodology [3] and test-driven development [4], including scoping the ontology [5], aligning ontologies [6], validating content coverage [7, 8, 4], and to assist in interrogating the ontological nature of an entity during alignment to an entity in a foundational ontology [9].

However, CQs are used in different ways and sometimes incorrectly and ambiguously [10, 11]. Domain experts need guidance on what good CQs look like [12], and ontology engineers have difficulties working with CQs in a systematic way from the beginning to the end of the ontology development task [13]. In some cases CQs are not answerable due to too imprecise wording for it to be amenable to a formalisation into a logic or query language [14] or due to expressiveness limitations or content coverage limitations. Further, existing ontology development methodologies make little use of elicitation and modelling techniques for CQs. Consequently, ontology developers have limited resources for developing CQs, other than the CLaRO controlled natural language and tool for CQ authoring [15] and preliminary automation of CQ generation [10, 12, 16]. Incomplete or ambiguous CQs, in turn, may result in incoherent or incomplete ontologies and the ontology-based systems that use them. Despite a recent increase in popularity in adoption of CQs, what they exactly are, or should be, or what 'exemplary' or 'good' CQs should look like, remains unclear, let alone what different types of CQs are specifically.

*Proceedings of the Joint Ontology Workshops (JOWO) - Episode XI: The Sicilian Summer under the Etna, co-located with the 15th International Conference on Formal Ontology in Information Systems (FOIS 2025), September 8-9, 2025, Catania, Italy*
*Corresponding author.
mkeet@cs.uct.ac.za (C. M. Keet); zdawood@csir.co.za (Z. C. Khan)
0000-0002-8281-0853 (C. M. Keet); 0000-0002-1081-9322 (Z. C. Khan)

We aim to contribute to theoretical insights into CQs for ontologies that, in turn, may address these problems by devising a first categorisation of CQs into five different principal types of CQs, being Scoping (SCQ), Validation (VCQ), Foundational (FCQ), Relationship (RCQ), and Metaproperty (MpCQ) questions, not only by informal description [17], but in detail. They each have a distinct purpose and, consequently, different characteristic constituents of a question of that type, showing that they are indeed distinct and distinguishable. This is further made clear in a conceptual Model for Competency Questions (QuO) and the formalisation of each type of CQ. In consulting the literature and ontologies to attempt to source examples, it showed a dearth of foundational, relationship, and metaproperty CQs, and therefore we created more CQs of those types, which further demonstrate their distinctions for the various ontology development tasks. To foster use, reuse, and further research, we (1) describe where the different types of CQs would typically be used at different stages and tasks in the NeOn methodology, (2) demonstrate use with a use case, and (3) extend the FAIR-compliant repository of CQs from [17], which categorises the CQs by type and their key constituents, which incorporates both an existing CQ dataset and new CQs and CQ templates.

The remainder of the paper is structured as follows. Section 2 discusses related work. This is followed by the introduction of the Questions for Ontologies (QuO) conceptual model in Section 3. Use-cases, their use in an ontology development methodology, and the repository of CQs, ROCQS, are elaborated on in Section 4. Lastly, we discuss in Section 5 and conclude in Section 6.

## 2. Related Work

The first mention of CQs for ontologies can be traced back to 1996: Uschold and Gruninger describe CQs as those questions that "specify the requirements for an ontology and as such are the mechanism for characterising the ontology design search space" [5]. They distinguished between informal and formal CQs, where informal CQs still must be consistent with the axioms in the ontology and serve as constraints on what the ontology can be, rather than determining a particular design with its corresponding ontological commitments.

Wisniewski et al. [11] collected 234 informal CQs for five ontologies in different subject domains, and analysed and translated them in SPARQL-OWL where possible, resulting in a first dataset of CQs, their patterns, and SPARQL-OWL queries [14]. The analysis showed there to be no 1:1 mapping to OWL constructs as had previously been assumed, which is partly due to modelling styles and partly due to different natural language sentence constructions for the same formalisation. Wisniewski et al.'s patterns also served as input to the development of the CLaRO controlled natural language to author CQs [15]. Since CLaRO is derived from existing CQs of domain ontologies, it is unclear whether they are also suitable for CQs such as those dealing with meta-properties in ontologies or CQs relevant for foundational ontologies. This also remains the question with their more recent work on automating CQ generation, since it has CLaRO in the loop as part of the CQ generation pipeline [10, 12]. A LLM-based method to generate CQs for existing ontologies to foster ontology reuse showed promising results, but also that some CQs are incorrect, could not be validated by the developers, or were not matched or usable due to issues such as ambiguity and subjectivity [16], and similarly for explorations of prompting LLMs for CQs for a new ontology [18].

Monfardini et al. [13] surveyed the use of CQs in ontology engineering recently, alongside their benefits and challenges. The results show that CQs have been considered useful and have helped mainly to define an ontology's scope and to evaluate an ontology, and that, mostly, CQs were authored iteratively and refined along the ontology development process. The authors conclude that the lack of practical and detailed guidelines and supporting tools contributes to the difficulties faced by ontology engineers. This might be a cause of Guizzardi and Guarino's [19] claim of an issue of vague or unspecific competency questions that hinders ontology development. Guizzardi and Guarino also contend that CQs typically serve as mere lookup queries within a model's structure but that their true value lies in their ability to guide ontology construction when framed as genuine context-dependent requests for explanation. This indicates CQs are expected to serve multiple purposes.

To some extent, the latter emerges also from ontology engineering methodologies. Among others, Tropos has been proposed to visually capture and refine CQs as goals, enabling analysis through relationships with plans and resources [20]. Though neither Tropos, nor aforementioned CLaRO, consider validation with CQs. The NeOn methodology [3] and test-driven development [4] do assume CQs for both scoping and validation, or for validation specifically [7, 8]. They also have been proposed for use in other ontology development tasks, such as to assist aligning ontologies [6, 9]. Thus, CQs may be of use in various ways throughout the ontology engineering process.

Overall, there is limited assistance in CQ authoring for ontologies and some confusion, notions of CQ quality and of serving different tasks and goals, and that nonetheless CQs are already perceived to be of use.

# 3. Model for Competency Questions

## 3.1. Preliminary analysis: purposes and core elements

As a first step towards model development, we elicitate possible purposes of CQs in relation to ontologies and entities represented in them, and elements that are relevant to CQs.

### 3.1.1. Purposes of CQs

Let us proceed to be more specific about direct and underlying motivations, which can also be referred to informally as *purpose*, *goal*, *function*, or *intent* beyond a mere 'information elicitation' function of a question, and therewith working towards a *why* to ask and answer a CQ. Consider:

- *Demarcation of the subject domain* for the ontology; e.g., "Which plant parts does Pumba the warthog eat?" helps demarcating the subject domain of the ontology to plants and animals (at least)—thus also excluding, among many topics, building architecture—and seems to require both type-level and instance-level information, i.e., a full knowledgebase rather than only the TBox, and thereby spur further interrogation about the type of artefact intended.
- *Validation* whether the right content is covered, which includes both classes/relationships/attributes of the domain and relevant axioms; e.g., querying the African Wildlife Ontology for "Which plant parts do warthogs eat?", it requires from the ontology that it contains the entities named Plant part (with subclasses), Warthog, and either the property eats or its reified version Eating, or their synonyms, and suitable axioms that relate these entities, such as, e.g., $\mathsf{Warthog} \sqsubseteq \exists \mathsf{eats.PlantPart}$.
- *Alignment of a domain entity to a foundational ontology entity*; e.g., and instantiating one of the BFO alignment questions from [9], "Is each warthog wholly present at different times?", where a 'yes' means warthogs are at least a type of endurant.
- *Elucidation or interrogation of ontological characteristics* of the entity; e.g., "Is each instance of Tree necessarily (at all times of its existence) an instance of Tree?" where a 'yes' means being a tree is a rigid property of trees.

This list of motivations—scoping, validation, alignment to a foundational ontology, and property interrogation—may not be exhaustive, but already demonstrates not all CQs are alike.

### 3.1.2. Components and attendant entities of CQs

One can observe from the aforementioned examples that different CQs have various *components*, which are essential for distinguishing between them. For instance, the "unfold in time" is a description of a key characteristic of perdurants, which are included in a foundational ontology, and there are mentions of specific entities (warthog, Pumba, eats etc.). More precisely, elements used by one or more types of CQ can be one or more of:

- An existing ontology or a prospective ontology yet to be developed, $o \in \mathcal{O}$, which may be specified as a domain ontology (also including core ontologies and top-domain ontologies), $o_d \in \mathcal{O}_D$, or a foundational or top-level ontology, $o_f \in \mathcal{O}_F$; e.g., the IDO and BFO, respectively;

– Domain entity, $d \in \mathcal{D}$, which refers to an entity $d$ in subject domain $\mathcal{D}$; e.g., the SARS-CoV-2 virus in the domain of virology, running in the domain of athletics;
– Subject domain $s \in \mathcal{S}$; e.g., building architecture, infectious diseases, data mining;
– Entity in the ontology, $e \in \mathcal{E}$; e.g., dolce:Perdurant in the DOLCE foundational ontology and awo:lion in the African Wildlife Ontology;
– Properties, be they 'meta', like being a sortal, mixin etc., $n \in \mathcal{N}$, or 'metameta', $m \in \mathcal{M}$, such as rigidity and unity that determine whether the entity is a sortal etc., or a relational property, $r \in \mathcal{R}$, such as transitivity and symmetry.
– Purpose (in its commonsense interpretation) of a CQ, $p \in \mathcal{P}$, such as scoping of the ontology, validation or verification, foundational ontology alignment, metaproperty analysis, or relationship assessment or representation.

The foundational ontology entities are expected to remain relatively static. The elements referred to in the last two items are *de facto* closed sets and may be assumed to change only rarely. The meta properties in $\mathcal{N}$ from [21] are Type, Quasi Type, Material Role, Phased sortal, Mixin (that are all Sortals), Category, Formal role, and Attribution (all are non-Sortals). The main/most used properties in $\mathcal{M}$ are rigidity (rigid, non-rigid, anti-rigid), identity (carrying or supplying an identity criterion), unity, and dependence (externally or not), which also form the basis of OntoClean [22], while others are being explored on relevance for ontology engineering, such as telicity (telic or atelic) for processes [23]. Relational properties in $\mathcal{R}$ include transitivity, irreflexivity, reflexivity, symmetry, asymmetry, and also antisymmetry, acyclicity, and intransitivity beyond the OWL ecosystem [24] and possibly also purely reflexive and strongly intransitive [25].

### 3.2. Descriptions and definitions of the types of CQs

Based on the motivation and component analyses, we introduce an inclusive description for CQs and, being the principal focus of this paper, we identify five main types of CQs for ontologies: those for scoping (SCQ) and validating (VCQ) content of an ontology, to align to a foundational ontology (FCQ) or they concern metaproperties (MpCQ) and those revolving around representing relationships (RCQ). We do not exclude the possibility that more types of CQs for ontologies may be identified later. The overarching description of competency questions for ontologies is generic, because it is intended only for setting the context rather than being overly prescriptive at this stage. The key points are that not all questions are competency questions for ontology, and that there is a particular purpose for it. Variables used are those introduced in Section 3.1.2 above.

**Definition 1 (Competency Question for an ontology).** *A* Competency Question *(*CQ*), $q \in \mathcal{Q}$, regarding an actual or prospective* ontology $o \in \mathcal{O}$ *is a specialised type of question. Each $q$ has at least one purpose $p \in \mathcal{P}$ associated with it.*

The specific types of CQs defined below each have distinguishing properties to assist identifying and categorising them, where those properties can serve future methods for CQ authoring regarding which elements must appear in a CQ of that type. We provide both a definition of each type, based on a controlled natural language, and a rendering of the corresponding diagrammatic model snippet in EER notation (a conceptual modelling language), where the latter serves the purposes of

- usage, as basis for designing a database or knowledge graph for CQs for analysis, use, and reuse;
- highlighting of the essential components, irrespective of what the best language of formalisation would be; and
- communication, for understandability by non-ontologist CQ authors.

Their respective formalisations follow the usual pattern of logic-based reconstructions of EER.

In particular, the aim is to take a step towards elucidating the components and features of a CQ, and what relates in what way, to identify and substantiate the claim *that* there are differences and *where* there is at least one difference with other types of CQs, rather than investigating the ontological nature of a single element. Therefore, unless otherwise specified, those primitives introduced that are not further defined are to be understood in their usual, commonsense, meaning.

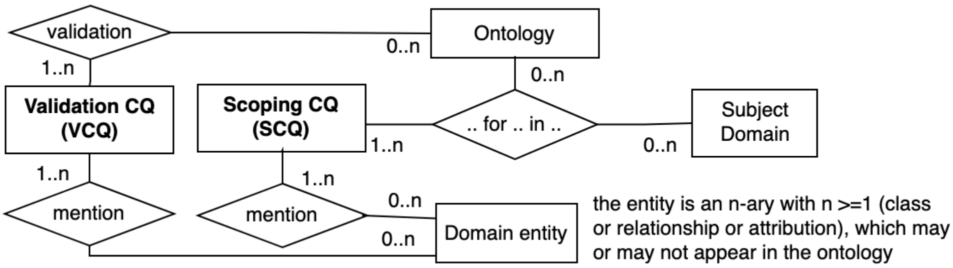


**Figure 1:** Illustration of the key aspects of SCQs and VCQs for ontologies, in EER notation with annotation.

We shall commence with the best known CQs, being the *Scoping CQ* (SCQ). They provide a rough demarcation of the possible worlds or interpretations for the ontology, or at least the so-called search space for candidate content to be added to the ontology from scratch to through reuse from an existing ontology. It is graphically sketched in Figure 1.

**Definition 2 (Scoping CQ).** *A* Scoping CQ *mentions at least one* domain entity $d \in \mathcal{D}$ *and contributes to describing the scope (as purpose* $p$*) for an* ontology $o \in \mathcal{O}$ *in a* subject domain $s \in \mathcal{S}$*.*

Formally, using a straight-forward formalisation of the EER model snippet, and with the vocabulary as primitives: $\forall x(SCQ(x) \rightarrow \exists y, z, w(mention(x, y) \land DomainEntity(y) \land usage(x, z, w) \land Ontology(z) \land SubjectDomain(w))$, where $Ontology$, $DomainEntity$, and $SubjectDomain$ are to be understood as described in Section 3.1, and $mention$ and $usage$ as primitives with their usual natural language meaning that can be suitably formalised accordingly. For instance, for SQC "Which animals are endangered?", $o_d =$ African Wildlife Ontology, $d =$ animals, and $s =$ is African Wildlife. SCQs typically would be expected to have non-empty answers with the elements asked for. For "Which animals are endangered?", the answer set should contain animal species and only those that are endangered. For an SCQ as a scoping device for an ontology's content, the actual answer may not be relevant, but it may assist CQ authoring if at least one sample answer is provided, if only to test that the question can be answered.

A *Validation CQ* (VCQ) may sound as if it were the same as, or else a kind of a SCQ, or the set of VCQs for an ontology to be a subset of the set of SCQs, since most SCQs are being reused for validation. There are subtle differences, however, with respect to the intent or purpose (validation versus scoping) and that they must be answerable by the ontology. SCQs need not be answerable, or even formalisable, whereas VCQs should be both formalisable and answerable. VCQs put testable constraints on the possible worlds or interpretations for the ontology as logical theory.

**Definition 3 (Validation CQ).** *A* Validation CQ *mentions at least one* domain entity $d \in \mathcal{D}$ *and is used for validating (i.e., as purpose* $p$*) the content of an* ontology $o \in \mathcal{O}$*. The expressivity of the logic required for formalising the VCQ (*$L_v$*) does not exceed that of the logic permitted for the ontology (*$L_o$*) (i.e.,* $L_v \subseteq L_o$ *must hold).*

The first part of the definition can be formalised as $\forall x(VCQ(x) \rightarrow \exists y, z(mention(x, y) \land DomainEntity(y) \land validate(x, z) \land Ontology(z))$. The logic (language) expressivity constraint may appear debatable. The constraint is included because without it, the VCQ formalisation may be beyond what possibly can be represented in the ontology. This would make the question unanswerable and impossible to validate, meaning it would not be a validation question. This suggests that being a VCQ may depend on the ontology development project, if the developers actively chose their $L_o$, or careful and precise formulation of a VCQ and/or training of CQ authors to write questions that can be validated.

While SCQs typically will ask for content, and VCQs may as well, VCQs may also be yes/no questions—at least the VCQs we collected do have a number of those types of questions (see ROCQS, introduced below). For instance, "Does (the) Dutch (language) have a region defined (where it is spoken)?" (based

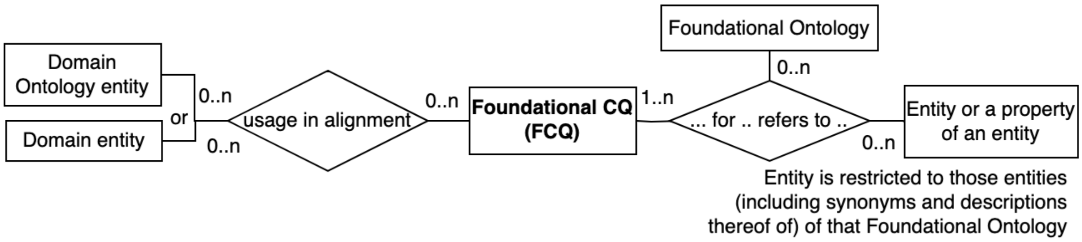


**Figure 2:** Illustration of the key aspects of FCQs, which are used to interrogate an entity to be aligned to an entity in a foundational ontology.

on the CQ with identifier mola_vcq_6 in ROCQS), which is *de facto* a question about the presence or absence of knowledge in the ontology, not reality, and "Does OpenOffice meet the ISO-4 standard?" (instantiation of the CQ swo_41 in ROCQS), which it either does or does not. As can be seen from these examples of collected CQs, they may return content from either the TBox or the ABox, as for SCQs.

By number of CQs that can be found in the literature, the SCQs far outnumber the others, to which we shall return in Section 4. The only other ones we could find, are those used to align an entity already in a domain ontology to an entity in a foundational ontology, *Foundational CQ*s (FCQ). They can be summarised as in Figure 2 and described as:

**Definition 4 (Foundational CQ).** *A* Foundational CQ *for a* foundational ontology $o_f \in \mathcal{O}_F$ *refers to an* entity $e \in \mathcal{E}$ *or characteristic thereof that is in the vocabulary of* $o_f$. *A FCQ is intended to be used to in the alignment process of a* domain ontology entity $e' \in \mathcal{E}$ *(with* $e \neq e'$*) to an entity* $e'' \in \mathcal{E}$ *that is in the vocabulary of* $o_f$ *(where* $e$ *may be the same as* $e''$*, but need not). An FCQ may also be used to interrogate a domain entity* $d \in \mathcal{D}$ *independently.*

The first sentence can be intuitively formalised as (and with abbreviations due to space and readability): $\forall x(FCQ(x) \rightarrow \exists y, z(referent(x, y, z) \land FoundOnto(y) \land Element(z) \land (containsVocab(y, z) \lor \exists w(containsVocab(y, w) \land hasProperty(w, z)))))$. While we could have chosen to push it into second order, since it refers to properties of elements, we chose not to, because for any implementation it will have to be pushed down to the instance level and would be stored as values in a database, and we follow for each EER model snippet the same common formalisation rules. The second part of the definition can be formalised in multiple logically equivalent ways; here, we reify the alignment relationship, resulting in: $\forall x(FCQ(x) \rightarrow \forall y, z(purpose(x, y) \land Alignment(y) \land aligningOf(y, z) \land (DomainEntity(z) \lor \exists w(Ontology(w) \land containsVocab(w, z)))))$.

Depending on the controlled natural language that may be designed for the FCQs, they may or may not have a slot for $d$ or $e'$ in the sentence. For instance, "Does it unfold in time?" can be disambiguated from the context when it is used, but one also can design a template "Does [...] unfold in time?" where the [...] is replaced with the name of the entity that is being aligned ($e'$). The 'unfold in time' description refers to perdurant of the DOLCE ontology, and thus is an example of a property $z$ of entity $e$.

Answers to FCQ questions in a decision diagram fashion may be either yes/no/not-applicable or any other answer option offered, where the consequence is arriving or staying at a certain entity in the foundational ontology rather that retrieving answers from the ontology as for SCQs and VCQs.

The *Relationship CQs* are underdeveloped, albeit known informally to some extent. They deal with the arity, or number of participants, of the relationship and ORM's notion of "elementary fact type" [26], the relationship's domain and range (i.e., participants), and whether one or more properties, such as transitivity, hold. Since they concern distinct aspects, we created subtypes of RCQs, as indicated in Figure 3 and described as follows.

**Definition 5 (Relationship CQ).** *A* Relationship CQ *can be used to determine several key characteristics of a relationship, each having their own purpose, being:*

- *a relationship's* arity *(arity CQ, aRCQ), where the relationship has an arity of at least 2, and at least two participants* $d, d' \in \mathcal{D}$ *(which may already be represented in the ontology as* $e, e' \in \mathcal{E}$*);*

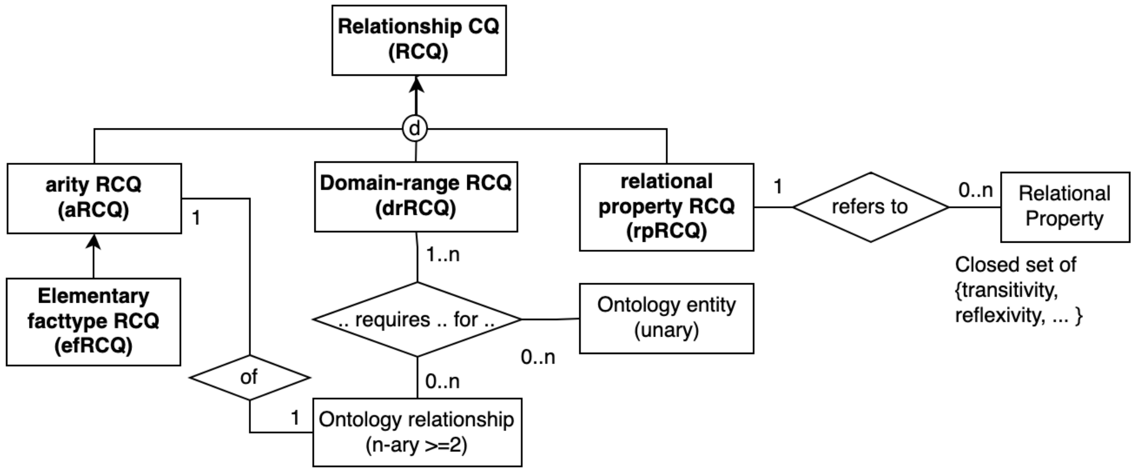


**Figure 3:** Illustration of the types of RCQs for ontologies and their salient aspects.

- *whether the relationship is elementary (efRCQ) and not further decomposable without losing information;*
- *its participating* entities *(domain-range CQ, drRCQ), which requires an* ontology entity $e \in \mathcal{E}$ *that is a unary (class/concept) for the* $n$*-ary* relationship $e' \in \mathcal{E}$ *(with* $e \neq e'$*) in the* ontology $o \in \mathcal{O}$*; or*
- *its* relational property *(relational property CQ, rpRCQ). The relational property* $r \in \mathcal{R}$ *asked about in a rpRCQ for a particular ontology* $o \in \mathcal{O}$ *should be one expressible in the representation language of* $o$*.*

Formally, and also here pushing it into FOL for the same practical reasons as for FCQs, $\forall x(RCQ(x) \rightarrow \exists^{=1}y\exists z(mention(x,y) \land Relationship(y) \land hasParticipant(y,z)))$. For efRCQ, the negation is less easy to capture effectively and the three sample questions are too few and diverse to extract features from. For instance, Halpin's "Can you rephrase the information in terms of a conjunction?" [26] and, inspired by an ORM context, "Does the uniqueness constraint span n or n-1 roles?". Functional dependencies have a rich literature in database theory, but its use is limited when one can represent only binary relationships as in OWL, and therefore we leave it without additional properties for the time being. The drRCQ can be formalised as follows $\forall x(drRCQ(x) \rightarrow \exists^{=1}y\exists z(requires(x,y,z) \land Relationship(y) \land Class(z) \land Ontology(w) \land containsVocab(w,y) \land containsVocab(w,z)))$. Finally, the rpRQC has only the basic, but unique among other types of CQs, reference to any one element of the closed set of relational properties from $\mathcal{R}$, hence, we obtain $\forall x(rpRQC(x) \rightarrow \exists^{=1}y(referent(x,y) \land RelationalProperty(y)))$.

Unlike the SCQs, VCQs, and FCQs, the RCQs can be either domain CQs or ontological CQs. Compare, e.g., drRCQs for the domain and range of the employs relationship, which normally takes domain ontology entities such as Organisation and Worker, given that employs is concrete, with an rpRCQ to interrogate its relational property with "If an organisation employs a worker, must it then never be the case that the worker employs the organisation?" that indicates a broader reach of the relationship. The drRCQs ought to have a universal reach as well, but practically, a number of ontologies have domains and ranges declared such that they are relevant for that particular ontology only.

Expectations of RCQ answers are again different, depending on the type of RCQ. For instance, "If a human loves, must it also love itself?" is expected to be answered by the modeller in the first instance, rather than be obtained from querying the ontology, and the answer is either reflexive or irreflexive, i.e., a property of the relation implied by 'loves', but it could be obtained from an ontology to check whether it has been declared. In contrast, an aRCQ such as "how many participants does a loving relation have?" must have a number as answer, which a SPARQL or SPARQL-OWL or DL query will not be able to answer or is trivial (all OWL object properties have two participants because OWL permits only binary relationships).

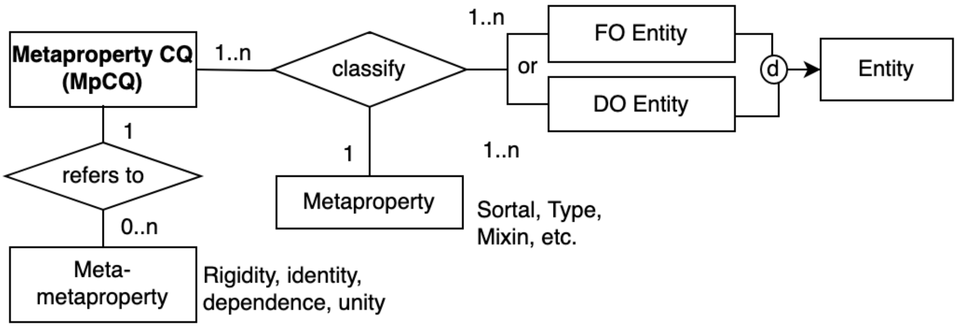


**Figure 4:** MpCQs for ontologies in EER notation, annotated with typical examples.

Last, the *Metaproperty CQ* (MpCQ) questions refer to a small set of metaproperties about the entities where intentionally the answer will be an absolute that holds across ontologies and, as with the RCQs, they seek to constrain the possible worlds or interpretations. Upon answering, one classifies the entity in a particular type of metaproperty. For instance, if "Are all persons necessarily always a person during their existence?" is answered in the affirmative, then that entity, person, is a sortal. A short description of MpCQs is as follows:

**Definition 6 (Metaproperty CQ).** *A* Metaproperty CQ *refers to exactly one* metameta- property $m \in \mathcal{M}$ *and is used to classify a foundational or domain* entity $e \in \mathcal{E}$ *or* $d \in \mathcal{D}$ *into, or as being a* metaproperty $n \in \mathcal{N}$.

The structure of an MpCQ is sketched in Figure 4. While we expect it to be used for a specific entity in either a foundational or domain ontology, this need not be the case. The diagram and definition can be formalised as: $\forall x(MpCQ(x) \rightarrow \exists^{=1} y \exists z, w(referent(x, y) \wedge Metametaproperty(y) \wedge classify(x, z, w) \wedge MetaProp(z) \wedge Entity(w)))$.

MpCQ answers are alike the rpRCQ ones, but then the single element answer is, e.g., 'telic' or 'anti-rigid', taken from $n \in \mathcal{N}$ rather than $m \in \mathcal{M}$. Considering expressiveness of popular ontology languages, it is expected it will be mostly modellers answering this question rather than querying the ontology for it.

In closing, comparing the scope of each type of CQ, note that SCQs and VCQs apply to an ontology as a whole (however the ontology answers the CQ), and VCQs may be specific to one particular ontology. This is arguably likewise for drRCQs, especially for domain-level relationships. FCQs, if linked to a particular FO, principally have that ontology as scope or relevance, albeit assuming applicability across ontologies. In contrast, the ontological CQs—aRCQ, efRCQ, rpRCQ, MpCQ—relate to an entity and the answer to the CQ ought to necessarily hold across all ontologies, although it may be that some ontology may not have the answer explicitly represented in the logical theory but only in the annotation due to expressivity limitations.

## 4. Usage of the Model for Competency Questions

We will introduce a use case and potential broader use on how QuO can assist first, and then describe how the CQs relate to methodologies, touch upon faulty CQs, and finally mention the updated repository of CQs, ROCQS.

### 4.1. Broad intended use: examples

We present a concrete use case in ontology development and then proceed to elaborating QuO's potential use in other phases and tasks in the ontology development processes.

### 4.1.1. Case study: Sewer networks

We commence with a free-form user story about ontology development of an ontology about sewer networks that introduces problems an ontologist may encounter. Subsequently we present where and how the various types of CQs have shown to be useful in the development of the SewerNet ontology [27].

> The ontology aims to assist with data and fault monitoring for the wastewater management domain. As part of building a comprehensive domain ontology, the ontology development team needs to define meaningful and formalised CQs but face difficulties. They are unsure how to scope the domain clearly, whether certain wastewater network concepts should be modelled at the instance or schema level, and whether repair of an element is a process or an event. Formalising complex relationships, such as transitivity constraints, poses challenges. Analysing and implementing these relationships requires consideration and may lead to unintended implications if not formalised accurately, such as incorrect deductions about faults of network element parts. Additionally, there is uncertainty on whether there should be foundational ontology mappings for such concrete tangible entities.

While the actual ontology developers (including one of the authors) did not create a user story, they did generate CQs as follows.

- **SCQs** to demarcate the subject domain, with a focus on domain entities such as monitoring and networking equipment; among others: "What can the pipes in a sewer network convey?" and "What is the material of the pipe?".
- **VCQs** to verify content with CQs. One of the VCQs was "What processes and events are involved in the network and its elements?", which was answered by executing the DL query `(process or event) and (participant some (Network or NetworkElement))` in Protégé v5.6.4.
- **FCQs** to align the domain entities to DOLCE; among others, "Is 'pipe circulation mode' something that can be perceived/measured (like color, size, smell, etc.)?", to which was answered 'yes' and, subsequently, the question "Is 'pipe circulation mode' an attribute of a (physical) object?".
- **RCQs** include a plethora of questions about relations. The efRCQs were discussed but not documented, and the domain and range axioms (drRCQs) and object properties with their already declared property characteristics (rpRCQs) in DOLCE were reused as-is. This induced an rpRCQ for a new subproperty, being "Is it true that if a node has weak connection to another node downstream, then that downstream node also has a weak connection to that first (upstream) node?" Consequently, symmetry was not declared for SewerNet's weak connection downstream, whereas dolce:weak-connection is symmetric.
- **MpCQs** concern properties of, typically, classes/unaries. They assisted disambiguation between Maintenance and Repair. For instance "Does maintenance have a definite endpoint?" and its counterpart "Is maintenance's endpoint determined outside that process?", which confirmed atelicity and therewith assisted representing Maintenance as a subclass of dolce:accomplishment, as compared to Repair's individual action [27]. The rigidity question for Operator, being "Is each instance of operator necessarily (at all times of its existence) an instance of operator?", quickly clarified it for the domain experts that it is anti-rigid.

As the SewerNet use-case demonstrates, each CQ type can be relevant for ontology development.

### 4.1.2. Prospective uses in methods and methodologies

More generally than the SewerNet case, different types of CQs can be useful at different stages in ontology development. To illustrate this, we consider the NeON methodology [3], among the available options, mainly because the overview diagram contains a comparatively large number of components and processes. In its summary figure, we indicated *where* which type of CQ may be used, as shown in Figure 5, whilst leaving the *how* for future work. Among others, SCQs are used in delineating the subject domain and setting the scope of the ontology (1. specification, 2./3. (non-)ontological resource

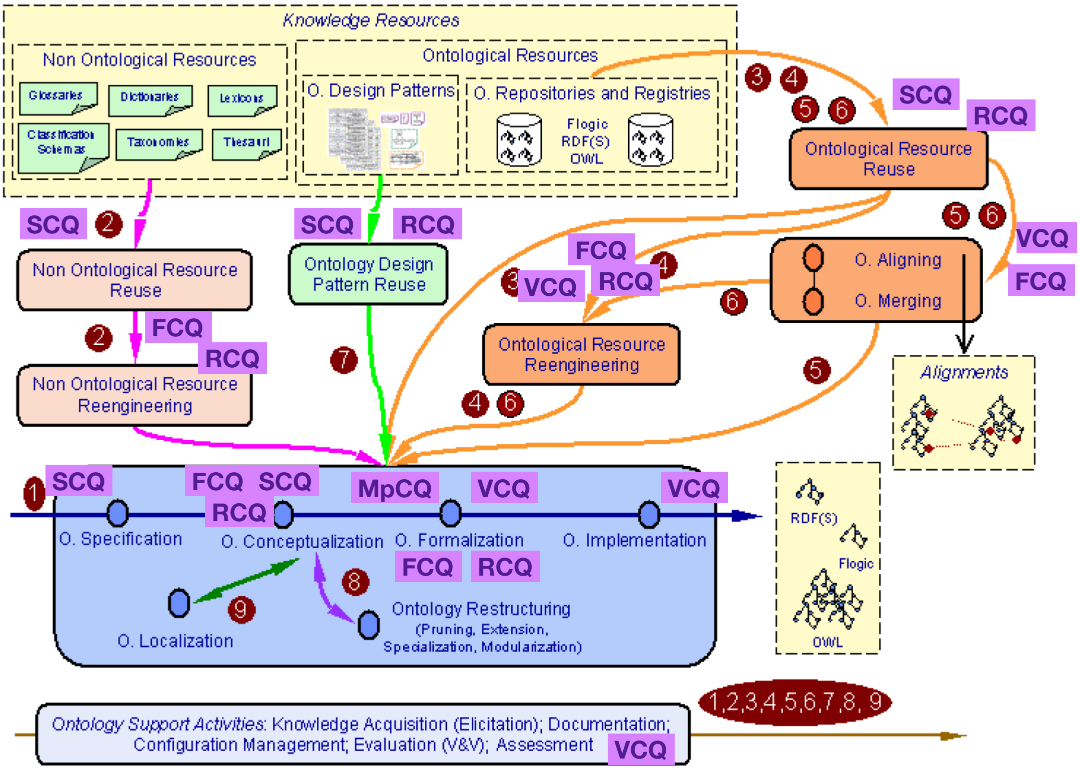


**Figure 5:** The NeON methodology annotated with where which types of CQ can be used with various tasks. For instance, SCQs can also be used to find ontologies for reuse (top-right, nos. 3, 4, 5, 6) and FCQs can assist with analysing non-ontological content (left, no.2), such as adding a top level to a thesaurus.

reuse). As ontology development progresses, VCQs become essential for ensuring the accuracy of information within the ontology (1. implementation). FCQs facilitate the alignment of domain entities with foundational ontologies (5. aligning, 1. formalization, 8. restructuring), enhancing clarity and semantic interoperability. RCQs focus on analysing the key characteristics of relationships within the ontology (1. formalization, 7. ODP reuse). They help formalise relationships in a way that promotes semantic interoperability, addressing issues related to the nature of relationships. MpCQs are relevant to classify entities based on their persistence and existence characteristics (1. formalization).

#### 4.1.3. Potential use in ontology-driven information systems

Besides CQ authoring for a domain ontology, one may consider CQs more broadly, such as possible consequences for an ontology-driven information system. That is, to not only use the ontology in the software, but also the CQs, both for assessing quality of the software and testing the data for adherence to the CQs.

For instance, CQs can link ultimately to, say, an advanced NLP system to improve the accuracy and relevance of various tasks that are mediated by a linguistic ontology. For instance, part-of-speech (POS) tagging, which assigns to each word a tag, such as noun, verb, and adjective. What the system is allowed to do, or prevent a POS tagger to make mistakes, may be constrained by the ontology. In the development of, or assessment whether to use or not, a particular ontology, say, OLIA [28], there may be a MpCQ “Must each annotation of a lexical unit (word) with a POS tag necessarily be NOT annotated with that POS tag for some time of the duration of the existence of that annotation relation?”. It interrogates about the metaproperty rigidity, with the following elements of the MpCQ definition: *metametaproperty* $m$ is Rigidity and domain *entity* $e$ is Lexical Unit. Rigidity for this CQ concerns whether a lexical unit may change POS tag. If the annotation relation were to be rigid (and tags disjoint), then any word that would be tagged differently across texts would contradict the knowledge in the ontology; e.g., ‘bat’ tagged as a noun (the animal) and elsewhere as a verb (to hit). Bat, the animal, is rigid; annotations of lexical units with their part-of-speech are not—a confusion of the two would result in problematic POS tagging software.

Another staple of NLP tasks is morphological analysis of words; e.g., where a word 'working' will be analysed and split up into 'work' as stem and 'ing' as gerund. An ontological assessment of that process, be it for an NLP ontology or the ontology-driven information system derived from it, may involve a CQ "Is something able to be present or participate in a morphological process?". This adheres to FCQ components for the DOLCE foundational ontology referring to its Event, with usage in alignment of OLIA's domain ontology entity morphological process: an event needs an object, therewith imposing a constraint on a prospective algorithm that it must have objects related to it. By propagating CQs and their answers into NLP systems, one can further ensure that they are better aligned with the theories of natural language, which is expected to contribute to better linguistic processing. The argument may analogously be extended to other ontology-driven applications.

## 4.2. Exemplary CQs, 'almost CQs', faulty CQs, and answerability

The previous sections assume CQs are correct. However, CQs can be problematic or 'faulty' for various reasons, such as i) due to syntax and semantics issues, ii) contextually due to expressivity mismatches of the query language or the ontology language, iii) due to the ontology's content, and iv) relative to the type of CQ. Due to space limitations, we briefly consider only the general idea how they manifest for causes ii and iv.

To illustrate cause iv: FCQs may be faulty in one principal way: it asks to make distinctions that are not made, or made differently, in the foundational ontology it was designed for and so it is of no use. For instance, to ask about process histories when trying to align the domain entity to DOLCE—that foundational ontology does not have process history—or to ask about abstract things when trying to align an entity to BFO, which does not contain abstract. Likewise, aforementioned SewerNet CQ on "something that can be perceived/measured" is useful for DOLCE but less so for aligning to an entity in BFO. It may also be the case that some FCQs are better than others because the questions are better understood by the developer [9].

Regarding bad RCQs, and rpRCQ specifically, they fall at least partially in issue ii: many computational ontology languages have very few relational properties or they are highly restricted in their use. While they may be ontologically challenging and push one's understanding of the entity, if the answer will not aid the modelling, evaluation, and use because it is beyond the representation or query language, then it is contextually a moot point. For instance, to investigate whether weak connection downstream is acyclic. Examples for other categories exist likewise, although it was difficult to devise clearly faulty MpCQs.

What may happen rather, is that a metaproperty does not apply to certain categories of entities but that such an entity is interrogated nonetheless, and then if the answer options are only, say, 'telic' and 'atelic', i.e., without an 'not applicable' third answer option, a question may not be perceived as answerable. For instance, the question "Is a house's endpoint determined outside that process?" is nonsensical in an endurantist commitment, since House is an endurant, not a perdurant and even less so a process.

There are also questions that seek to verify constraints, asking whether content is covered right. These are not CQs but rather general constraint-checking queries and we list a few types here for comparison.

- Checking for disjointness axioms between classes; e.g., "Are there animals that are both a carnivore and a herbivore?", where a 'no' means either verifying that the ontology has or entails the disjointness, or, when used in development mode [4], to add $\mathsf{Carnivore} \sqcap \mathsf{Herbivore} \sqsubseteq \bot$.
- Checking cardinality constraints; e.g., "Do humans have more than four limbs?" where upon a 'no' it must not be the case that the ontology contains a qualified cardinality constraint $\mathsf{Human} \sqsubseteq \geq n\, \mathsf{hasPart.Limb}$ where $n \geq 4$ without also $\mathsf{Human} \sqsubseteq = 4\, \mathsf{hasPart.Limb}$ or $\mathsf{Human} \sqsubseteq \leq 4\, \mathsf{hasPart.Limb}$.

**Scoping CQs**
What is the material of the pipe?
What are the types of wearable sensors?
Which animals eat [these animals]?

**Validation CQs**
Can [word] ever be placed in [slot]?
Is each dialect a language?
is ChickenTopping disjoint from HamTopping?

**Foundational CQs**
Are you able to be present or participate in `pouring (a drink)'?
Is `pipe circulation mode' wholly present at any time of its existence?

**Relationship CQs**
If a human loves, must it also love itself?
How many participants does [relation] have?

**Metaproperty CQs**
Does maintenance have a definite endpoint?
Is each instance of [X] necessarily an instance of [X]?

***Problematic questions***
** To what extent does [the software] support appropriate open standards?* (the 'to what extent' is vague)
** How can I get problems with [it] fixed?* (Too broad of a question; how-to)
** Where do I categorise bulk like [this bulk]?* (asks for instruction how to model)
** What is the gender information?* (imprecise: where? data? recorded?)

**Figure 6:** Sampling of CQs per type of CQ, illustrating both templated CQs and instantiated ones, and annotated problematic questions.

Based on the examples, these sort of questions appear to be Boolean queries, although not all Boolean questions are necessarily constraint-checking queries, as we have seen for VCQs.

To contribute to further analysis of bad, good, and exemplary questions, we created the Repository of Ontology Competency QuestionS (ROCQS) [17] which consists of 558 CQs for all types described in this paper and questions annotated with why they were deemed problematic. ROCQS conforms to the FAIR principles to enhance their usability in the ontology development process: it is publicly available[1] the CQs are linked with the specific ontologies they assess for context and understanding, and the SewerNet use-case is meant to encourage correct usage of CQs. We plan to enable edit access of ROCQS, either through a Wiki or a portal as a front with a database back-end based on the EER diagram snippets presented in Section 3. A sampling of CQs by type is shown in Fig. 6, together with an illustration of problematic questions and why.

## 5. Discussion

This first attempt at disentangling CQs for ontologies and working towards identification of CQ components and attendant entities showed that on the same basis, more types of CQs could be devised, such as the MpCQs, where each of them is plausibly relevant for one or more tasks in the ontology development processes well beyond only scoping and validation. While there were ample SCQs and VCQs readily available, perhaps also thanks to introduction and initial framing of CQs for ontologies as SCQs and VCQs in [5], we needed to design templates and generate CQs for other categories. Which type will have a greater overall impact on ontology development and quality is yet to be investigated as future work.

The application of CQs in the ontology development process, as demonstrated with SewerNet ontology use case, provides guidance that is applicable to various domains. The use of ROCQS may also be facilitated by extending it with more questions than the current largest collection of CQs that has been categorised and evaluated in part, and it is meanwhile part of a larger benchmark collection that is being created[2]. ROCQS may also help to keep track of questions perceived to be easier or harder. Understandability of CQs was noted in [9], in that different wordings or sentence structures of interrogating about the same modelling choices may affect their practical use. Further, it may assist in examining Watson's recent exploration of "worthwhileness" of the information-seeking act of questions [29] and what worthwhile may mean for CQs for ontologies.

Finally, with a better understanding of what the underlying components of CQs are, the CQs may be analysed and generated better. That is, not only an NLP-based analysis of a sentence, but also the meaning behind the words and their categories of elements, which then could be used for more informed prompting, training, or fine-tuning LLMs to generate CQs automatically from scratch or retrofitted based on an existing ontology (e.g., [16, 30, 31, 10, 12, 32]), better manual guidance beyond the CLaRO tool [15], and improved automated generation of candidate axioms in ontology development [33].

[1] https://zubeidaiscyber.github.io/ROCQS/
[2] https://github.com/KE-UniLiv/CQ-benchmark/blob/main/README.md

## 6. Conclusion

The paper presented a novel model for Competency Questions within the context of ontology engineering, QuO, identifying five main types of CQs, divided into ontological CQs and domain CQs: Scoping (SCQ), Validation (VCQ), Foundational (FCQ), Relationship (RCQ), and Metaproperty (MpCQ) questions. Each type serves distinct purposes within ontology development and they each differ in key constituents of the questions, therewith enhancing the clarity and effectiveness of CQs in ontology development. This, in turn, also assists in determining how CQs can be good or faulty. Diverse usage of the CQs at different stages of ontology development was demonstrated with a use-case and additions to the NeOn methodology. To facilitate further research and use of CQs, we extended the ROCQS repository, where the questions are categorised by type, including existing CQ datasets, new sample questions and question templates, and annotated problematic questions.

There are multiple avenues for future work, most notably on the effectiveness of FCQs, RCQs, and MpCQs on ontology quality and usability, and prospects for automated generation and evaluation of such CQs, and how to best integrate QuO in the various ontology development methodologies. We also plan to conduct user-based evaluations of QuO and ROCQS with ontology engineers and domain experts.

## Declaration on Generative AI

The authors have not employed any Generative AI tools.

## References

[1] T. D. Lyon, K. J. Saywitz, D. L. Kaplan, J. S. Dorado, Reducing maltreated children's reluctance to answer hypothetical oath-taking competency questions, Law and Human Behavior 25 (2001) 81–92.

[2] P. Williams, Resourcing for the future? information technology provision and competency questions for school-based initial teacher education, J. of Inf. Techn. for Teacher Ed. 5 (1996) 271–282.

[3] M. C. Suarez-Figueroa, et al., NeOn Methodology for Building Contextualized Ontology Networks, NeOn Deliverable D5.4.1, NeOn Project, 2008.

[4] C. M. Keet, A. Lawrynowicz, Test-driven development of ontologies, in: H. Sack, et al. (Eds.), Proc. of ESWC'16, volume 9678 of *LNCS*, Springer, Berlin, 2016, pp. 642–657.

[5] M. Uschold, M. Gruninger, Ontologies: principles, methods and applications, The Knowledge Engineering Review 11 (1996) 93–136.

[6] E. Thiéblin, O. Haemmerlé, C. Trojahn, Complex matching based on competency questions for alignment: a first sketch, in: 13th WS Ontology Matching (OM'18), CEUR-WS, Monterey, USA, 2018, pp. 66–70.

[7] C. Bezerra, F. Freitas, Verifying description logic ontologies based on competency questions and unit testing, in: ONTOBRAS'17, 2017, pp. 159–164.

[8] C. Bezerra, F. Freitas, F. Santana, Evaluating ontologies with competency questions, in: Proc. of WI-IAT'13 - Vol 03, IEEE Computer Society, 2013, pp. 284–285.

[9] C. H. Bernabé, C. M. Keet, Z. C. Khan, Z. Mahlaza, A method to improve alignments between domain and foundational ontologies, in: Proc. of FOIS'23, volume 377 of *FAIA*, IOS Press, 2023, pp. 125–139.

[10] M.-J. Antia, C. M. Keet, Automating the generation of competency questions for ontologies with agocqs, in: Proc. of KGSWC'23, Springer, 2023, pp. 213–227.

[11] D. Wiśniewski, J. Potoniec, A. Ławrynowicz, C. M. Keet, Analysis of ontology competency questions and their formalizations in SPARQL-OWL, Journal of Web Semantics 59 (2019) 100534.

[12] Z. Mahlaza, C. M. Keet, N. Chahinian, B. Haydar, On the feasibility of llm-based automated generation and filtering of competency questions for ontologies, in: Proc of LDK'25, ACL, 2025, p. (in press).
[13] G. K. Q. Monfardini, J. S. Salamon, M. P. Barcellos, Use of competency questions in ontology engineering: A survey, in: Proc. of ER'23, Springer, 2023, p. 45–64.
[14] J. Potoniec, D. Wisniewski, A. Lawrynowicz, C. M. Keet, Dataset of ontology competency questions to sparql-owl queries translations, Data in Brief (2020) in press.
[15] C. M. Keet, Z. Mahlaza, M.-J. Antia, CLaRO: a controlled language for authoring competency questions, in: Proc. of MTSR'19, volume 1057 of *CCIS*, Springer, 2019, pp. 3–15.
[16] R. Alharbi, V. Tamma, F. Grasso, T. R. Payne, An experiment in retrofitting competency questions for existing ontologies, ArXiv abs/2311.05662 (2023).
[17] Z. Khan, C. M. Keet, On the roles of competency questions in ontology engineering, in: M. Alam, M. Rospocher, M. van Erp, L. Hollink, G. A. Gesese (Eds.), 24th International Conference on Knowledge Engineering and Knowledge Management (EKAW'24), volume 15370 of *LNAI*, Springer, 2024, pp. 123–132. November 26-28, Amsterdam, The Netherlands.
[18] Y. Rebboud, L. Tailhardat, P. Lisena, R. Troncy, Can llms generate competency questions?, in: Extended Semantic Web Conference (ESWC'24), volume 659, Springer, 2024, pp. 227–242.
[19] G. Guizzardi, N. Guarino, Semantics, ontology and explanation, CoRR abs/2304.11124 (2023). arXiv:2304.11124.
[20] P. Fernandes, R. Guizzardi, G. Guizzardi, Using goal modeling to capture competency questions in ontology-based systems, Journal of Information and Data Management 2 (2011) 527–540.
[21] N. Guarino, C. Welty, A formal ontology of properties, in: R. Dieng, O. Corby (Eds.), Proc. of EKAW'00, volume 1937 of *LNCS*, Springer, 2000, pp. 97–112.
[22] N. Guarino, C. Welty, Identity, unity, and individuality: towards a formal toolkit for ontological analysis, in: W. Horn (Ed.), Proc. of ECAI'00, IOS Press, 2000, pp. 219–223.
[23] N. Guarino, G. Guizzardi, Processes as variable embodiments, Synthese (2024) in print.
[24] C. M. Keet, Detecting and revising flaws in OWL object property expressions, in: A. ten Teije, et al. (Eds.), Proc. of EKAW'12, volume 7603 of *LNAI*, Springer, 2012, pp. 252–266.
[25] T. A. Halpin, M. Curland, Enriched support for ring constraints, in: R. Meersman, et al. (Eds.), OTM Workshops 2011, volume 7046 of *LNCS*, Springer, 2011, pp. 309–318.
[26] T. Halpin, Information Modeling and Relational Databases, San Francisco: Morgan Kaufmann Publishers, 2001.
[27] C. M. Keet, B. Haydar, N. Chahinian, The SewerNet domain ontology: on clarifying and harmonising terminology, in: EGU General Assembly 2025, 2025, pp. EGU25–8662. doi:10.5194/egusphere-egu25-8662, Vienna, Austria, 27 Apr–2 May 2025.
[28] C. Chiarcos, M. Sukhareva, Olia–ontologies of linguistic annotation, Semantic Web 6 (2015) 379–386.
[29] L. Watson, Narr Francke Attempto, 2022, pp. 176–182.
[30] R. Alharbi, V. Tamma, F. Grasso, T. R. Payne, A review and comparison of competency question engineering approaches, in: M. Alam, M. Rospocher, M. van Erp, L. Hollink, G. A. Gesese (Eds.), Knowledge Engineering and Knowledge Management - 24th International Conference, EKAW, 2024, Proceedings, volume 15370 of *LNCS*, Springer, 2024, pp. 271–290. 2024, Amsterdam, The Netherlands, November 26-28.
[31] R. Alharbi, V. Tamma, F. Grasso, T. R. Payne, An experiment in retrofitting competency questions for existing ontologies, in: J. Hong, J. W. Park (Eds.), Proc. of SAC 2024, ACM, 2024, pp. 1650–1658. Avila, Spain, April 8-12, 2024.
[32] X. Pan, J. van Ossenbruggen, V. de Boer, Z. Huang, A RAG approach for generating competency questions in ontology engineering, 2025. arXiv:2409.08820.
[33] D. Wisniewski, J. Potoniec, A. Lawrynowicz, SeeQuery: An automatic method for recommending translations of ontology competency questions into SPARQL-OWL, in: G. Demartini, G. Zuccon, J. S. Culpepper, Z. Huang, H. Tong (Eds.), Proc. of CIKM '21, ACM, 2021, pp. 2119–2128. Virtual, Queensland, Australia, November 1 - 5, 2021.