Use of Competency Questions in Ontology Engineering: A Survey

Glaice Kelly Q. Monfardini[1,2](✉), Jordana S. Salamon[1], and Monalessa P. Barcellos[1]

[1] Ontology and Conceptual Modeling Research Group (NEMO), Computer Science Department, Federal University of Espírito Santo, Vitoria, Brazil
{jssalamon,monalessa}@inf.ufes.br
[2] Federal Institute of Espírito Santo, Vitoria, Brazil
glaice.monfardini@ifes.edu.br

Abstract. The interest in the ontology subject has grown in recent decades. Ontologies can be used to assign semantics to information items and solve interoperability and knowledge-related problems. Many methods have been proposed to improve the ontology engineering process. The use of competence questions (CQs) is suggested by several of them as a means to define the ontology requirements and help identify the necessary concepts, properties, and relations. CQs are questions that the ontology should be able to answer. Thus, they provide a mechanism to verify if the ontology is in accordance with the established requirements and properly represents the desired knowledge. Despite the important role of CQs, there is a lack of deeper investigation to provide evidence about their use. Therefore, aiming to investigate how CQs have been used in ontology engineering practice, we performed a survey with 63 ontology engineers. The results indicate that CQs have helped mainly to define the ontology scope and evaluate the ontology conceptualization. However, ontology engineers still face difficulties when writing, using, and managing CQs. Although there is a range of methods and tools that support ontology development, guidance regarding CQs is still limited. This paper presents our study and discusses its main findings.

Keywords: Competency Questions · Ontology · Survey

1 Introduction

An ontology is a formal representation of a common conceptualization of a universe of discourse [17]. Ontologies have been a useful instrument for reducing conceptual ambiguities, making knowledge structures transparent, supporting knowledge sharing and interoperability between systems [49]. They have been successfully used in several domains, such as IT Service Management [30], Health [38], Education [52], and Software Engineering [4]. Nowadays, ontology engineers are supported by a wide range of ontology engineering (OE) methods and tools. However, building ontologies is still a complex task even for experts [27].

J. P. A. Almeida et al. (Eds.): ER 2023, LNCS 14320, pp. 45–64, 2023.
https://doi.org/10.1007/978-3-031-47262-6_3

To assist ontology engineers in the ontology development process, ontology engineering methods break it into other processes and recommend activities for each one [20]. Although methods differ in many aspects, they often include a process or activity addressing requirements specification, when the ontology scope, intended uses, users, and competence are established [50]. Some methods suggest defining the ontology requirements by means of competency questions (CQs) specified in natural language (e.g., [9,28,31,43]). CQs encompass the purpose of the knowledge base and suggest the concepts and relationships to be included in the ontology [2]. They can also be used in later stages of ontology development to verify and validate the knowledge represented by the ontology aiming to ensure that the ontology correctly reflects the real world [15].

Despite the important role CQs play in ontology engineering, there is a lack of consensual and detailed guidance on how to identify, write, and use them [1,50]. Even works that address how to specify ontology requirements (e.g., [1,11,33,44, 50]), point out that it is still necessary to deepen studies about CQs. This can lead to doubts, contradictions, oversights, and ambiguities when defining CQs. Not discovering and properly defining CQs early in ontology development may result in a poorly specified ontology, increasing the time and effort spent in the development process [1,10] and hampering ontology quality [10,11].

Exploring CQs in ontology development is not a new idea itself. However, a broader spectrum of CQs and their utility in ontology engineering has not been investigated in depth [36]. To take a step in this direction, we decided to investigate the use of CQs in ontology engineering practice. For that, we performed a survey with 63 ontology engineers aiming to understand how they have used CQs when developing ontologies, the perceived benefits, and faced difficulties. The results provide a preliminary panoramic picture of the state of practice of the use of CQs in ontology engineering. With this panorama, we intend to share practices and perceptions with other ontology engineers and shine a light on research opportunities to provide advances in the research topic.

In summary, the results corroborate statements from the literature (e.g., [9,12,15,28,33,43]) by showing that CQs have been used mainly in requirements specification and ontology evaluation and, thus, help define the ontology scope and evaluate the ontology conceptualization. Moreover, most of the time they have been defined iteratively and refined along the ontology development process. Time constraints have been the main reason for not using CQs and there is a lack of supporting tools. Furthermore, guidance on how to define and use CQs is still limited, which causes ontology engineers to face difficulties when writing, using, and managing CQs.

In this paper, we provide an overview of our study, summarize the main findings, and discuss the results. It is organized as follows: Sect. 2 presents the theoretical background for the paper; Sect. 3 presents the study protocol; Sect. 4 synthesizes the results; Sect. 5 discusses the results; Sect. 6 addresses the study limitations; and Sect. 7 presents our final considerations.

2 Ontologies and Competency Questions

An ontology is a formal and explicit specification of a shared conceptualization [41]. The conceptualization is an abstract and simplified view of the world which is intended to be represented for some reason. Every knowledge base, knowledge-based system, or knowledge level agent is committed, either explicitly or implicitly, with one conceptualization [45].

Ontologies have been widely used in several domains in applications related to knowledge management, natural language processing, intelligent integration information, information retrieval, database design, among others [6], and have become the predominant way to deal with semantic aspects in semantic integration initiatives [23]. They can solve or minimize problems related to communication between people, organizations, and systems by eliminating or reducing the lack of knowledge of the concepts involved in communication processes [48].

An important distinction sets apart ontologies as conceptual models, called *reference ontologies*, from ontologies as computational artifacts, called *operational ontologies* [18]. A reference ontology is constructed with the goal of making the best possible description of the domain in reality, representing a model of consensus within a community, regardless of its computational properties. Operational ontologies, in turn, are machine-readable ontologies designed with the focus on guaranteeing desirable computational properties [9].

The literature presents several OE methods (e.g., [3,9,12,28,31,43,48]). In general, developing ontologies involves management, development, and support activities. The first covers the organizational setup of the overall process (e.g., managing resources, controlling the project schedule, and the quality of the produced artifacts). The second refers to ontology development itself and includes activities such as ontology specification, conceptualization, formalization, and implementation. The third involves activities related to knowledge acquisition, documentation, and configuration management, among others, which are carried out in parallel with development activities to support them [7].

When designing an ontology, requirements can be captured through CQs. They play a key role, consisting of a set of questions that the ontology to be built should be able to answer [13]. By establishing CQs, we reach an effective way to determine what is relevant to the ontology and what is not. They define the ontology scope and provide a way for evaluating the ontology [9]. Therefore, CQs can be used to support both, ontology specification and ontology evaluation. In the former, CQs help model the domain, i.e., through questions that the ontology should be able to answer, it is possible to have a notion of which are the relevant concepts of the domain and the relationships between them. In the latter, CQs can be used to identify ontology flaws in domain modeling, and thus contribute to the ontology quality assessment [48].

CQs can be informal or formal [16]. Informal CQs are expressed in natural language and do not require knowledge of Descriptive Logic, facilitating its use by people unfamiliar with it. They connect the proposed ontology to its application scenarios, thus providing an informal justification for the ontology. Formal CQs, in turn, are expressed in formal language and are created from informal questions by using axioms and the ontology terminology [16].

CQs can be identified from different sources, using different strategies, and can be written in different granularity levels. When defining CQs, the ontology engineer can start with complex questions that are decomposed into simpler ones (top-down approach) or with simple questions that are composed to create complex ones (bottom-up approach). The ontology engineer can also start just writing down important questions that are composed or decomposed later on to form abstract and simple questions, respectively (middle-out approach) [42]. Simple CQs are important for deriving test cases, while complex and more abstract CQs are important to guide ontology modularization [9].

3 Study Design

A survey aims at identifying the characteristics of a broad population by generalizing on the data collected from a representative sample of individuals [8]. It is conducted to produce a snapshot of the situation to capture the current status [51]. We chose this method because, as we aimed at a panoramic view, we needed to reach several ontology engineers and ask about many practices. We followed the process defined in [51], which comprises five activities:*scoping*, when we scope the study problem and establish its goals; *planning*, when the study design is determined; *operation*, which consists in collecting data; *analysis and interpretation*, which involves analyzing data to get conclusions about the research topic; and *presentation and package*, when the results are communicated.

The study **goal** was to investigate the use of CQs in OE practice. Aligned with the study goal, we defined the following two main **research questions**: (RQ1) How have CQs been defined, used, and supported in OE? (RQ2) Which benefits and difficulties have been perceived?

The **instrument** used in the study was a form created by using Google Forms. It contains a consent term for participation in the study and two sections of questions. The first has five closed questions to characterize the participants. The second has 16 closed questions related to the study research questions. Three of them allow the participant to complement the answer by providing further information in text format. There is also an open question for collecting comments and suggestions. In some questions referring to the frequency in which the participants perform some practices, we used a scale based on the Likert scale but excluded the neutral option in order to obtain meaningful information. The form used in the study is available in the study package [25].

The **participants** must be a sample of the target population. Thus, we aimed at ontology engineers with knowledge of and experience in OE and CQs.

The **procedure** followed in the study consisted of three steps. In the first, we ran a small pilot to evaluate the form and the study protocol. We asked two ontology engineers with experience in OE and CQs to answer the questionnaire and report problems, suggestions, and response time. Based on their feedback, we made minor adjustments in the form. In the second step, we sent messages inviting people to participate in the study. The messages were sent to research groups that work with ontologies, mailing lists involving OE researchers and practitioners, contact networks in universities, public, and private organizations, and authors of the papers selected in an ongoing systematic literature mapping about CQs we are conducting. We also asked the invitees to forward the invitation to other people they thought could participate in the study. The final step consisted of gathering data from the answered questionnaires, representing data in tables and graphs, and analyzing them.

4 Study Execution and Data Synthesis

The invitation was sent in late March and early April 2023. We contacted 115 people and received 65 answers until May 1st, 2023, which amounts to a response rate of 56%. Two respondents declared in the questionnaire that they have never used CQs when developing ontologies. As they did not have a suitable profile for participating in the study, we removed their answers from the sample, resulting in 63 participants. In this section, we summarize data collected in the study. For questions in which the participants could choose more than one answer, the sum of the absolute values is higher than 63 and, thus, the sum of the rates exceeds 100%. For simplification reasons, we rounded the percentage values to the first decimal place. The complete set of collected data plus tables and graphs representing them is available in the study package [25].

Most of the participants are Brazilians (32; 50.8%). Other participants are from Spain (9; 14.3%), Germany (5; 7.9%), Netherlands (4; 6.3%), Malaysia (3; 4.8%), Italy (2; 3.2%), United States (1; 1.6%), Mexico (1; 1.6%), Argentina (1; 1.6%), Belgium (1; 1.6%), South Africa (1; 1.6%), Uganda (1; 1.6%). Two participants did not inform where they were from. The participants' profile was identified through questions regarding the context in which they have worked with ontologies, how long they have worked with OE, how they have acquired knowledge of the subject, and their experience level with CQs. Figure 1 summarizes data about the participants. As can be noticed, most participants have worked in the academic context (85.7% in total, 63.5% exclusively in this context), have worked with ontology development for five or more years (68.3%), and have high or very high experience in CQs (58.7%). Knowledge of the subject has been acquired in several ways and most of the participants have learned from different sources – mainly masters/PhD courses (71.4%), scientific events (60.3%), and searching by themselves (60.3%).

In the following, we synthesize collected data by grouping questions into six topics: Ontology development, Use of CQs in OE, Ways of developing CQs, CQs supporting tools, CQs management, and Benefits and difficulties of using CQs. Questions related to the five first topics aimed to get data to answer the research question RQ1. Questions related to the last topic are aimed to answer RQ2.

Ontology Development: The way ontology engineers develop ontologies can influence the use of CQs. Thus, to identify how ontology engineers have developed ontologies, we asked them about the ontology types they have developed, how

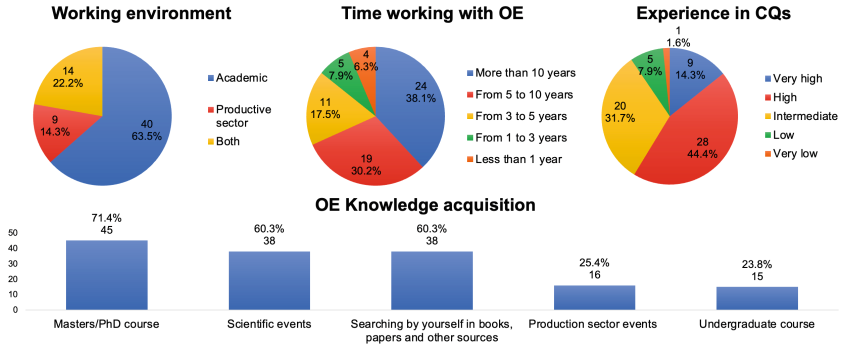


Fig. 1. Participants profile

often they have used OE methods, and which ones they have used (Uschold and King [47], TOVE [16], METHONTOLOGY [12], Ontology 101 [28], NeOn [43], SaBiO [9], XD [3], LOT [33], or others). Figure 2 shows the results. Most participants (54%) have developed both, reference and operational ontologies. Only 6.3% of the participants have developed operational ontologies exclusively, while 39.7% have focused on reference ontologies. Most of the participants (90.5%) have often used OE methods when developing ontologies (always or most of the time). The used methods are diverse, with a predominance of SABiO [9] (47.6%), METHONTOLOGY [12] (25.4%), and NeOn [43] (23.8%). Some participants (17.5%) have adapted existing methods and a few (3.2%) have combined different methods. Moreover, three participants (4.8%) chose the option "Other" and declared the use of Ontokem [46], OBO Foundry [40], and Modular Ontology Modeling (MOMo) [39](each one cited by one participant).

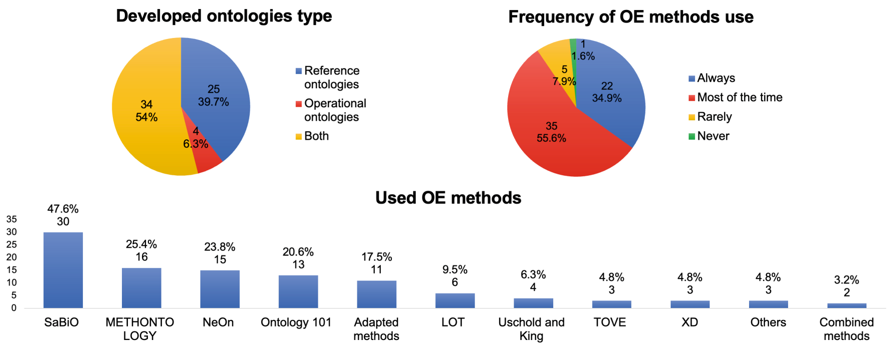


Fig. 2. Ontology development

Use of CQs in OE: To understand how the participants have used CQs when developing ontologies, we asked how often and in which phase of the OE process they have used them. Figure 3 shows the results. Most participants (32;

50.8%) declared that they have always used CQs when developing ontologies. For those who answered that have not always used CQs (34.9% indicated that have used them most of the time and 14.3% rarely), we asked what has caused them not to use CQs. Most of them (27%) pointed out time constraints. Some participants indicated that do not always use CQs because do not think they are necessary (12.7%) or because they find them difficult (11.1%). Reasons informed by participants who selected the "Others" option (9.5%) were a lack of understanding of how the CQs aid in the OE process, the type or purpose of the ontology, difficulties in writing CQs, and keeping track of them while discussing the domain with domain experts. Concerning when the participants have used CQs, we asked them to indicate the ontology development phases and provide information about how they have used CQs in each phase they selected. Most of the participants (90.5%) have used CQs in Requirements Specification, to represent functional requirements, define the ontology scope, and capture knowledge the ontology needs to represent. 68.3% have used CQs in Verification, Validation & Testing, to guide these activities by checking how the ontology model answers the CQs, testing the completeness of the ontology, and the ontology implementation. 56.6% have used CQs in Conceptualization, to help identify concepts and relationships, guide on what needs to be addressed in the ontology model, document the ontology, and guide and deepen the scope. 23.8% have used CQs in Design, to define axiom rules and aid the design of the ontological model. 12.7% of the participants have used CQs to support Implementation decisions.

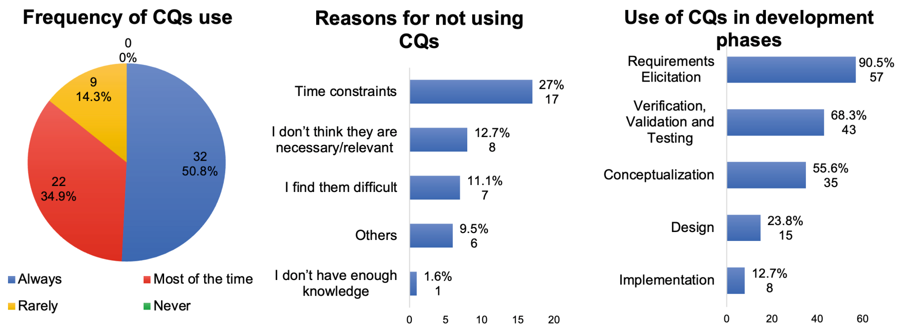


Fig. 3. Use of CQs in OE

Ways of Developing CQs: To capture how CQs have been developed, we asked the participants about the procedure they have followed to develop CQs, the terminology used in CQs, the types of CQs, and the sources used to define CQs. Table 1 summarizes the results. Regarding the procedure adopted to define the CQs, we asked if CQs have been defined iteratively or at once and if they are adjusted/refined along the ontology development. Most of the participants (63.5%) have defined CQs iteratively and refined them during the ontology development. 34.9% of the participants have defined all CQs at the beginning of the ontology development but have also adjusted them during the ontology development. As for the terminology adopted in the CQs, we investigated if the terms used in the CQs are closer to the ontology terminology or to the users terminology. Most of the participants (88.9%) indicated that they have used terms closer to the user. We also asked the participants to provide information about the type of CQs they have defined. In this question, we consider that *CQs in the universe of concepts* are those whose answer is given directly by the ontology concepts/terms, while *CQs in the universe of instances* are those whose answer is obtained from instances of the ontology concepts [1]. In the questionnaire [25], we provided a brief explanation and examples regarding the types of CQs considered in the question. Most participants (65.1%) have used both types of CQs, while 25.4% have preferred the use of CQs in the universe of concepts. Only one participant (1.6%) declared that have preferred the use of CQs in the universe of instances. To investigate the sources of CQs, we asked the participants to indicate which ones they have used. The predominant sources have been domain experts (92.1%), papers, books, standards, and other documents about the domain (87.3%), and existing ontologies or ontology patterns (76.2%).

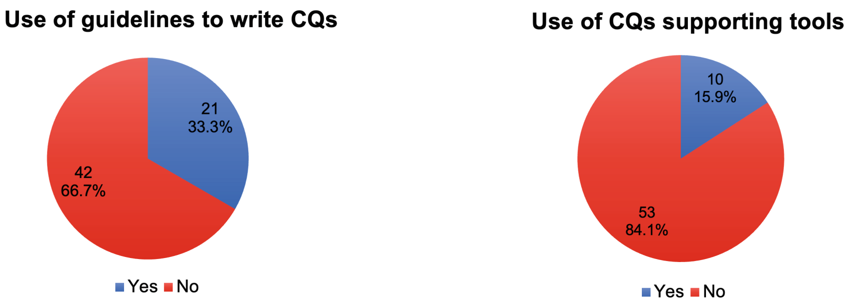


Fig. 4. CQs supporting tools

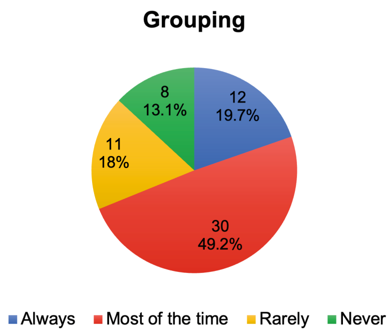
CQs Supporting Tools: To investigate if ontology engineers have been supported by tools or guidelines when defining or using CQs, we asked the participants about methods, frameworks or approaches that have provided them with guidelines on how to write CQs. Most of the participants (66.7%) declared that have not had the support of any guideline. Two declared that have used CLaRO [24] and the others informed that they have followed guidelines given by the OE methods they use or basic guidelines defined by themselves (e.g., general knowledge of lexico-syntactic patterns used to express CQs). We also asked the participants about the tools they have used. The majority of them (84.1%) declared that have not used any tool. The others indicated that have used some tools and cited text editors, electronic spreadsheets, Protegé [26], CLaRO (and associated tools) [24], OntoKEM Tool [46], and Freemind [34]. Figure 4 shows the results related to CQs supporting tools.

CQs Management: Depending on the ontology scope, many CQs may be necessary. Organizing them in groups or modules can help divide the problem, better

Table 1. Procedure, terminology, types and knowledge sources

Procedure to define CQs	Quantity	%
I identify the CQs iteratively and I adjust/refine them in the course of the ontology development if necessary	40	63.5%
I identify all needed CQs in the beginning of ontology development and I adjust/refine them in the course of the ontology development if necessary	22	34.9%
I identify the CQs iteratively	1	1.6%
I identify all needed CQs in the beginning of ontology development	0	0%
Terminology used in CQs	**Quantity**	**%**
Initially, I define questions with terminology closer to that used by users. In the course of the ontology development, I review the questions to bring them closer to the ontology concepts	45	71.4%
They are questions that use the terminology used by users	11	17.5%
They are questions that use the terminology used in the ontology	7	11.1%
Types of CQs	**Quantity**	**%**
I use both CQs in the universe of concepts and CQs in the universe of instances	41	65.1%
They are usually CQs in the universe of concepts (e.g., CQ1 above)	16	25.4%
I don't use CQs in the requirements elicitation phase	5	7.9%
They are usually CQs in the universe of instances (e.g., CQ2 above)	1	1.6%
Knowledge sources used to define CQs	**Quantity**	**%**
Interaction with domain experts (interviews, surveys, etc.)	58	92.1%
Papers, books, standards, and other documents about the domain	55	87.3%
Existing ontologies or ontology patterns	48	76.2%
Information systems developed for the domain	26	41.3%
Others	4	6.3%

understand the addressed domain and contribute to establishing the ontology modularization [9,39]. To investigate if CQs grouping has been a concern, we asked whether and how the participants have grouped CQs. Most participants (86.9%) declared that have grouped CQs at some degree – 19.7% always, 49.2% most of the time, and 18% rarely. The others (13.1%) informed that have never grouped CQs. The participants who have grouped CQs informed that the CQs groups have been based on the ontology modules, subontologies, or subdomains, or considering the proximity of concepts. We also investigated if ontology engi-



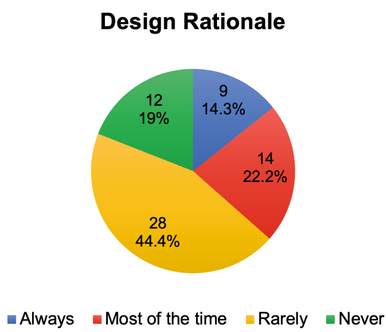


Fig. 5. CQs management

neers have defined the CQs design rationale. A design rationale is the explicit listing of decisions and the reasons why those decisions were made [22]. Its primary goal is to provide a means to record and communicate the argumentation and reasoning behind the design process [19]. Therefore, the design rationale makes explicit the reasons that led someone to define a CQ, i.e., the intentions behind the CQ. Most of the participants have not been concerned with describing the design rationale of the CQs (19% have never defined and 44.4% have rarely defined). Around one third of the participants declared that have defined the design rationale of the CQs (14.3% have always done that and 22.2% have done that most of the time). Figure 5 illustrates these results.

Benefits and Difficulties of Using CQs: Aiming at identifying the benefits and difficulties of using CQs, we asked the participants to indicate which ones they have perceived. We provided a list of options and the participants were allowed to indicate others. Figure 6 shows the results. The most cited benefits were: CQs help define the ontology scope (92.1%) and CQs aid in ontology evaluation (82.5%). On the other hand, the main difficulties reported were: ensuring that the defined CQs are the ones necessary and sufficient for the ontology (77.8%), identifying CQs truly capable of representing the ontology scope (63.5%), and writing the CQs properly (49.2%). Difficulties reported by the participants that selected the "Others" option were: identifying CQs when the ontology engineer is still learning about the domain; extracting CQs from the interaction with domain experts; making domain experts understand the value of CQs and how they help the ontology development; and work with CQs in a systematic way from the beginning to the end of ontology development.

5 Discussion

In this section, we discuss the collected data and results by considering the topics identified in the previous session.

Concerning **ontology development**, most of the participants (54%) have developed both, reference and operational ontologies, which indicates that attention has been given to ontologies as conceptual models and also as computational

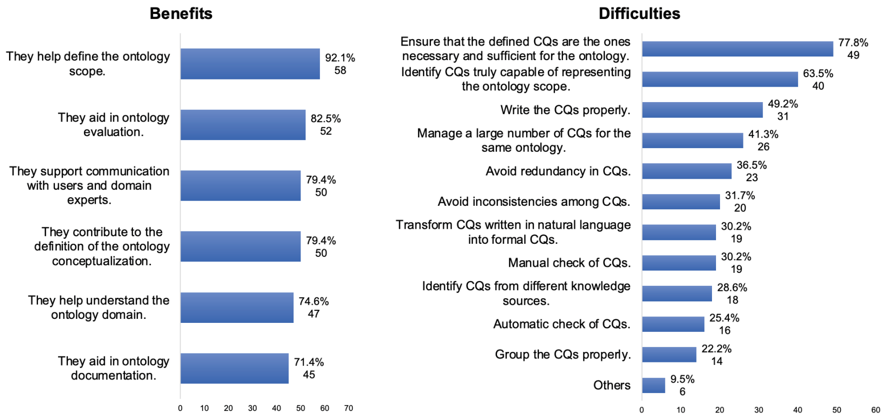


Fig. 6. Benefits and difficulties of using CQs

artifacts. This can also suggest that there has been a concern with modeling the knowledge addressed in the ontology before building the machine-readable version to handle data related to that knowledge. However, when we look at each ontology type, we notice a predominance of reference ontologies (93.7% in total) when compared to operational ontologies (60.3% in total). This indicates that in many cases, ontologies have been developed to be used at the conceptual level (e.g., in ontology-driven software development [29]), not requiring operational ontologies to be applied at run-time (e.g., to support reasoning). More than 90% of the participants have used OE methods to guide ontology development. This is aligned with the OE literature (e.g., [9,11,14]), which points out that developing ontologies is not an easy task and, thus, guidance is needed. This also indicates that the community has some maturity regarding the use of standardized practices. We must observe that this result may have been influenced by the fact that most participants work in the academic environment, where many OE methods have been proposed and their use is encouraged.

Regarding the **use of CQs in OE**, as the study focused on ontology engineers with experience in CQs, it was expected that CQs would be used to some degree. Around 85% of the participants declared that have systematically used CQs, which suggests that the key role of CQs has been recognized. It must be noted that this result is probably related to the fact that most participants have used OE methods such as SABiO [9], METHONTOLOGY [12], and NeOn [43], which recommend the use of CQs. The predominance of the use of CQs in Requirements Elicitation, Conceptualization, and Verification, Validation and Testing phases corroborates the literature, which states that CQs should be used mainly to support specifying the requirements the ontology should fulfill [16], identifying the ontology concepts, properties and relationships [9], and verifying if the ontology properly represents the intended knowledge and meets the established requirements [15]. The results also highlight time constraints as the main reason for not using CQs. On one hand, time and effort are indeed necessary to understand the domain, identify the stakeholders needs and describe the requirements in the form of CQs. On the other hand, CQs help define the ontology scope and establish a kind of "contract" about what the ontology must address, supporting communication among stakeholders and providing the basis for the next ontology development activities [44]. Therefore, the time spent on defining the CQs is rewarded when rework decreases, the ontology is documented, and the CQs aid in other development steps [44]. Despite that, 13% of the participants declare that, in some cases, they do not consider CQs necessary. Moreover, 11% find them difficult. This suggests a need for guidelines and ways to facilitate the use of CQs for ontology engineers to make the most of them. Furthermore, this also raises attention to the need for investigating CQs limitations in depth and comparing the use of CQs to other methods that address ontology requirements.

As for the **ways of developing CQs**, there has been a predominance of iterative approaches (63.5%), in which the ontology engineer defines CQs in a development process performed in multiple cycles and refines the CQs as the knowledge of the domain grows and gets more mature. The use of iterative approaches is particularly important when developing large ontologies, when requirements are not clear at the beginning, or when there are many stakeholders or many conflicts between stakeholders' needs. The uncertainty about the ontology requirements or the need to focus on a domain portion each time lead to a gradual definition of CQs. Around one third of the participants have defined all needed CQs at the beginning of the ontology development and refined them later, if necessary. This approach is suitable when the ontology scope is well-known and the ontology is not large or complex. Although the two aforementioned approaches differ mainly in when CQs are defined (at each iteration or at the beginning of the OE process), both consider the need to refine CQs along the ontology development. This fact is a recognition of the dynamism of CQs, which is itself a consequence of the dynamic and knowledge-intensive character of the OE process. In both cases, top-down, bottom-up, or middle-out strategies (see Sect. 2) can be used to refine the CQs. The predominance of the iterative approach also suggests that OE methods should address ontology development as an iterative process. This is aligned with some perceptions from the literature, which points out that some ontology applications require OE methods that help ontology engineers continuously gather and prioritize requirements from several stakeholders, keep domain experts engaged, deliver ontology modules according to time demands, respond to changing knowledge, and evolving the ontology, in an agile [5] and continuous [37] approach.

When asked about the types of CQs, most of the participants (65.1%) indicated that have used CQs expressed in an interrogative form that works over both concepts and instances [1]. CQs focusing on the ontology concepts concern the ontology conceptualization itself and, thus, usually help identify the ontology concepts and relationships. CQs focusing on instances, in turn, are usually concerned with data handled by the ontology and, thus, are particularly valuable for evaluating the ontology. This result is consistent with the OE phases where CQs have been used more often (Requirements Elicitation, Conceptualization, and Verification, Validation and Testing).

As for the terminology used in the CQs, around 70% of the participants declared that they initially define CQs closer to the users and later review the CQs to bring them closer to the ontology concepts terminology. These results once again emphasize the dynamic and evolutive character of CQs. The growing understanding of the domain promotes a better understanding of the ontology purpose and scope, which can lead to changes in CQs already captured [9]. A terminology closer to users adopts a vocabulary easily understood by domain experts. This terminology may be changed along the ontology development process to be more consistent with naming conventions used to represent the ontology concepts. However, this change requires careful attention because if, on one hand, it can be beneficial to ontology engineers involved in the development process, it can prevent users from properly understanding the ontology. Moreover, it can also affect ontology reuse because if the scope of an ontology candidate to reuse is represented by CQs and the person interested in reusing that ontology is not able to understand them, the opportunity to reuse it can be lost.

Ontology engineers have used multiple sources of knowledge to define CQs and interaction with domain experts (92.1%), documents (87.3%), and existing ontologies or ontology patterns (76.2%) have been the main ones. They can be used together to elucidate knowledge. For example, brainstorming techniques, informal interviews with experts, and inspecting similar ontologies allow elaborating a first glossary with terms potentially relevant. Formal and informal analysis of text (documents) might be used to refine the terms. Interviews with experts might help build concept classification trees [12]. Reusing ontologies or design patterns[1] provides knowledge about ontology requirements and also helps speed up the ontology development process. By containing concepts and relationships relevant to the domain of interest, ontology and ontology patterns helps identify CQs. Moreover, if the ontology/ontology pattern also provides the respective CQs, they can be reused in the new ontology.

With respect to **CQs supporting tools**, the results show a lack of tools to support defining and using CQs (84.1% of the participants declared not having had the support of any tool) and also a lack of guidelines on how to write CQs (66.1% have not followed any guideline). Concerning guidelines, some participants informed that have used the ones provided by the OE methods they use. However, the existing guidelines on how to define, write and use have not been enough. As for supporting tools, the participants cited generic tools such as text editors and electronic spreadsheets. Protégé [26] was also cited, but it is a tool for ontology implementation, thus support for requirements elicitation is limited. Automation is a key factor to build, release, and maintain ontologies effectively [21]. Collaborative editing tools and communication systems are necessary to support requirements elicitation activities. Providing support for writing CQs and automating tasks such as grouping CQs, retrieving terms candidates from the CQs for the ontology, generating documentation, and tracing requirements could help decrease development time and improve the ontology quality [33].

[1] An ontology design pattern is a generic solution to a recurring ontology modeling problem [39].

Regarding **CQs management**, grouping CQs has been a common practice (around 70% of the participants have grouped CQs always or most of the time). Grouping CQs help manage the ontology complexity by dividing the problem into smaller ones and organizing knowledge. CQs grouping can serve as a basis to the ontology modularization (or vice-versa), providing a way to structure the ontology as an interconnected collection of modules, each of which resonating with the corresponding part of the domain conceptualization [9,39,43]. Grouping CQs is especially important when managing large ontologies, which require many CQs. In these cases, it is hard to keep track of CQs and prioritize them if we do not know which part of the ontology they impact.

The results also show that describing the design rationale behind the CQs has not been usual. More than 60% of the respondents declared that have never or rarely made the design rationale explicit. Clarifying the design rationale is important to enable a better understanding of the ontology scope and requirements. It contributes to the ontology development and also to its reuse. The lack of concern with declaring the design rationale has been highlighted in ontology reuse literature. One of the challenges of ontology reuse is the obscurity of the design rationale of the available ontologies [22]. Unknown design rationale makes it difficult to select the ontologies to be reused as well as to understand them, which is crucial to integrate them properly [35]. We hypothesize that the lack of design rationale is related to the time constraints indicated by the participants as a drawback when using CQs. Identifying and writing CQs demand time, which is certainly increased if one describes the reasons for defining each of them.

Finally, with regard to the **benefits and difficulties of using CQs**, most of the participants declared to have perceived all the benefits presented in the questionnaire. The most cited ones (CQs help define the ontology scope (92.1%) and CQs aid in ontology evaluation (82.5%)) are consistent with the OE phases where the participants have used CQs the most (Requirements Elicitation, and Verification, Validation and Testing). These results are also consistent with the literature (e.g., [9,15,33,43]) and provide evidence of CQs usefulness.

On the other hand, the reported difficulties demonstrate that identifying CQs is not trivial. Although ontology engineers have used several OE methods that suggest the use of CQs, they have still faced difficulties mainly to ensure that the defined CQs are the ones necessary and sufficient for the ontology, identify CQs truly capable of representing the ontology scope, and write the CQs properly. We believe that these difficulties (particularly the first and second ones) are due to the fact that defining CQs involves a lot of tacit knowledge and, even though general guidelines as the ones provided by some OE methods are helpful, they have not been enough to address the gap between theory and practice. As a knowledge-intensive activity, defining CQs relies on the ontology engineer's knowledge and experience. Thus, there is bias and subjectivity [32] because it depends on the way the ontology engineer thinks to follow in one direction or another (e.g., to use one type of CQ or another, to choose one or other terminology). Concerning difficulty of writing CQs correctly, some initiatives in the literature have tried to mitigate this issue through some controlled natural language or lexico-syntactic patterns [1,11,32,36,50]. However, it seems that, in general, the proposed guidelines have not reached the ontology engineers or have not been enough. Anyway, there is a need to better support ontology engineers to define CQs either by providing detailed and practical guidelines and tools, or by improving the access and use of the existing ones.

6 Threats to Validity and Limitations of the Study

In this section, we discuss some threats involved in the survey and that should be considered together with the results. We use the classification presented in [51].

The main threat refers to the study sample, which might not reflect experiences from the entire OE community. Ideally, the sample should be larger and the geographic distribution of the people more diverse. Thus, the number of participants and the fact that the sample was selected by convenience is a limitation that affects *External Validity* (i.e., to what extent it is possible to generalize the study results). To minimize this threat, we invited people from different countries and organizations and also the authors of papers selected in an ongoing systematic literature mapping we are carrying out. Moreover, we asked people to freely invite other people.

The decisions and data interpretations made by the researchers affect *Reliability Validity*, which refers to what extent data and analysis depend on specific researchers. To minimize this threat, analysis was initially carried out by two of the authors and, thus, reviewed by the other. Discussions were performed to refine the conclusions and reach a consensus.

There are also threats to *Construct Validity*, which refers to the constructs involved in the study and how they can affect the results. The main threat is the possibility of the participants misunderstand the questions. To address this threat, we performed a pilot that allowed us to improve and clarify questions. Moreover, we provided examples and definitions for the used terms, so that the participants could better understand how to answer the questions. Another threat is related to the scale used in some questions. Since we did not provide any common grounds, different participants may have interpreted terms (e.g., rarely) subjectively. The alternatives of answer provided in each question can also represent a threat, since they may not represent all the relevant alternatives. To address this threat, when defining the questions and the respective alternatives of answer, we considered results from the systematic literature mapping about CQs we are carrying out and, in addition, we included "Others" as an alternative the participants could choose and provide further information. Still regarding the questionnaire, it is important to be aware that the results reflect the participants' personal experience, interpretation and beliefs. Hence, the answers can embed subjectivity that could not be captured through the questionnaire.

Finally, concerning *Internal Validity*, which refers to the ability of a new study to repeat the behavior with the same participants and objects, the main threat refers to the participants providing inaccurate answers for thinking that they could be evaluated. To mitigate this threat, we informed the participants that data would not be evaluated individually. In addition, the participants were free to inform or not their identification (email) when filling in the form.

7 Final Considerations

This paper presented a survey that provides a panoramic picture of how CQs have been used in OE. In summary, the results show that CQs have been considered useful and have helped mainly define ontology scope and evaluate ontology conceptualization. Most of the time they have been defined iteratively and refined along the ontology development process. Different knowledge sources and types of CQs have been considered. Grouping CQs has been frequent while making the design rationale explicit has not been a common concern. Time constraints have been the main reason for not using CQs and there is a lack of supporting tools. Although there are OE methods that provide guidelines to define CQs, they are still limited, which causes ontology engineers to face difficulties when writing, using, and managing CQs.

These results provide a panorama of the use of CQs in OE and also raise some issues that can be addressed in future research. The lack of practical and detailed guidelines and supporting tools certainly contributes to the difficulties faced by ontology engineers and to increasing the time needed to use CQs. There are opportunities to go deeper into existing guidelines or provide new ones and enrich examples of using CQs along the OE process. The study reported in this paper gives a step towards better understand the use of CQs in OE. However, further investigation is needed to address CQs limitations and improve CQs use. Moreover, as there are other techniques to support requirements elicitation (e.g., natural language statements, tabular information [33]), comparing their use and also studying the combination of them may provide further evidence and help ontology engineers to use the ones more suitable for their needs.

Currently, we are carrying out a systematic mapping of the literature to provide a panorama of the state of the art about the use of CQs in OE. We intend to analyze the results of both studies to reach an overview of the state of the art and the state of practice in CQs use and shine a light on the road ahead. Moreover, we envision repeating the survey with a larger and wider geographically distributed sample. We also intend to investigate aspects related to the CQs content. We expect that, based on the studies results, we can establish recommendations to help ontology engineers use CQs. Aiming at supporting reuse, efforts towards the creation of a knowledge base containing domain-related CQs can also be addressed in future work.

Acknowledgments. We thank all the study participants and also all the people who helped spread the study call for participation. This research is supported by FAPES (Process 2023-5L1FC and T.O. 1022/2022).

References

1. Bezerra, C., Santana, F., Freitas, F.L.G.: CQChecker: a tool to check ontologies in OWL-DL using competency questions written in controlled natural language. Learn. Nonlinear Models **12**, 115–129 (2014)
2. Bharti, P., Yang, Q., Forbes, A., Romanchikova, M., Hippolyte, J.L.: Ontology development for measurement process and uncertainty of results. Meas. Sens. **18**, 100325 (2021)
3. Blomqvist, E., Hammar, K., Presutti, V.: Engineering ontologies with patterns-the extreme design methodology. Ontol. Eng. Ontol. Design Patterns **25**, 23–50 (2016). https://doi.org/10.3233/978-1-61499-676-7-23
4. Borges Ruy, F., de Almeida Falbo, R., Perini Barcellos, M., Dornelas Costa, S., Guizzardi, G.: SEON: a software engineering ontology network. In: Blomqvist, E., Ciancarini, P., Poggi, F., Vitali, F. (eds.) EKAW 2016. LNCS (LNAI), vol. 10024, pp. 527–542. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-49004-5_34
5. Copeland, M., Brown, A., Parkinson, H.E., Stevens, R., Malone, J.: The SWO project: a case study for applying agile ontology engineering methods for community driven ontologies. ICBO **7**, 2012 (2012)
6. Corcho, O., Fernández-López, M., Gómez-Pérez, A.: Ontological engineering: principles, methods, tools and languages. In: Calero, C., Ruiz, F., Piattini, M. (eds.) Ontologies for Software Engineering and Software Technology, pp. 1–48. Springer, Heidelberg (2006). https://doi.org/10.1007/3-540-34518-3_1
7. Corcho, O., Fernandez-Lopez, M., Gomez-Perez, A.: Ontological engineering: what are ontologies and how can we build them? In: Semantic Web Services: Theory, Tools and Applications, pp. 44–70. IGI Global (2007)
8. Easterbrook, S., Singer, J., Storey, M.A., Damian, D.: Selecting empirical methods for software engineering research. Guide Adv. Empir. Softw. Eng. 285–311 (2008)
9. Falbo, R.D.A.: SABiO: systematic approach for building ontologies. In: 1st Joint Workshop ONTO.COM/ODISE on Ontologies in Conceptual Modeling and Information Systems Engineering. Fois (2014)
10. Fernandes, P.C.B., Guizzardi, R.S., Guizzardi, G.: Using goal modeling to capture competency questions in ontology-based systems. J. Inf. Data Manag. **2**(3), 527 (2011)
11. Fernández-Izquierdo, A., Poveda-Villalón, M., García-Castro, R.: CORAL: a corpus of ontological requirements annotated with Lexico-syntactic patterns. In: Hitzler, P., et al. (eds.) ESWC 2019. LNCS, vol. 11503, pp. 443–458. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-21348-0_29
12. Fernández-López, M., Gómez-Pérez, A., Juristo, N.: Methontology: from ontological art towards ontological engineering. American Association for Artificial Intelligence (1997)
13. Fox, M.S., Grüninger, M.: Ontologies for enterprise integration. In: CoopIS, pp. 82–89. Citeseer (1994)
14. Gašević, D., Djurić, D., Devedžić, V.: Model driven Architecture and Ontology Development, vol. 10. Springer, Cham (2006). https://doi.org/10.1007/3-540-32182-9
15. Gómez-Pérez, A.: Evaluation of ontologies. Int. J. Intell. Syst. **16**(3), 391–409 (2001)
16. Grüninger, M., Fox, M.S.: The role of competency questions in enterprise engineering. In: Rolstadås, A. (ed.) Benchmarking—Theory and Practice. IAICT, pp. 22–31. Springer, Boston, MA (1995). https://doi.org/10.1007/978-0-387-34847-6_3

17. Guarino, N.: Formal ontology in information systems. In: Proceedings of the First International Conference (FOIS 1998), 6–8 June 1998, Trento, Italy, vol. 46. IOS press (1998)
18. Guizzardi, G.: Conceptualizations, modeling languages, and (meta) models. In: Databases and Information Systems IV: Selected Papers from the Seventh International Baltic Conference, DB&IS 2006, vol. 155, p. 18. IOS Press (2007)
19. Horner, J., Atwood, M.E.: Effective design rationale: understanding the barriers. In: Dutoit, A.H., McCall, R., Mistrík, I., Paech, B. (eds.) Rationale Management in Software Engineering, pp. 73–90. Springer, Heidelberg (2006). https://doi.org/10.1007/978-3-540-30998-7_3
20. Iqbal, R., Murad, M.A.A., Mustapha, A., Sharef, N.M., et al.: An analysis of ontology engineering methodologies: a literature review. Res. J. Appl. Sci. Eng. Technol. **6**(16), 2993–3000 (2013)
21. Jackson, R.C., Balhoff, J.P., Douglass, E., Harris, N.L., Mungall, C.J., Overton, J.A.: ROBOT: a tool for automating ontology workflows. BMC Bioinform. **20**, 1–10 (2019)
22. Jarczyk, A.P., Löffler, P., Shipman, F.M.: Design rationale for software engineering: a survey. In: Proceedings of the Hawaii International Conference on System Sciences, vol. 25, p. 577. IEEE Institute of Electrical and Electronics (1992)
23. Júnior, P.S.S., Barcellos, M.P., de Almeida Falbo, R., Almeida, J.P.A.: From a scrum reference ontology to the integration of applications for data-driven software development. Inf. Softw. Technol. **136**, 106570 (2021)
24. Keet, C.M., Mahlaza, Z., Antia, M.-J.: CLaRO: a controlled language for authoring competency questions. In: Garoufallou, E., Fallucchi, F., William De Luca, E. (eds.) MTSR 2019. CCIS, vol. 1057, pp. 3–15. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-36599-8_1
25. Monfardini, G.K.Q., Salamon, J.S., Barcellos, M.P.: Survey about the use of competency questions in ontology engineering - protocol & data extraction (2022). https://doi.org/10.6084/m9.figshare.23280698
26. Musen, M.: The protégé project: a look back and a look forward. AI Matt. **1**(4), 4–12 (2015). https://doi.org/10.1145/2557001.25757003
27. Noppens, O., Liebig, T.: Ontology patterns and beyond: towards a universal pattern language. In: Proceedings of the 2009 International Conference on Ontology Patterns, vol. 516, pp. 179–186 (2009)
28. Noy, N.F., McGuinness, D.L., et al.: Ontology development 101: a guide to creating your first ontology (2001)
29. Pan, J.Z., Staab, S., Aßmann, U., Ebert, J., Zhao, Y.: Ontology-Driven Software Development. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-31226-7
30. Pardo, C., Pino, F.J., Garcia, F., Baldassarre, M.T., Piattini, M.: From chaos to the systematic harmonization of multiple reference models: a harmonization framework applied in two case studies. J. Syst. Softw. **86**(1), 125–143 (2013)
31. Peroni, S.: A simplified agile methodology for ontology development. In: Dragoni, M., Poveda-Villalón, M., Jimenez-Ruiz, E. (eds.) OWLED/ORE -2016. LNCS, vol. 10161, pp. 55–69. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-54627-8_5
32. Potoniec, J., Wiśniewski, D., Ławrynowicz, A., Keet, C.M.: Dataset of ontology competency questions to SPARQL-OWL queries translations. Data Brief **29**, 105098 (2020)

33. Poveda-Villalón, M., Fernández-Izquierdo, A., Fernández-López, M., García-Castro, R.: LOT: an industrial oriented ontology engineering framework. Eng. Appl. Artif. Intell. **111**, 104755 (2022)
34. Freemind Project: Freemind (2023). http://freemind.sourceforge.net/wiki/index.php/Documentation. Accessed 31 May 2023
35. Reginato, C., et al.: A goal-oriented framework for ontology reuse. Appl. Ontol. **17**(3), 365–399 (2022). https://doi.org/10.3233/AO-220269
36. Ren, Y., Parvizi, A., Mellish, C., Pan, J.Z., van Deemter, K., Stevens, R.: Towards competency question-driven ontology authoring. In: Presutti, V., d'Amato, C., Gandon, F., d'Aquin, M., Staab, S., Tordai, A. (eds.) ESWC 2014. LNCS, vol. 8465, pp. 752–767. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-07443-6_50
37. Salamon, J.S., Barcellos, M.P.: Towards a framework for continuous ontology engineering. In: XV Seminar on Ontology Research in Brazil (ONTOBRAS 2022), pp. 158–165 (2022)
38. Sene, A., Kamsu-Foguem, B., Rumeau, P.: Data mining for decision support with uncertainty on the airplane. Data Knowl. Eng. **117**, 18–36 (2018)
39. Shimizu, C., Hammar, K., Hitzler, P.: Modular ontology modeling. Semant. Web **14**(3), 459–489 (2023)
40. Smith, B., et al.: The obo foundry: coordinated evolution of ontologies to support biomedical data integration. Nat. Biotechnol. **25**(11), 1251–1255 (2007)
41. Studer, R., Benjamins, V.R., Fensel, D.: Knowledge engineering: principles and methods. Data Knowl. Eng. **25**(1–2), 161–197 (1998)
42. Suárez-Figueroa, M.C., Gómez-Pérez, A.: Ontology requirements specification. In: Suárez-Figueroa, M.C., Gómez-Pérez, A., Motta, E., Gangemi, A. (eds.) Ontology Engineering in a Networked World, pp. 93–106. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-24794-1_5
43. Suárez-Figueroa, M.C., Gómez-Pérez, A., Fernández-López, M.: The NeOn methodology for ontology engineering. In: Suárez-Figueroa, M.C., Gómez-Pérez, A., Motta, E., Gangemi, A. (eds.) Ontology Engineering in a Networked World, pp. 9–34. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-24794-1_2
44. Suárez-Figueroa, M.C., Gómez-Pérez, A., Villazón-Terrazas, B.: How to write and use the ontology requirements specification document. In: Meersman, R., Dillon, T., Herrero, P. (eds.) OTM 2009, Part II. LNCS, vol. 5871, pp. 966–982. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-05151-7_16
45. Sure, Y., Staab, S., Studer, R.: On-to-knowledge methodology (OTKM). In: Staab, S., Studer, R. (eds.) Handbook on Ontologies. International Handbooks on Information Systems, pp. 117–132. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-24750-0_6
46. Todesco, J.L., Rautenberg, S., Speroni, R., Guembarovski, R., Gauthier, F.O.: ontoKEM: a web tool for ontologies' construction and documentation. In: IKE, pp. 86–92 (2009)
47. Uschold, M., King, M.: Towards a methodology for building ontologies (1995)
48. Uschold, M., Gruninger, M.: Ontologies: principles, methods and applications. Knowl. Eng. Rev. **11**(2), 93–136 (1996)
49. Uschold, M., Jasper, R.: A framework for understanding and classifying ontology applications. In: Proceedings of the IJCAI-99 Workshop on Ontologies and Problem-Solving Methods (KRR5), Stockholm, Sweden, vol. 2 (1999)

50. Wiśniewski, D., Potoniec, J., Ławrynowicz, A., Keet, C.M.: Analysis of ontology competency questions and their formalizations in SPARQL-OWL. J. Web Semant. **59**, 100534 (2019)
51. Wohlin, C., Runeson, P., Höst, M., Ohlsson, M.C., Regnell, B., Wesslén, A.: Experimentation in Software Engineering. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-29044-2
52. Yago, H., Clemente, J., Rodriguez, D.: ON-SMMILE: ontology network-based student model for multiple learning environments. Data Knowl. Eng. **115**, 48–67 (2018)