CLaRO: a Controlled Language for Authoring Competency Questions

C. Maria Keet[0000−0002−8281−0853], Zola Mahlaza[0000−0001−9829−1480], and Mary-Jane Antia[0000−0002−9983−6267]

Department of Computer Science, University of Cape Town, South Africa
mkeet,zmahlaza,mjantia@cs.uct.ac.za

Abstract. Competency Questions (CQs) assist in the development and maintenance of ontologies and similar knowledge organisation systems. The absence of tools to support the authoring of CQs has hampered their effective use. The few existing question templates have limited coverage of sentence constructions and are restricted to OWL. We aim to address this by proposing the CLaRO template-based CNL to author CQs. For its design, we exploited a new dataset of 234 CQs that had been processed automatically into 106 patterns, which we analysed and used to design a template-based CNL, with an additional CNL model and XML serialisation. The CNL was evaluated, showing coverage of about 90% with the 93 templates and their 41 variants. CLaRO has the potential to facilitate streamlining formalising ontology content requirements and, given that about one third of the CQs in the test sets turned out to be invalid questions, assist in writing good questions.

1 Introduction

The specification of Competency Questions (CQ) is a step in the process of the development of ontologies and similar artefacts—called "OMS" in [16], for Ontologies, Models and Specifications that also comprises knowledge organisation systems (KOSs). CQs aim to provide insights into the contents of an OMS, to demarcate its scope, and, ideally, are to be used in the verification step during testing of the model [22, 20, 10]. They function alike requirements in the traditional requirements engineering setting, but then are formulated as questions that such an OMS should be able to answer. For instance, Do lions eat grass? that some wildlife ontology may have to be able to answer, Which software can perform clustering? for a structured controlled vocabulary about software, and What are the related terms of propaganda? for the ERIC thesaurus. However, CQs are rarely published at all or in full except in a few cases, notably, [14, 5]. Two main reasons put forward for their low uptake are, firstly, the lack of guidance for formalising them—be this in SPARQL, SPARQL-OWL, OWL or another language—which affects testing of the OMS, and, secondly, the 'free text' nature of CQs makes operationalising them difficult.

A well-known solution direction to such problems is to constrain the natural language so as to streamline the input, which facilitate their formalisation into the desired target logic or query language. A few CQ types, patterns, and "archetypes" have been proposed based on a manual analysis of a small set of CQs [17, 4], which go in the direction of a controlled natural language (CNL) that constrains a full language to a subset of its vocabulary and grammar. However, their 19 resp. 14 patterns are merged with types of ontology elements, therewith constraining its usage to OWL and a particular modelling style, and their adequacy, or coverage, is unknown. Currently, no CNL exists for CQs that has been shown to be adequate in coverage and be at the natural language layer.

In this paper, we seek to address these shortcomings by developing a CNL for CQs. We reuse the novel CQ and CQ pattern dataset of [24] and based on the analysis of the patterns and other design decisions, we convert them into a template-based CNL, called CLaRO: **C**ompetency question **La**nguage for specifying **R**equirements for an **O**ntology, model, or specification. CLaRO is evaluated against a random selection of CQs from the CQ dataset [24] for verification, two newly collected set of CQs that were not part of the training set, and related work. CLaRO's coverage was found to range from good to excellent and substantially outperforming the related work. Overall, this resulted in 93 core templates and 41 variants, which cover about 90% of the CQs of the test sets. We have created a proof-of-concept CQ tool to assist authoring CQs with CLaRO. All data, results, CLaRO, the tool, and a screencast thereof are available as supplementary material at `https://github.com/mkeet/CLaRO`.

The remainder of this paper is structured as follows. Related work is discussed in Section 2. The CNL design and evaluation are described in Sections 3 and 4, respectively. We discuss in Section 6 and conclude in Section 7.

2 Related work

CQs have been proposed for use in several fields, such as education and law (e.g., [23, 12]). In ontology engineering, CQs are deemed important for demarcation of the scope of an ontology and alignment of source and target ontologies [22, 21], and verification and evaluation [2, 1, 3, 10]. In spite of their acknowledged importance, few CQs are available publicly [24].

CNLs for CQ specifically do not exist, but there are four contributions in that direction. Wisniewski et al. [24] recently compiled 234 CQs from 5 ontologies into a freely available dataset[1], and analysed the questions with NLP to chunk it and replace nouns and verbs with variables for entities and predicates, resulting in 106 CQ patterns. Earlier work [4, 6, 17] incorporate ontology elements explicitly, using 1:1 mappings between noun or noun phrase in the CQ and OWL class ("[CE]") and verb and OWL object property ("[OPE]") in an "archetype" that is template-like; e.g., "Which [CE1] [OPE] [CE2]?". However, a CQ, in general, does not need to be for OWL nor does a verb need to become an object property in the ontology. Limited in number of templates, they have limited coverage for CQ patterns; e.g., a simple subclass request, like DemCare_CQ_8.What are the types of diagnosis? [5] has no applicable pattern in [17].

[1] `https://github.com/CQ2SPARQLOWL/Dataset`

A difficulty with CQs is that they may require different formalisations to query the OMS depending on the usage scenario, such as SPARQL-OWL or SPARQL [24, 25] and others. In addition, given that a CNL for CQs is supposed to function for specifying requirements for any ontology, the logic-based knowledge representation must be decoupled from the natural language. At the same time, it is well-known that the other extreme—free-form sentences—is hard to formalise, be this for query or axiom generation; e.g., most recently, Salgueiro et al.'s system allows free-text as input, but only four types of questions may generate answers [19]. A middle way to bridge this gap is to design a CNL.

CNLs for computation have been proposed as a solution for various information management aspects [18, 11], yet all systems surveyed focus on assertions for ontology authoring, even those for queries (e.g., "give me all writers who ..." rather than "which writers...?"). When there are CNL questions, they are for instances in databases or RDF stores, rather than the TBox-level of typical CQs, hence, take a different form, and/or require the ontology to already exist so as to assist in query formulation [7]. Thus, to the best of our knowledge, there is no CNL for CQs for ontologies that is technology- and KOS-independent.

3 CNL design

The approach to the design of the CNL for CQs is a semi-automated and data-driven bottom-up. The input data is taken from the novel dataset of CQ patterns [24]; this is summarised first so as to keep the paper self-contained. We analysed those CQ patterns, which informed the CNL design and specification that is described in Section 3.2.

3.1 Preliminaries: CQ patterns

The automated CQ pattern creation process—described in detail in [24]—uses 234 CQs that were collected from five publicly available CQ sets for publicly available ontologies (SWO, Dam@care, OntoDT, AWO, and Stuff), which is the largest data set of type-level CQs for ontologies. These CQs were used to create *domain-independent CQ patterns*, which are the general structures of the CQs that is shared among more than one CQ and thus irrespective of an ontology's vocabulary. The following process was applied for each CQ in the dataset [24]:

1. *Entity Chunk (EC) and Predicate Chunk (PC) identification*, where ECs contain nouns and noun phrases and PCs are verbs and may contain adpositions/particles and they may have an auxiliary part that may be located in a different place of the question than the main part of PC. This was computed automatically with SpaCy (`https://spacy.io/`) and algorithms developed by [24]. Subsequently, it was manually verified and, if accepted, a sequential number was added to distinguish different ECs and PCs. For instance, the CQ Which country do I have to visit to see elephants? would be chunked into *Which EC1 PC1 I PC1 to PC2 EC2?*, as "do" and "have to visit" belong together, and "see" is a second PC.

2. *Generalisable pattern selection*: The resultant domain-independent form of every CQ was a "candidate pattern". To determine whether it would be added to the list of patterns, it made use of the notions of "dematerialized" and "materialised" CQs:
   - Dematerialised CQ: the CQ has 'replaceable' content already. For instance, SWO's What software can perform task x? is meant to be used such that the user fills in a real task from the ontology for the placeholder "task x", and thus there would be multiple instances of such questions, and therefore it produced a CQ pattern.
   - Materialised CQ: each entity is mentioned in the CQ. If a candidate pattern based on such a CQ was unique, it was rejected as CQ pattern.

This procedure resulted in the list of 106 subject domain-independent patterns.

Analysis of the CQ patterns We analyse these 106 patterns on both structural features and their meanings, such as the use of synonyms, and other aspects that may emerge on closer manual analysis of the patterns.

Considering the anatomy of the CQ patterns: text chunks can appear anywhere in the pattern and have between 0 and 4 text chunks; a pattern has at most 4 EC and 2 PC variables, and an overall of at most 5 variables. Because a PC variable can be split up into different chunks (e.g., a "do we need" is chunked as *PC1 we PC1*), the highest number of slots for the variables is 6. Split PCs either have another variable, text, or a single space between the slots, and there are at most 3 chunks for a PC variable.

There are common sources of variation in the patterns. Illustrative examples for each type of variation are as follows.

1. Singular/plural; e.g., pattern *82. What type of EC1 is EC2?* and *83. What types of EC1 are EC2?*.
2. Superfluous words in the sentence. For instance, the "or not" in *27. Is EC1 EC2 or not?* and "possible" in *51. What are the possible types of EC1?*.
3. Impersonal and personal sentences and patterns. A CQ alike swo37.Can we collaborate with developers of [software x]? could also have been written as, say, "Is it possible to collaborate with developers of [software x]?".
4. Synonym usage in the text chunks, of which there are few; e.g., "kind of" and "type of" are used synonymously in CQs (cf. *79. What kind of EC1 is EC2?* and *82. What type of EC1 is EC2?*).
5. The same information request can be formulated in different ways, such that it would need/use a different pattern; e.g., the CQ swo15.What software can I use [my data] with to support [my task]? can be rewritten as, e.g., "Which software can use [my data] to support [my task]?" as well as "Which software can support [my task] with [my data]?" and corresponding different patterns.

Personal pronouns only appear in the SWO CQ set, "kind of" appears only in the AWO and Stuff CQ sets that were authored by the same author, "type of" appears only in the SWO CQ set, and "types of" appears only in the Dem@Care CQ set. This suggests there might be either author preference or some (un)conscious authoring choice to generate more questions in the same way.

Finally, negation—in the sense of both disjointness among classes and for a class' properties—is present in the CQs, but only once each and thus did not result in a pattern in [24]. It appears in Ren et al.'s and Bezerra et al's Pizza example CQs, but not in the original Pizza CQ set. Nonetheless, one may expect also negative CQs to be posed.

3.2 The CLaRO CNL

Design considerations There are two extreme design options for a CNL, which is often template-based: 1) minimalist with the fewest amount of templates that are shortest and 2) including variants to allow flexibility and have better flowing text. The latter approach has been proposed elsewhere for a CNL for temporal conceptual modelling [9], which was based on a user evaluation on template preferences. Also, since different formulation habits were detected in the dataset, we will keep all CQ patterns and convert them into templates, but also generate a 'default' CQ template, where applicable. Because there is a limited number of CQ patterns, a template-based approach will be taken for the CNL at this stage, rather than specification of a grammar.

Finally, while there is no negation in any of the patterns of [24], there is in the CQ set and elsewhere; therefore, we deem it reasonable to add a few templates to cover these cases. Even though that hiding the negation makes it less cumbersome for a CNL, it will make it harder for processing it automatically into a query over the resource, whereas it is an easy signal in a template.

Specification The generation of the 'default' templates applies to those CQ patterns of [24] where there were issues or commonalities regarding, mainly: 1) singular/plural forms, 2) the personal pronouns in a pattern, 3) removing redundant words in text chunks, and 4) synonym usage. To illustrate some of these changes (see github repo for details), consider CQ pattern *1.Are there any EC1 for EC2?*: it is in the plural and has the redundant "any" word, which therefore results in a template of *Is there [EC1] for [EC2]?*, which turned out to be identical to CQ pattern 30, and thus removed so as to obtain a list of unique sentences. Regarding synonyms, 'type of' was selected over 'kind' and 'category' for the defaults. This resulted in the merger of, e.g., CQ pattern 79 and 82 into template *70.What type of [EC1] is [EC2]?*.

Applying these transformations manually and removing any duplicates that were generated during this process resulted in 89 templates with 40 variants, where the variants have an additional letter designation; e.g., *22.Is [EC1] [EC2]?* and variant *22a.Is [EC1] [EC2] or not?*. Fourteen of the 89 templates are fragments of others; e.g., *22.Is [EC1] [EC2]?* is a template fragment of *23.Is [EC1] [EC2] for [EC3]*. This may be of interest to further reduce the number of templates as well as be of interest for a predictive editor in tool design.

To cater for the negations, three basic templates were attached, so as to cover the cases of 'does not PCi', 'PCi no ECi', and class disjointness (numbers 90-92).

Storing templates and CQs While CQ templates can be stored in a simple txt file, it serves to store them in a structured way so that multiple tools can use and analyse them in the same manner. To the best of our knowledge, there is no standard for storing a CNL. Therefore, we designed our own data model for storing CQ templates, which is depicted in Fig. 1 in UML Class Diagram notation. To permit extensions to CLaRO, there may be CQs that do not instantiate a template (hence, the 0..* on instantiates). Also, users should be permitted to author CQs without an ontology being present already, as this activity may happen before the OMS development; hence, the 0..* on the for association end.

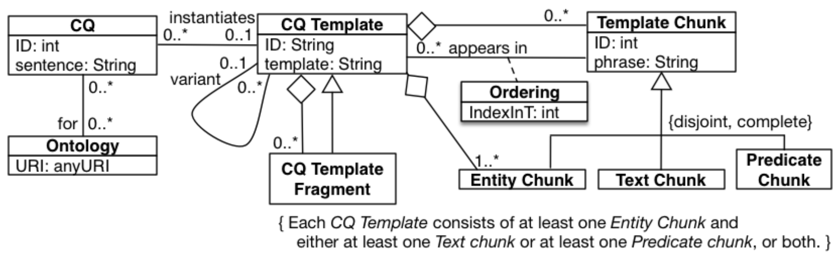

Fig. 1. Data model for storing CQ templates.

4 Evaluation

We conduct an evaluation of CLaRO to answer the following two questions:

RQ1: Does CLaRO cover the CQs from the training set?
RQ2: Is CLaRO sufficiently comprehensive for unseen CQs?

CLaRO should be able to deal with the CQs of the original data set of [24], but may not, because not all CQs resulted in a pattern. Also, the CQ patterns were obtained automatically and a verification was not performed. In addition, for the time being that there is no advanced CQ tool, authors will author a question manually and thus may need to do the chunking of a CQ themselves.

The second question aims to assess whether CLaRO provides a broad enough coverage of possible sentence templates to be adequate beyond the training data.

Finally, we compare CLaRO to the templates of Ren et al. and Bezerra et al..

4.1 Design

Methods To answer Question 1, we take a random selection of 10% of the CQs in the dataset and test them on authorability with the CNL. This set is called SetA. Each sentence is manually chunked into ECs and PCs by one of the authors and then checked against CLaRO's templates. For each CQ in SetA, record whether it can be authored in the CNL and, if not, why not, then compute percent coverage. Afterward, the manual chunking was compared against the mapping of CQs to CQ patterns that was kindly provided by D. Wisniewski.

To answer Question 2, we collect a new set of CQs that are at least for a different ontology, are authored by people other than those who authored the CQs in the data set, and are ideally also in a different domain. The target is 20 type-level CQs. This set is called SetB. A second test set, SetC, is created from half of the Pizza ontology CQs so that it is about the same size as SetB; they are kept separate, as there is some overlap in CQ authors of the SWO and Pizza CQs. For each CQ in SetB and SetC, record whether it can be authored in the CNL. If it cannot be authored directly, attempt to manually reformulate it into a sentence with equivalent meaning that does fit with one of the templates. Compute percent coverage for both the original set and the set with reformulations (if any). Compare the outcomes of SetA, SetB, and SetC.

The comparison with Ren et al. and Bezerra et al.'s templates is two-fold. First, we compare their respective templates to the CLaRO templates, with the alignment that their CE maps to CLaRO's EC and their OP/OPE/DP to CLaRO's PC. Second, from this comparison follows at least part, if not fully, the coverage of their template sets for the CQs in SetA, SetB, and SetC. If there is no equivalent template, then Ren et al.'s, respectively, Bezerra et al.'s template, is checked against the CQ in question, and tested against the CQs for which CLaRO does not have a fitting template (if applicable). Second, assess the coverage of their template sets for the CQs in SetA, SetB, and SetC.

Materials To construct SetA, we take every 10th CQ from the list of [24], being: swo01, ... swo81, stuff_03, awo_2, awo_12, DemCare_CQ_9, ... , DemCare_CQ_99, ontodt_02, and ontodt_12, resulting in a set of 24 CQs. For SetB, we assess CQs from a recent [15] and a related paper [25] and filling it up to 20 with the CQ set of the Vicinity project[2]. The scopes of the ontologies that the CQs relate to are at least partially different from those in the dataset and, to the best of our knowledge, there is no overlap in CQ authors. The CQs are different from those in Wisniewski et al. [24]'s CQ set. The Pizza ontology CQs are sourced from R. Stevens' slides[3]. Every other CQ in the list is selected, resulting in 21 sentences. The templates of Bezerra et al. and Ren et al. are taken as published in [4, 17].

4.2 Results and Discussion

Verification with training set CQs Manual chunking of the CQs in SetA and testing against CLaRO yielded a 70.8% initial success rate. Further analysis on alternative ways of chunking the sentences increased it to 83.3%. The four remaining cases demonstrate challenges with bottom-up approaches to designing a CNL. For instance, we chunked swo11.Which visualisation software is there for [this data]

[2] `http://vicinity.iot.linkeddata.es/vicinity/`; last accessed: 20 Dec. 2018.

[3] page 4 of `http://studentnet.cs.manchester.ac.uk/pgt/2014/COMP60421/slides/Week2-CQ.pdf`; last accessed: 9-1-2019.

and what will it cost? as `Which [visualisation software]`$_{EC1}$ `is there for [this data]`$_{EC2}$ `and what [will]`$_{PC1}$ `[it]`$_{EC1}$ `[cost]`$_{PC1}$`?`, because of the referring expression "it" that the automated chunker had not recognised and allocated the consecutive EC3 variable label. Another issue is the (mis)use of 'What..." vs "Which..." in the question formulation that affected first-hit matching but passed after grammar correction.

The automatically chunked CQs had a 91.7% initial success rate, and 100% upon further analysis. The two that failed initially, DemCare_CQ_29 and DemCare_CQ_89, were sentences with unique sentence structures in the CQ set, and therefore did not qualify to become a CQ pattern, hence, did not enter CLaRO as such. The manual chunking of DemCare_CQ_89—different from the way the algorithm had done it—did match template 42.

Overall, RQ1 can thus be answered in the affirmative, but noting the challenges to chunk it in the 'right' way.

Coverage of CLaRO The results for SetB are mixed. 25% of the sentences were not CQs for OMSs, such as hero5.Why universities are organized into departments? Of the remaining 15, five had a direct match with a CLaRO template. Three more matched after rewording the plural into the singular (What are ... into What is ...) and a grammar rephrasing. Given that the 'What are/is' also appeared in SetA, hence, twice now, we add the following template as variant to CLaRO: *60a. What are [EC1] of [EC2]?*.

The remaining seven did not have a match, of which six would match a template of *What is [EC1]?*, such as vic1.What is an organization?. Such simple definition request CQs appeared only once as a materialised CQ in the original dataset (DemCare_CQ_4). When we extend CLaRO with the template *93. What is [EC1]?*, then the coverage for SetB is 93.3% out of the 15 valid CQs (splitting up CQ hero3, it reaches 100% of the valid CQs).

Nine sentences of SetC were invalid as CQ, due to, among others, imperatives and an extra-ontological modelling discussion question. Of the remaining 12, four were successfully matched in the first round and six more after rewording, reaching 83.3% coverage. Rewriting the imperatives into questions, all five passed immediately, totalling to a coverage of 88.2% of the valid CQs in SetC.

Overall, CLaRO's 134 templates can process unseen CQs with a good level of coverage, thereby answering RQ2 in the positive. Given that 34.1% of the questions in SetB and SetC turned out not to be CQs for ontologies and the different levels of coverage for SetB and SetC, this evaluation has to be considered preliminary. The percentage of improper CQs suggests that a CNL for CQs may be a welcome addition, so that CQ authors may be encouraged more to write grammatically better and answerable questions.

Comparison of CLaRO *against related work* Regarding Ren et al.'s 19 templates, one is not a question (R1b. "Find [CE1] with [CE2]."), two match after rewriting the template into grammatically correct English (from "Be there ..." into "Is there..."), and two are ambiguous of which one does not have a match. The one that does not match, R12. "Do [CE1] have [QM] values of [DP]?", is based on a

Table 1. Aggregate results for coverage of the test sets; best values of the comparison are highlighted in italics. The CLaRO data is for the complete set of 134 templates.

		SetA	**SetB**	**SetC**	**Combined**
	|Total CQs|	24	20	21	65
	|Valid CQs|	24	15	12	51
Match	Ren et al.	6	5	6	17
	Bezerra et al.	3	3	4	10
	CLaRO	*20*	*14*	*11*	*45*
Pct. coverage (valid CQs)	Ren et al.	25	33	50	33
	Bezerra et al.	13	20	33	20
	CLaRO	*83*	*93*	*92*	*88*

CQ “Do pizzas have different values of size?”, which is not in the Pizza QC set. The Pizza CQ set does have “Do pizzas come in different sizes?”, which can be chunked into *PC1 EC1 PC1 EC2?*, which does match template number 29.

Three of Bezerra et al’s 14 templates do not have a matching CLaRO template. The first one, B3 “From which + <property> + <class>?” is based on the sample sentence “From which nation is American pizza?”, which is not in the Pizza CQ set and with that sample sentence, the template should have had two classes. B10, also has a pizza example, but it is also not in the Pizza CQ set. The B9 mismatch is a variant with negation, “Are + <class> + <class>disjoint?”, which can be reworded into the disjointness template 92 repeatedly.

While CLaRO does not fully encompass the other two sets of templates, it substantially outperforms them on coverage for actual CQs; as can be seen from the aggregate data included in Table 1.

5 CQ authoring tool

We developed a tool to aid domain experts and CQ authors in writing questions so that they do not have to start from scratch.

The main components of the tool are the user interface, template processing module, and storage module. The user interface is responsible for accepting the user’s input, displaying user-friendly template suggestions, and listing all the user-defined CQs. The template processing module is responsible for generating possible template suggestions given some user input and associating the final user input with a CLaRO template. When the user provides input, the system suggests a set of user-friendly forms of CLaRO templates, which are generated within the autocomplete module by replacing all instances of the numbered abbreviations ECi and PCi (for $i \in \mathbb{N}$) with the English full form “noun phrase” and “verb phrase”, respectively, from CLaRO’s templates. For instance, CLaRO’s template 1 is transformed from *Is there [EC1] for [EC2]?* to “Is there [noun phrase] for [noun phrase]?”. The auto-complete function filters out non-relevant suggestions among the set of all possible ones for some given user input. A suggestion is considered relevant by the tool either if it starts with or contains the user input, which is configurable in the tool. For instance, when the user types "What type", then templates 70, 70a, and 71 are retrieved and rendered in their user-friendly form. Once the user selects a suggestion, they can edit the verb and noun phrase slots to finalise the CQ. They can also edit the selected template and write a question that does not fit within any CLaRO template. The CQs and their corresponding CLaRO templates, if any, are then saved to disk. The storage module is responsible for loading CLaRO templates from disk and loading/saving the user defined questions to disk. The storage module serialises the set of user defined CQs according to an XML schema that implements the model described in Section 3.2.

Since the CQs may be created also for artefacts similar to ontologies (e.g., thesauri) and ontologies not formalised in OWL, we deemed it best to create a stand-alone tool that is not tightly coupled with an existing KOS editor. The source code and a screencast of the tool are available as supplementary material.

6 Discussion

CLaRO is, to the best of our knowledge, the first CNL for competency questions for ontologies, surpassing the previously published archetypes and patterns [4, 13, 17] principally on the following aspects: i) decoupling of the language and cognition from the ontology artefact layer where design decision already have been taken, ii) more variants in sentences structures to accommodate for several question formulation preferences, and therewith iii) better coverage for CQs.

Trying to find new CQs was a non-trivial endeavour, and of those we could find that were listed as CQs, it turned out that about a third of the questions were invalid as CQ. It is unclear what the main reason for that is, but it is certainly clear that CLaRO can assist with reducing that percentage for newly created CQs. Wisniewski's et al.'s dataset [24] does not have invalid CQs, which means they either have been curated upfront (not described to be the case in [24]), or all the good CQ sets available went into that dataset.

It was expected that the Pizza CQs (SetC) would yield a higher percentage of coverage than the other newly sourced CQs (SetB), due to the overlap in people involved in Pizza and SWO. This turned out to be the case in the strict sense: the original coverage for SetB before adding template 93 to CLaRO was 53.3% whereas for SetC it was 83.3%. With the required manual interventions—a new template and rephrasing the imperatives—this increased the coverage to 93.3% and 88.2%, respectively, which is similar.

The model for storing the CQ templates (Fig. 1) may appear straightforward. To the best of our knowledge, however, there is no other model as precursor to an XML schema for storage of any CNL, even though there are template-based CNLs that are stored in XML notation. This model, therefore, may contribute toward the development of a *de facto* standard for storing template-based CNLs, not only for CQs, but generally for any CNL. This then may perhaps be wrapped in an extended version of, e.g., the NIF for NLP tool exchange of text annotations [8] when linked to a chunker for analysis free text CQs.

While the templates of CLaRO cover more sentence structures than the earlier proposed patterns and archetypes, the evaluation also has shown that more sentence structures may be possible than currently are covered with CLaRO. Therefore, the CLaRO editor allows also new free-form CQs. A planned extension is to have the editor learn from the input given. Also, it could be integrated in ontology authoring methods, such as TDD [10] and iterative development [13], and other activities, such as CQs for validation [25] and alignment [21].

7 Conclusions

The paper presented the, to the best of our knowledge, first Controlled Natural Language for competency questions: **C**ompetency question **La**nguage for specifying **R**equirements for an **O**ntology, model, or specification (CLaRO). It was designed in a bottom-up way, availing of a new dataset of questions and their patterns. Those patterns were analysed, and converted into a template-based Controlled Natural Language, CLaRO. The language was evaluated with questions from the training set and a small new set of competency questions, which demonstrated good to excellent coverage. Overall, the process resulted in 93 core templates and 41 variants, which cover over 90% of the CQs of the test sets. CLaRO also comes with a basic CQ authoring tool, where the CQs are stored in XML format for possible further processing.

We are currently working on an intelligent editor for CLaRO in order to offer more effective software-support for authoring competency questions, such as automated chunking and self-learning of new templates.

References

1. Azzaoui, K., Jacoby, E., Senger, S., et al.: Scientific competency questions as the basis for semantically enriched open pharmacological space development. Drug Discovery Today **18**(17), 843 – 852 (2013)
2. Bezerra, C., Freitas, F.: Verifying description logic ontologies based on competency questions and unit testing. In: ONTOBRAS. pp. 159–164 (2017)
3. Bezerra, C., Freitas, F., Santana, F.: Evaluating ontologies with competency questions. In: Proc. of IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT) 2013. pp. 284–285. IEEE Computer Society, Washington, DC, USA (2013)
4. Bezerra, C., Santana, F., Freitas, F.: CQChecker: A tool to check ontologies in OWL-DL using competency questions written in controlled natural language. Learning & Nonlinear Models **12**(2), 4 (2014)
5. Dasiopoulou, S., Meditskos, G., Efstathiou, V.: Semantic knowledge structures and representation. Tech. Rep. D5.1, FP7-288199 Dem@Care: Dementia Ambient Care: Multi-Sensing Monitoring for Intelligence Remote Management and Decision Support, `http://www.demcare.eu/downloads/D5.1SemanticKnowledgeStructures_andRepresentation.pdf`
6. Dennis, M., van Deemter K., Dell'Aglio, D., Pan, J.Z.: Computing authoring tests from competency questions: Experimental validation. In: The Semantic Web – ISWC 2017. LNCS, vol. 10587, pp. 243–259. Springer (2017)

7. Hallett, C., Power, R., Scott, D.: Composing questions through conceptual authoring. Computational Linguistics **33**(1), 105–133 (2007)
8. Hellmann, S., Lehmann, J., Auer, S., Brümmer, M.: Integrating NLP using linked data. In: The Semantic Web – ISWC 2013. pp. 98–113. LNCS, Springer (2013)
9. Keet, C.M.: Natural language template selection for temporal constraints. In: CREOL, JOWO'17. CEUR-WS, vol. 2050, p. 12 (2017), 21-23 Sept 2017, Bolzano, Italy
10. Keet, C.M., Lawrynowicz, A.: Test-driven development of ontologies. In: Proc. of ESWC'16. LNCS, vol. 9678, pp. 642–657. Springer (2016)
11. Kuhn, T.: A survey and classification of controlled natural languages. Computational Linguistics **40**(1), 121–170 (March 2014)
12. Lyon, T.D., Saywitz, K.J., Kaplan, D.L., Dorado, J.S.: Reducing maltreated children's reluctance to answer hypothetical oath-taking competency questions. Law and Human Behavior **25**(1), 81–92 (Feb 2001)
13. Malheiros, Y., Freitas, F.: A method to develop description logic ontologies iteratively based on competency questions: an implementation. In: ONTOBRAS. p. 142–153 (2013)
14. Malone, J., Brown, A., Lister, A.L., Ison, J., Hull, D., Parkinson, H., Stevens, R.: The software ontology (swo): a resource for reproducibility in biomedical data analysis, curation and digital preservation. J. Biomed. Sem. **5**(1), 25 (Jun 2014)
15. Moreira, J., Pires, L.F., van Sinderen, M., Daniele, L.: Saref4health: IoT standard-based ontology-driven healthcare systems. In: Proc. of FOIS'18. FAIA, vol. 306, pp. 239–252. IOS Press (2018)
16. Mossakowski, T., Codescu, M., Neuhaus, F., Kutz, O.: The Road to Universal Logic–Festschrift for 50th birthday of Jean-Yves Beziau, Volume II, chap. The distributed ontology, modelling and specification language - DOL. Studies in Universal Logic, Birkhäuser (2015)
17. Ren, Y., Parvizi, A., Mellish, C., Pan, J.Z., van Deemter, K., Stevens, R.: Towards competency question-driven ontology authoring. In: Proc. of ESWC'14. LNCS, Springer (2014)
18. Safwat, H., Davis, B.: CNLs for the semantic web: a state of the art. Language Resources & Evaluation **51**(1), 191–220 (2017)
19. Salgueiro, A.M., Alves, C.B., Balsa, J.: Querying an ontology using natural language. In: PROPOR 2018. LNAI, vol. 11122, pp. 164–169 (2018)
20. Suarez-Figueroa, M.C., de Cea, G.A., Buil, C., et al.: NeOn methodology for building contextualized ontology networks. NeOn Deliverable D5.4.1, NeOn Project (2008)
21. Thiéblin, E., Haemmerlé, O., Trojahn, C.: Complex matching based on competency questions for alignment: a first sketch. In: 13th International Workshop on Ontology Matching (OM'18). pp. 66–70. CEUR-WS, Monterey, US (2018)
22. Uschold, M., Gruninger, M.: Ontologies: principles, methods and applications. Knowledge Engineering Review **11**(2), 93–136 (1996)
23. Williams, P.: Resourcing for the future? information technology provision and competency questions for school-based initial teacher education. J. of IT for Teacher Ed. **5**(3), 271–282 (1996)
24. Wisniewski, D., Potoniec, J., Lawrynowicz, A., Keet, C.M.: Competency questions and SPARQL-OWL queries dataset and analysis. Technical Report 1811.09529 (November 2018), `https://arxiv.org/abs/1811.09529`
25. Zemmouchi-Ghomari, L., Ghomari, A.R.: Translating natural language competency questions into SPARQL queries: a case study. In: First Int. Conf. on Building and Exploring Web Based Environments. pp. 81–86. IARIA (2013)