1 Methodology

This chapter discusses the methods used to perform this research, thereby enabling the evaluation of reliability and validity. First, interviews were conducted to understand the collaboration between experts and non-expert adopters in the current landscape of quantum computing. Additionally, these interviews aimed to identify supposed collaborative inefficiencies in the collaboration between quantum computing experts and non-expert adopters. As a consequence, SRQ 3 and SRQ 4 were aimed to be answered through analysis of these interviews.

Second, ontology engineering was approach systematically to capture the knowledge and satisfy the requirements specified by the expert interviews. This way, it was aimed to achieve a knowledge representation that ensures reusability and maintainability. Additionally, this systematic approach aimed to benefit from the expertise of ontology experts and ensure consistency with ontologies created in the past. With this approach to ontology engineering, this research intends to answer SRQ 5 and SRQ 6.

Section 3.1 explains the methodology applied throughout the expert interviews. Subsequently, the design process of the ontology is described throughout Section 3.2.

1.1 Understanding Quantum Divide via Expert Interviews

This research phase uses relatively unstructured interviews. While unstructured interviews can be hard to manage and process due to the inconsistency in interview questions (Gill et al., 2008; Turner, 2010), they are considered most suitable due to the required depth in answers and the unfamiliarity with the quantum computing domain. First, some general key questions were constructed to provide a starting point for the interview, which are stated in Appendix B. Aiming for an informal conversational environment, adapting to the answers to these questions was important. This flexibility allowed for discovering information and perspectives that were not yet known. Furthermore, the qualitative research guide by Turner (2010) states that this flexibility results in a more personal approach to each interview. This personal approach seemed particularly useful, as honesty and trust are required for interviewees to express their feelings about deficiencies in the collaboration between quantum computing experts and non-expert adopters.

The most appropriate suggestions for conducting qualitative research proposed by Turner (2010) were applied to ensure the validity of the interviews and the data processing. In Sections 3.1.1, 3.1.2, 3.1.3, and 3.1.4, the selection of candidates, a pilot test, the interview procedure, and the data interpretation method are described, respectively.

1.1.1 Participants

Creswell (2007) stresses the importance of finding the right research participants. It is essential to find participants who do “not hesitate to speak and share ideas”, especially in one-on-one interviews (Creswell, 2007, p. 133). By accurately describing the research’s purpose when contacting candidates, two candidates recognized that their colleagues were more suitable to participate in the research. In total, 12 candidates were approached, eventually resulting in the eight research participants (P) briefly introduced in Table 1.

Table 1, Overview of research participants in the expert interviews

No	Organization	Function	Task/Expertise
P1	Quantum ecosystem	Research associate	Researcher on quantum algorithm analysis
P2	Quantum ecosystem	Research associate	Managing platform
P3	University	Doctor of philosophy	Quantum technologies & communication
P4	University	Assistant professor	Quantum computing, machine learning, and artificial intelligence
P5	Research center for quantum software	Researcher	Quantum Innovation Officer
P6	Research institute mathematics & computer science	Senior researcher	Computer science & theoretical physics
P7	Consultancy firm	Manager	Quantum specialist in cyber risk services
P8	University. Research center for quantum software	Institute coordinator	Coordinating research and development, focus on quantum and society

The diversity of research participants was necessary to ensure a genuine assessment of the collaboration between quantum computing experts and non-expert adopters. Participants were diverse in their focus of interest, obtained from a description of their job's task, as can be derived from Table 1. However, obtaining a diverse set of relevant participants, varying their level of quantum expertise, was difficult. Therefore, the participant's seniority was used as a proxy instead. Furthermore, in the knowledge obtained from the interviews, the level of expertise was also observed by observing the use of differing jargon to explain the quantum theory. Figure 11 positions the research participants amongst the two dimensions mentioned above, illustrating diversity. It can be seen that quantum computing non-expert adopters were not directly involved in this research besides the author. Although, P3 and P7 can be considered experts that focus on non-expert adopters in their daily operations. Similarly, P2, P5, and P8 focus on a quantum ecosystem, or the discipline of quantum and society, which both involve an understanding of quantum computing adoption.

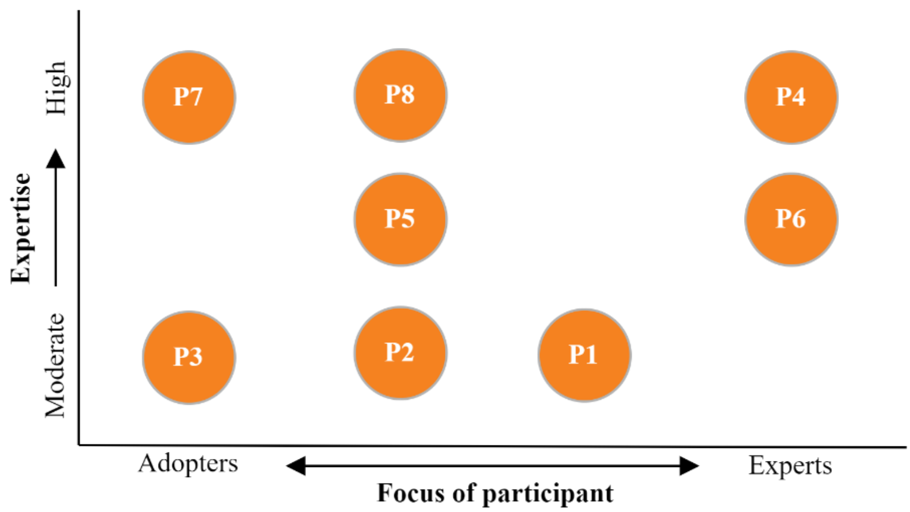


Figure 1, Research participant's diversity in focus and expertise

1.1.2 Pilot-Test

A pilot test was performed to identify any limitations in the initial interview questions and prepare for the expert interviews. The test is beneficial when the participant is similar to the selected research participants (Turner, 2010). Such a participant was represented by a master's student in applied physics with a basic understanding of quantum theory. In addition, the interview questions were shared with a supervisor of this research. The test resulted in the following refinements:

- A minimal understanding was required to discuss quantum computing knowledge in the interviews. However, in the early stage of this research, quantum computing was not sufficiently understood. Therefore, it was decided to focus more on the challenges and nature of the collaboration between quantum computing experts and non-expert adopters (i.e., SRQ 3 and SRQ 4). Before this decision, the aim was to answer SRQ 5 as well.
- Additional interview questions on general background information were formulated.

1.1.3 Interview Procedure

Firstly, targeted candidates were contacted by email. The email's contents, which can be seen in Appendix C, contained a short introduction about the interviewer, the purpose of the research, and the candidate's relevance to the research. If the candidate was positive about contributing to the research, the date and time were scheduled to conduct the interview. Subsequently, the interview took place in a face-to-face online environment (e.g., Microsoft Teams) to support the previously mentioned personal and conversational approach. With the participants' permission, the interview audio was recorded. The recordings were transcribed to ensure the insights were documented accurately and entirely. Firstly, audio recordings were automatically transcribed through Microsoft Word's transcription functionality. After that, these transcriptions were examined manually to check for incompleteness or incorrectness.

1.1.4 Data Interpretation

Thematic analysis (TA) was performed on the transcripts to systematically identify and organize the interview findings (Braun & Clarke, 2012). TA is beneficial due to its flexibility, as coding unstructured interviews is perceived difficult due to inconsistency in the interview questions (Creswell, 2007). Qualitative data analysis software *QDA Miner* (Provalis Research, 2012) is used for computer-assisted coding and analysis to perform the TA. As proposed by Braun & Clarke (2012), the following steps were carried out, sometimes iteratively:

1. **Familiarization with data:** Familiarization was carried out in multiple ways. Firstly, transcribing itself can be seen as data familiarization. When transcribing, the audio recordings were often consulted to ensure some transcriptions were not misunderstood. In addition, the review of literature stated in Section 5.1 was carried out in parallel with conducting and processing the interviews, which was highly beneficial for familiarizing with the data collected and, most importantly, familiarizing with the quantum computing domain.
2. **Generating initial codes:** Interesting excerpts of the transcripts were coded with QDA Miner. Although most codes were descriptive, in the sense that they reflected the excerpts directly, they became more interpretative when more interviews were processed.
3. **Searching for themes:** Identifying patterns in the codes, preferably targeted towards SRQ 3 and SRQ 4.
4. **Reviewing themes:** The eight transcriptions eventually resulted in 187 excerpts of text that were coded, which is quite extensive. Themes were continuously reviewed to ensure that all codes

fitted well with their corresponding theme (e.g., changing boundaries of themes might cause a relocation to a different theme).

5. **Defining and naming themes:** Themes should have clear boundaries (i.e., they are not too general), be non-repetitive (i.e., they do not overlap), and be relevant to the research questions (Braun & Clarke, 2012). The last point was difficult to achieve as the interviews were approached inductively, which meant that it aimed to obtain insights from the data without preconceived theories. An overview of all themes and codes identified can be found in Table 19 in Appendix D.
6. **Producing report:** The presentation of this qualitative research with vivid examples. The aim was to provide well-grounded conclusions, preferably backed up with literature. The results of this step are stated in Chapter 4.

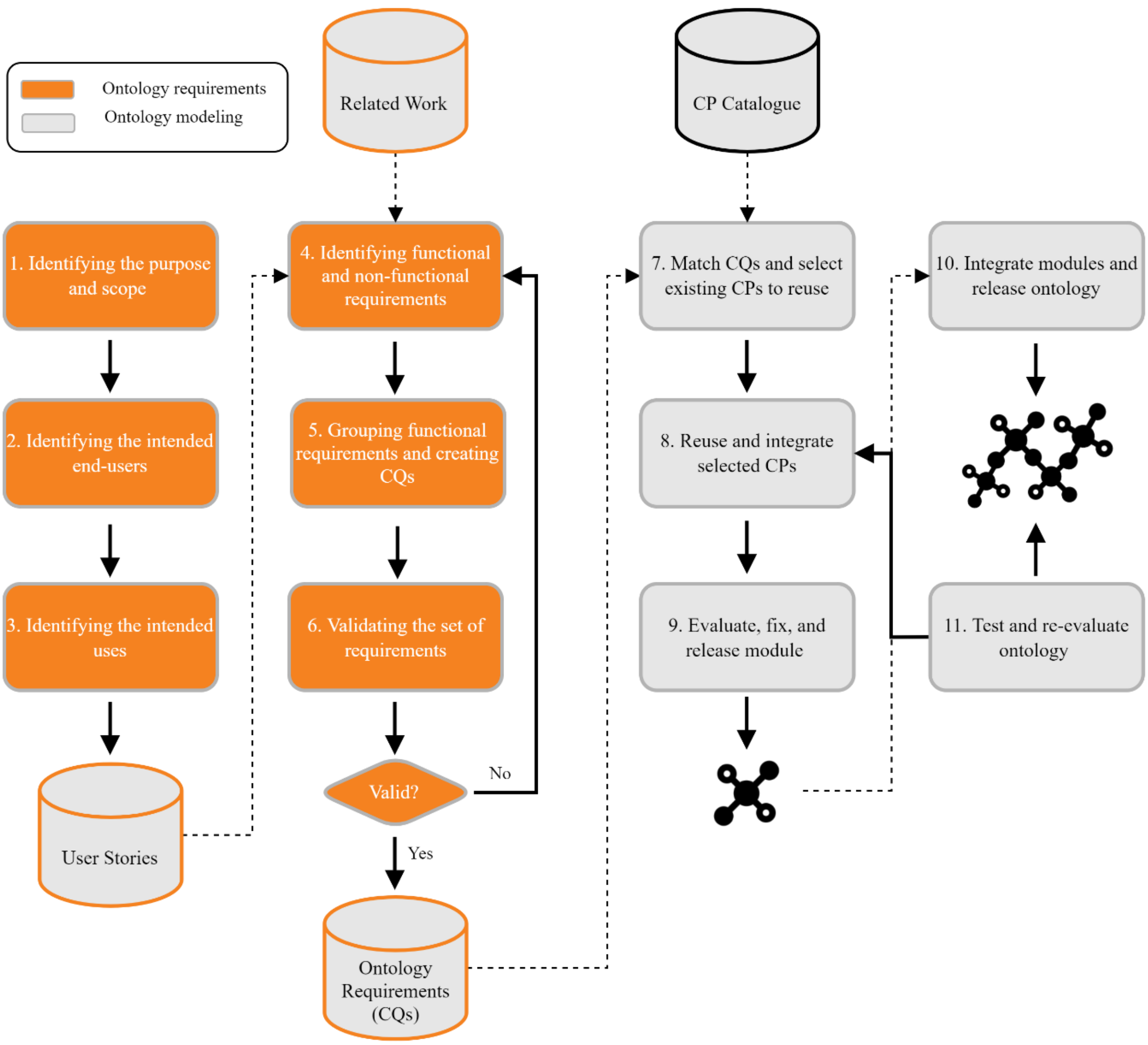
1.2 Ontology Engineering

The NeOn methodology for ontology engineering was used to support the systematic creation of an ontology, which focuses on reusing and re-engineering existing ontologies (Suárez-Figueroa et al., 2012). In doing so, it was aimed to benefit from the expertise of ontology experts, as ontology engineering is a substantive field in computer science. Furthermore, this approach ensures consistency with previously created ontologies. The NeOn methodology proposes seven scenarios that are commonly faced in ontology engineering. Scenarios 6 and 7 seemed most applicable, as they both include operations on merging and re-engineering, thereby supporting the modeling of varying knowledge, which is a characteristic of the knowledge specified to include in the ontology (as listed in Section 5.1). Eventually, it was decided to focus on scenario 7, which reuses ontology design patterns (ODP).

Gangemi & Presutti (2009) describe an ODP as a "modeling solution to a recurrent ontology design problem." With the support of ODPs, the average developer perceives guidance, thereby improving the ontology's quality, task coverage, and usability while modeling issues are prevented (Blomqvist et al., 2009). Furthermore, ODPs are generally accessible, as best practices are available in an ODP catalog, reviewed by the Association for Ontology Design & Patterns (n.d.). More specifically, this research focuses on reusing Content ODPs (CP), which are domain- and use-case-dependent. Due to these dependencies, CPs are particularly useful for knowledge engineering, as a domain can contain many use cases, and one use case can be found across various domains (Gangemi & Presutti, 2009). As a result, existing CPs can be used as modeling solutions for various domains by generalizing the use case it serves.

The creation of the quantum computing ontology is approached by adapting and complementing the step-wise approach to requirement specification (Suárez-Figueroa & Gómez-Pérez, 2012) (Figure 43 in Appendix E) with the eXtreme Design approach to ontology engineering (Blomqvist et al., 2016; Presutti et al., 2009) (Figure 44 in Appendix E), which both have proved to be successful in ontology engineering research (Ansari et al., 2018; Carriero et al., 2021; Scrocca et al., 2021). An overview of the methodology carried out in ontology engineering is illustrated in Figure 12.

Throughout this chapter, Section 3.2.1 describes the methodology to the ontology requirements specification, which involves the findings of the expert interviews. Subsequently, Section 3.2.2 describes the modeling steps carried out in creating the ontology. Lastly, Section 3.2.3 formulates the procedure for evaluating the ontology.



CP: Content Ontology Design Pattern
CQ: Competency Question
Figure 2, Overview of the ontology engineering process, combining requirement specification (Suárez-Figueroa & Gómez-Pérez, 2012) and the eXtreme design approach (Blomqvist et al., 2016; Presutti et al., 2009)

1.2.1 Requirements

The methodology for the ontology requirements specification is clarified by the first six tasks of Figure 12. Throughout this chapter, the execution of these tasks is described.

Task 1-3

The first three tasks were aimed at deriving user stories by extensively involving domain experts. Therefore, these user stories were captured throughout the conducted expert interviews. More precisely, the identification of end-users (i.e., task 2) was reflected in answering SRQ 3, and the intended uses of the ontology (i.e., task 3) were reflected in answering SRQ 4. The resulting user stories, represented by the expert interviews' outcomes, are stated in Chapter 4.

Task 4

Subsequently, task 4 identifies the ontology's functional and non-functional requirements. According to Suárez-Figueroa et al. (2009), functional requirements specify the knowledge represented by the ontology. On the other hand, non-functional requirements denote general characteristics unrelated to the ontology’s content.

While Suárez-Figueroa & Gómez-Pérez (2012) obtained this set of requirements solely from analyzing user stories, this was perceived to be unfeasible in this research due to the limited quantum computing domain knowledge of this research's author. Furthermore, due to the knowledge gap between the interviewer and the experts, it was often challenging to discuss functional requirements with the domain experts. Therefore, an analysis of related work (i.e., literature and knowledge-based systems) was carried out to compensate for this knowledge gap and derive functional requirements (Section 5.1). On the contrary, non-functional requirements could be derived from the expert interviews (Section 5.2), although literature also substantiates these.

Task 5

Afterward, the functional requirements were grouped to enable a modular development of the ontology (Suárez-Figueroa & Gómez-Pérez, 2012). A modular ontology design improves maintainability, reusability, and the efficiency of reasoning (Stuckenschmidt & Klein, 2007). Furthermore, grouping the requirements on their domain or use case supports the reuse of the earlier CPs, which are also created for a specific domain or use case. These groups of functional requirements, represented by specified knowledge, are stated in Section 5.1.

Additionally, task 5 transforms the specified knowledge into competency questions (CQ). CQs capture relevant domain problems and tasks, and the answers to these questions correspond to the specified knowledge. These questions can be used to characterize an ontology, as they define which axioms (i.e., expressions) an ontology should contain to solve the problems and tasks (Grüninger & Fox, 1995). An overview of the CQs derived from the functional requirements is stated in Section 5.3.

Task 6

Lastly, the functional requirements, now represented by CQs, were validated against criteria proposed by Suárez-Figueroa & Gómez-Pérez (2012). These criteria are disclosed in Table 2, where a distinction is made between validated and unfeasible criteria. For example, some criteria could not be explicitly validated due to domain experts' missing involvement in tasks 4-6. Alternatively, the majority of criteria were validated through the review of related work, which was beneficial in, for example, satisfying the traceability criterion. On the other hand, some criteria, such as consistency and unambiguity, caused iterating through tasks 4-6 due to the complexity of the quantum computing domain. Furthermore, the verifiability criterion was not validated at the requirement specification but in testing the eventual implementation (Section 7.3).

Table 2, Validation criteria for ontology requirements, derived from (Suárez-Figueroa & Gómez-Pérez, 2012, p. 100)

Criteria	Explanation
Validated	
Correctness	Each requirement is represented in features of the ontology.
Consistency	There are no internal conflicts across requirements.
Unambiguity	Each requirement is unambiguous.
Conciseness	There exist no duplicates or irrelevant requirements.
Modifiability	The structure of requirements allows for modifications in a convenient way.
Traceability	The origin of requirements is captured for each requirement.
Verifiability	It is tested whether each requirement is satisfied by the ontology.
Unfeasible	
Completeness	Domain experts conclude that there are no additional requirements.
Understandability	Domain experts and users assess the understandability.

Realism	Each requirement makes sense to domain experts.

1.2.2 Modeling

Tasks 7-10, as seen in Figure 12, represent the steps to model the ontology based on the ontology requirements. This chapter describes the operations performed related to these tasks.

As stated in the previous chapter, requirements were grouped to enable a modular development of the ontology. Thus, the final ontology consists of an integration of developed ontology modules based on groups of CQs.

Task 7-8

Reusing existing CPs requires selection (i.e., task 7) and application (i.e., task 8) (Presutti & Gangemi, 2008). The selection consists of finding existing CPs from the ODP catalog (Association for Ontology Design & Patterns, n.d.) that match closely with the group of CQs to be covered. This process is challenging as the CPs often target a more generic use case or domain that does not match precisely with the local use cases in the problem space (e.g., quantum computing). Therefore, accurate and precise matching is performed in the application process, which requires a set of operations. An overview of these operations, derived from Presutti & Gangemi (2008), is provided in Table 3.

Table 3, Operations required for reusing existing CPs, derived from (Presutti & Gangemi, 2008, p. 131)

Operation	Explanation
Import	Including a CP in the ontology under development
Clone	Duplicating an ontology element (i.e., class and object property)
Specialization	Creating sub-classes of a CP's class or sub-properties of a CP's object property
Generalization	The inverse of specialization
Composition	Relating classes of one CP to classes of other CPs
Expansion	Enriching an ontology with new elements not necessarily part of the CP

Throughout developing an ontology module, two scenarios could occur:

- There are existing CPs that match the defined set of CQs. After finding this appropriate CP, several matching activities can take place (Presutti & Gangemi, 2008, p.138):
  - **Precise or redundant matching:** The CP exactly matches the set of CQs. The import operation satisfies the matching process.
  - **Broader matching:** The CP is more general than the set of CQs. After importing, the specialization operation is required to match the local domain.
  - **Narrower matching:** The CP is more specific than the set of CQs. After importing, the generalization operation is required to match the local domain.
  - **Partial matching:** The CP partially matches our set of CQs. CQs are divided into smaller sets of CQs (i.e., creating subdomains). Subsequently, one of the smaller sets of CQs is covered by the initially chosen CP, and others require a new selection process. This matching process mainly involves import and composition operations.
- No existing CP matches the defined set of CQs:
  - If this occurred, the ontology module was created by extracting elements from the widely adopted foundational ontology DOLCE+DnS Ultra Lite (DUL), potentially followed by one or multiple operations of Table 3.
  - In addition, DUL was often consulted when a CP only matched partially by expanding the existing CP with elements of DUL. A foundational ontology, such as DUL, expresses general

concepts and relations (Presutti et al., 2012). DUL is mainly based on DOLCE (e.g., objects, events, qualities) and DnS (e.g., situations, descriptions, concepts). Thereby it can easily be reused in any domain through the specialization operation, which makes DUL widely used in the design of ODPs and CPs. Consequently, using DUL to define new CPs was considered consistent with the current practices.

- Furthermore, DUL is considered an ontology of "particulars," contrasting with other foundational ontologies (Mascardi et al., 2007). An ontology of particulars focuses on instances rather than properties. It is believed that this feature contributes to the maintainability of the ontology by limiting the necessary object properties (Bassiliades et al., 2018). Maintainability of the ontology is demanded, as research in the quantum computing domain is rapidly evolving due to the continuous development of quantum hardware.

Task 9

Having finished an ontology module by either one of the scenarios stated in the previous paragraph, the module was evaluated individually (i.e., task 9). Firstly, it was validated whether the module could cover the corresponding set of CQs by creating scenarios consisting of sample data relating to the domain the module addresses. This sample data constitutes the answers to the corresponding set of CQs. In this way, modeling inconveniencies were found and fixed. If the design of the ontology module satisfied this test, it was subsequently defined in Web Ontology Language (OWL) with open-source ontology editor Protégé (Musen, 2015). OWL defines and instantiates ontologies by providing extensive semantic meaning to classes and object properties (Bechhofer et al., 2004). In addition, each ontology module was subject to an evaluation procedure explained in Section 3.2.3.

When succeeding in the evaluation, the ontology modules were released. The design of these ontology modules is described throughout Section 6.1. In this documentation, the generally accepted documentation approach of Gangemi & Presutti (2009) is followed, inspired by Alexander's (1979) theories on pattern languages. The concepts used to document the modules accurately are stated in Table 4.

Table 4, Documentation of modules, derived from Gangemi & Presutti (2009, p. 229)

Concept	Explanation
Name	Name of CP
Intent	Generic use case of the CP
Competency questions	Indicates the set of CQs that the module covers
Scenarios	Fact-based requirements and how the module addresses these
Diagram	An illustration of the module's design
Elements	Description of the classes and properties in the module
Reengineered from	The reference ontology (CP) used to create the module
Building block	A reference to an implementation/OWL file of the module

Task 10

After evaluating the distinct ontology modules, these modules were imported into a new OWL file and integrated into the final ontology (i.e., task 10). This method is beneficial to preserve maintainability, as an adapted module only requires a new import. However, integrating ontology modules consists of manual alignment tasks. Deciding on an alignment- and refactoring strategy is "one of the methodology's most crucial and difficult steps" (Blomqvist et al., 2016, p. 19). For example, two ontology modules consisted of identical classes (e.g., a generic class such as "parameter"). Maintaining these identical classes by defining a so-called equivalence property enables independent reuse of the modules in prospective ontology engineering projects. On the other hand, duplicate classes make the ontology less intuitive and increase the complexity for the end-user (Blomqvist et al., 2016). Refactoring was often carried out to solve such integration issues. An additional upper-level ontology model, stated in Section 6.2, was defined to illustrate the most important connections of the ontology modules, enabling integration.

Task 11

Lastly, the final ontology was evaluated to check on any complications arising from the module integration, following the procedure described in Section 3.2.3. Furthermore, the ontology's functioning was tested by creating use cases based on competency questions (i.e., functional requirements) and writing and running corresponding SPARQL queries, as SPARQL facilitates querying graph data (W3C, 2013). These use cases are stated in Section 7.3.

The ontology was deployed and populated using various cloud services offered by Amazon Web Services (AWS) to facilitate testing and enable end usage of the ontology. The corresponding architecture underlying the orchestration of cloud services and Python scripts is described throughout Chapter 7, providing an answer to SRQ 7.

1.2.3 Evaluation

As previously mentioned, a systematic methodology for ontology engineering contributes to the quality of the ontology. However, as can be noticed throughout Section 3.2.2, ontology modeling is considered challenging and requires some subjective decisions. Consequently, assessing the quality of an ontology demands objective evaluation. Several ontology evaluation dimensions exist (Fernández et al., 2009; García et al., 2010; Poveda-Villalón et al., 2014), of which this research considers syntax, domain-cohesion, structure, functionality, and usability.

An ontology engineering expert was consulted to discuss the ontology (modules), focusing on evaluating the syntax and structure dimensions. In addition, the expert had a background in applied physics and mathematics, making an evaluation of domain cohesion possible. As a result, the following improvement points were found and subsequently processed.

- In some modules, cohesion to the domain was weak, as the model did not correctly represent the actual relationships in the domain. For example, classes were modeled as subclasses while not necessarily being so, which was mainly caused by lacking domain knowledge.
- Object properties were excessively specialized, which hampers the overall goal of the semantic web. Furthermore, these object properties made integrating ontology modules (i.e., task 10) challenging, as it demanded extensive refactoring.
- Ontologies should relate concepts naturally and logically. Therefore, inverse relationships were emphasized, especially in the visualization of the ontology modules.
- Proper use of vocabulary is essential in ontology engineering. Therefore, the overall name of the ontology and some classes were renamed.

In addition, the ontology evaluation tool OOPS (i.e., Ontology Pitfall Scanner) was used to evaluate the ontology based on 40 criteria related to structure, functionality, and usability. Each criterion is weighted with an importance/relevance level (i.e., critical, important, and minor). Where critical criteria are crucial to correct, minor criteria do not comprise the ontology and do not create a problem (Poveda-

Villalón et al., 2014). OOPS was used continuously in task 9 and task 11 of Figure 12. Section 7.3 states the results of this evaluation.