Using competency questions to enhance patterns selection in ontology patterns languages

Glaice Kelly da Silva Quirino Monfardini[1,2,*], Monalessa P. Barcellos[1]

[1]*Ontology & Conceptual Modeling Research Group (NEMO), Computer Science Department, Federal University of Espírito Santo, Vitória - ES, Brazil*

[2]*Federal Institute of Espírito Santo, Aracruz - ES, Brazil*

Abstract

Developing ontologies is a complex task and reuse can be helpful. A recent promising reuse approach is the use of Ontology Pattern Languages (OPLs). An OPL establishes relationships between ontology patterns (OP) and provides a process guiding the selection and use of them for modeling problem-solving. By following the OPL flow, the ontology engineer can select the desired OPs and reuse them. However, some limitations have been noticed when using OPLs, particularly regarding the pattern selection mechanism. Aiming to improve pattern selection in OPLs and strengthen their use, in this work, we investigate the use of Competency Questions (CQs) to support the selection of OPs in OPLs. CQs play an important role in the ontology development lifecycle, as they represent the ontology requirements. They are questions that the ontology should be able to answer. With them, it is possible to have a notion of which are the relevant concepts of the domain and the relationships between them. Therefore, we hypothesize that using CQs in OPLs helps ontology engineers select suitable OPs to be reused in the development of new ontologies.

Keywords

Reuse, Competency Questions, Ontology Patterns Language, Ontology Engineering

1. Introduction

Although nowadays ontology engineers are supported by a wide range of ontology engineering methods and tools, building ontologies is still a difficult task even for experts [1]. Reusing knowledge resources, specifically ontology patterns (OPs), has become a popular technique in the Ontology Engineering field. Ontology reuse allows for speeding up the ontology development process, saving time and money, and promoting the application of good practices [2]. A proposal that shows promise in this context is the use of Ontology Pattern Language (OPL). OPLs favor reuse through their network of interconnected Domain-related Ontology Patterns (DROPs) that provide holistic support for the development of ontologies in a given domain [3].

An OPL contains a set of interconnected DROPs, as well as a modeling flow that provides guidance on how to use and combine them in a specific order to solve some modeling problems in that domain [4]. The notion of OPL provides a strong sense of connection between DROPs, expressing different types of relationships between them [4]. For example, to be applied, a pattern may require the prior application of other patterns (dependency); a larger pattern can be composed of smaller ones; or multiple patterns may solve the same problem in different ways (variant patterns). These relationships impose restrictions on the order in which patterns can be applied. Thus, an OPL provides explicit guidance on how to use and integrate related patterns into a concrete conceptual model of a new domain ontology [3]. OPLs encourage the application of one pattern at a time, following the order prescribed by the language. By allowing the selection of DROPs in a guided manner, an OPL frees the ontology engineer from the need to know in advance the entire set of concepts and relationships covered by the OPL [3].

To develop a domain ontology using an OPL, the ontology engineer must select the patterns that

---

Proceedings of the 17th Seminar on Ontology Research in Brazil (ONTOBRAS 2024) and 8th Doctoral and Masters Consortium on Ontologies (WTDO 2024), Vitória, Brazil, October 07-10, 2024

[*] Corresponding author.

glaice.monfardini@ifes.edu.br (G. K. d. S. Q. Monfardini); monalessa@inf.ufes.br (M. P. Barcellos)
0000-0002-1559-6201 (G. K. d. S. Q. Monfardini); 0000-0002-6225-9478 (M. P. Barcellos)

are pertinent to their domain of interest. Then, the ontology engineer can complement the conceptual model resulting from the integration of the selected patterns to address aspects not covered by them. The ontology engineer can, for example, create new concepts as specializations of concepts from the selected patterns to address specific aspects of the domain or include new concepts and relate them in another way (for example, through associations) to the concepts arising from the patterns.

For effective reuse of ontology patterns in OPLs, two main functionalities must be guaranteed: selection and application. The selection corresponds to obtaining the most appropriate pattern for the domain modeling problem to be addressed [5]. Aiming to address these functionalities, in [6], OPL-ML, a modeling language for representing OPLs, was proposed. When using OPLs represented with this modeling language, the ontology engineer must follow the flows of a process model (an adaptation of the UML activity diagram) and apply one pattern at a time, according to the order of the patterns in the paths chosen in the model. Thus, the selection is supported by the OPL process model that guides the ontology engineer in choosing the patterns necessary for modeling the domain of interest. The application, in turn, is assisted by two models (process model and structural model) plus the patterns specification [3].

In recent years, OPLs have been developed for various domains, such as software processes [4, 7], organizations [8], measurement [9] and services [10]. Even when using OPL-ML Quirino et al. [6], experiences of applying these OPLs revealed a great effort required to use the OPL specification that, to date, is recorded in documents. Following the flow in the process model of a large OPL can be confusing and obtaining the desired patterns only in document format requires the effort of manually creating the conceptual model of the selected patterns in a modeling tool. Moreover, following the process model to select the patterns may hinder details about the patterns scope because the model does not provide much information (e.g., to avoid being visually polluted). Therefore, although OPLs speed up the ontology development process, there is still a need to improve pattern selection to enhance the support OPLs provide to the ontology development process.

Competency Questions (CQs) have been recognized as a useful method for representing ontology scope and identifying the questions that the ontology should be able to answer. CQs are thus a means to determine ontology specifications to design and evaluate a knowledge model [11]. Considering the potential of CQs in the development of ontologies, particularly in representing requirements, we hypothesize that using CQs helps ontology engineers select suitable ontology patterns of OPLs to be reused in the development of new ontologies. Thus, the doctoral research described in this paper aims to explore the use of CQs to support the selection of patterns in OPLs and answer the following research question: How can CQs be used to enhance pattern selection in OPLs? As a contribution to the state of the art, this work will provide a multivocal panorama of the use of CQs in Ontology Engineering, promoting a better understanding of the use of this method considering the literature records and also ontology engineers reports. Additionally, we will build knowledge and theoretical support (guidelines) on the use of CQs to structure OPLs using CQs to facilitate their understanding and support pattern selection. As a contribution to the state of the practice, we will restructure existing OPL taking CQs as a basis and we will develop computational support to help ontology engineers use OPLs by decreasing the complexity of their specification.

This paper is organized as follows: Section 2 presents some background notions that support this proposal, Section 3 discusses some related work, Section 4 briefly describes the research method, and Section 5 presents the current state of research and outlines future work.

2. Background

Ontology reuse has long been recognized as a key attribute of ontologies, but the principles and practices of reuse remain underdeveloped. The lack of design through reuse presents a serious problem for the ontology community. In general, reuse can be defined as the process in which available ontological knowledge is used as input to generate new ontologies [12]. It is a special case of design, referring to the task of taking some existing ontology and manipulating it in some way to satisfy the design requirements. Some more specific, related, and sometimes overlapping subtypes of reuse have been defined, such as merging and alignment, integration, and the application of OPs [13].

The use of OPs is an approach that favors the reuse of encoded experiences and good practices [4]. Patterns are vehicles for encapsulating knowledge. A pattern describes a recurring problem that arises in specific contexts and presents a well-proven solution for the problem [14]. Domain-related OPs (DROPs) are reusable fragments extracted from domain ontologies. DROPs should capture the core knowledge related to a domain, and thus they can be seen as fragments of a core ontology of that domain [15]. A domain ontology typically results from the composition of several DROPs, with appropriate dependencies between them, plus the necessary expansion based on specific needs [16]. DROPs are often organized in catalogs. However, this is not enough to truly favor reuse. In a conventional catalog, there is a lack of a strong sense of connection. We need something stronger than simply knowing that another pattern in the collection is related in some way. When collections are presented in conjunction with, e.g., pattern sequences, we get a stronger sense of connection [14]. This is especially important for reusing DROPs. Thus, a good choice is to organize DROPs in OPLs [4].

An OPL is a network of interrelated DROPs that provides holistic support for solving ontology development problems for a specific domain. An OPL contains a set of interrelated DROPs plus a process providing explicit guidance on what problems can arise in that domain, informing the order to address these problems, and suggesting one or more patterns to solve each specific problem. An OPL is not a method for building ontologies. It only deals with reuse in ontology development, and its guidance can be followed by ontology engineers using whatever ontology development method that considers ontology reuse as one of its activities. Some examples of OPLs can be found at [17].

In Ontology Engineering, CQs play a paramount role. They represent the requirements of the ontology and are an effective way to determine what is relevant to the ontology and what is not. CQs can be used to support both, ontology specification and ontology evaluation [18]. 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 [19].

3. Related works

Exploring CQs in ontology development is not a new idea in itself [19, 20, 21]. CQs have been used mainly in requirements specification and ontology evaluation and, thus, help define the ontology scope and evaluate the ontology conceptualization [22, 23, 18, 24, 25, 26]. However, CQs have also been used to assist in other ontology development contexts. For example, some works have proposed approaches that use CQs to generate ontologies automatically or semi-automatically. In [27] was presented the Frame-based Ontology Design Outlet (FrODO), which is a method and tool capable of automatically constructing ontologies from CQs. The domain ontologies produced by FrODO are drafts that can be used to feed agile ontology design methodologies. In [28] a new method for developing ontologies that exploit analogies between different but related knowledge domains represented by ontologies was proposed. It processes two ontologies for related problems or knowledge domains, where one of them is complete while the other needs further development; then, it generates a series of CQs for the domain expert to extend the new ontology that will be generated as a result.

There are also some works in the literature that propose the use of CQs in the selection of DROPs. In [29], the author developed a DROPs search method exploiting both the similarity between pattern CQs and user queries, and the relative abstraction level of general pattern solutions versus concrete user queries in XD Tools, a method shown to increase recall when searching for candidate DROPs. [30], in turn, describes an approach, along with an evaluation strategy for a tool named ODPReco that can recommend the possible DROPs to use in a given ontology. ODPReco analyzes the lexical, structural, and behavioral aspects of an ontology, along with learning from existing DROP implementations to recommend DROPs that can be used for refactoring an ontology. The behavioral aspect of the ontology can be analyzed by making use of the CQs associated with the given ontology. However, they highlight the absence of CQs in some ontologies, which impedes this behavioral analysis. [31, 32] proposed strategies for selecting DROPs to improve the quality of existing ontologies. Among the strategies, they suggested the selection of patterns via requirements (using the CQs of the DROPs). Then, they compared the CQs of the DROPs base with the CQs of the ontology to be improved. They concluded that this strategy was hampered by the fact that CQs do not need to completely cover the scope of the ontology and, mainly, because of the difference in the level of abstraction between them, which varies according to the ontology engineer. The need to improve the documentation of DROPs was also evident.

Like in the aforementioned works, in our work, we propose to use CQs in the selection of patterns. However, this selection occurs in the context of OPLs. Unlike pattern catalogs, OPLs include, in addition to the patterns themselves, a rich description of their relationships and a process that guides the order of application according to the problems to be modeled. OPLs patterns come from a core ontology and have a documentation standard that facilitates the use of these patterns. This solves the problems of abstraction levels, scope and documentation mentioned in the works above. By using CQs, we intend to improve the pattern selection mechanism in OPLs, facilitating their understanding and application.

4. Research method

The research approach adopted in this work is exploratory empirical with studies involving both quantitative and qualitative approaches. We follow the Design Science Research (DSR) paradigm, which concerns extending *"human and organizational capabilities by creating new and innovative artifacts"* [33]. In this work, the main artifact to be produced is a set of guidelines on the use of CQs to structure OPLs. From this, others will be built: OPLs restructured taking CQs as a basis and (ii) a supporting tool to OPL use. DSR comprises the following steps [34]: (i) *Problem identification and Motivation*, (ii) *Definition of the objectives for a solution,* (iii) *Design and Development,* (iv) *Demonstration,* (v) *Evaluation*, and (vi) *Communication.* These steps are organized in an iterative process, with three cycles: *Relevance Cycle*, *Design Cycle*, and *Rigor Cycle*.

A DSR project begins with the *Relevance Cycle*, which involves defining the problem to be addressed, the requirements, and the criteria for evaluating the results [33], including the steps (i) and (ii). The *Design Cycle* involves developing and evaluating artifacts or theories to solve the identified problem [33], comprising steps (iii), (iv), and (v). Finally, the *Rigor Cycle* refers to using and generating knowledge [33], comprising the step (vi) and the use of knowledge and foundations along with the work. These steps, the results produced so far, and future actions are described in the next section.

5. Current state of the research and future works

In the *Problem identification and motivation* step, the problem was identified from the literature (e.g. [6]) and it was also observed in practice by some ontology engineers when applying OPLs. In the current approach, which uses OPL-ML [6] and records the OPL specification in documents, the ontology engineering needs to follow the flows of the OPL process model to select the desired patterns. In the model (adapted from the UML activity diagram), the patterns are represented as rectangles with the pattern name inside it and arrows are used to represent the OPL flow by connecting the patterns according to their relationships (e.g., dependency, variance). To avoid being visually cluttered, the process model does not provide much information about the patterns or conditions to select them. Thus, the ontology engineering often needs to look for information in the OPL specification, making the OPLs use tiresome and confusing, particularly for large OPLs. Moreover, while following the OPL process frees the ontology engineering of deep knowledge about the domain, it also makes pattern selection less flexible and dos not stimulate the ontology engineer to reflect about the ontology scope in-depth.

These difficulties were initially reported by OPL users in seminars and other meetings discussing the topic. Therefore, it was noted that although OPLs speed up the ontology development process, there is still a need to improve pattern selection to enhance the support that OPLs provide to the ontology development process. Given how much CQs help in defining the scope of patterns and understanding them, we envisioned the possibility of using the CQs to improve patterns selecion.

Thus, in the *Definition of the objectives for a solution* step, we decided to explore the use of CQs to support the selection of patterns in OPLs and provide theoretical (guidelines) on the use of CQs to structure OPLs and computational (tool) to support using CQs to select patterns in some OPLs.

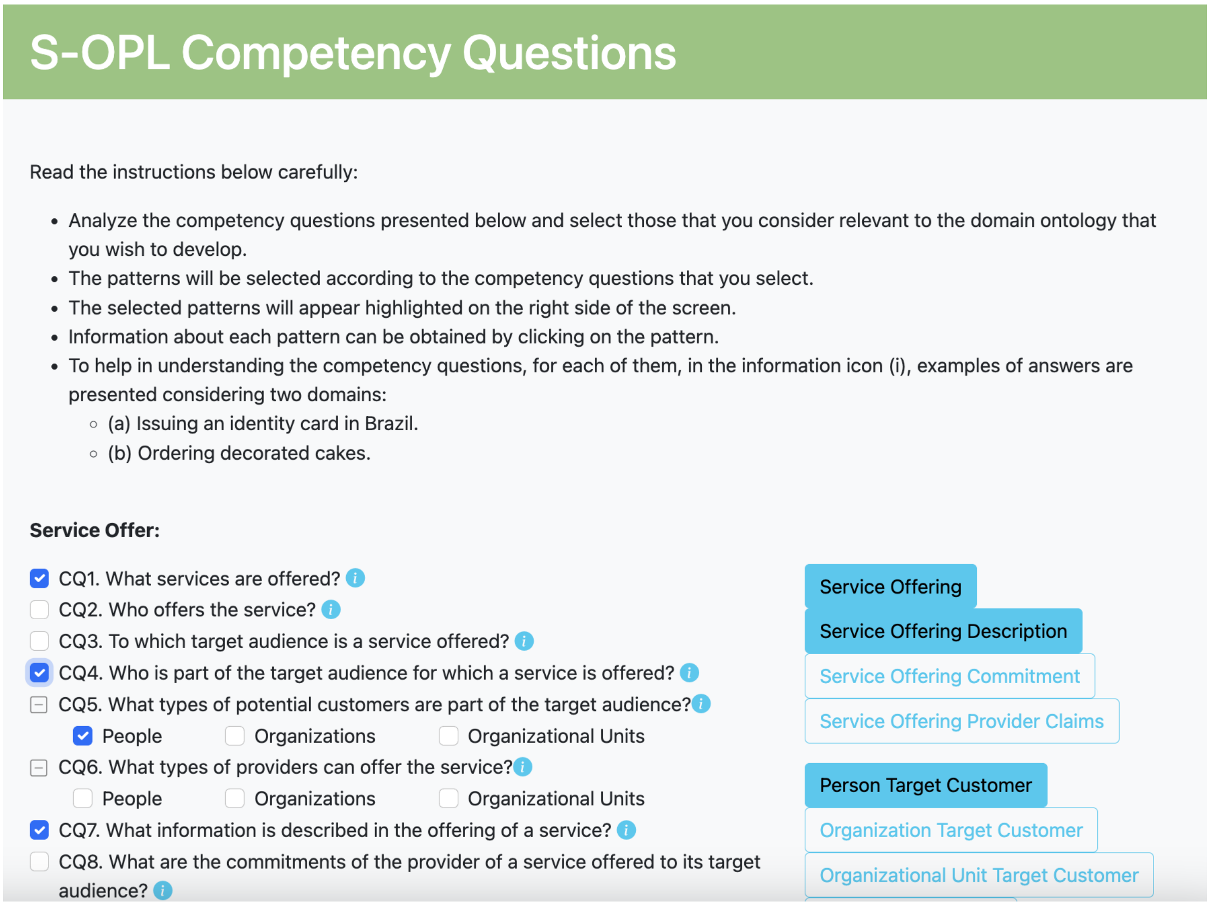
In the *Design and development* step, we aim at developing the guidelines. We started by performing two studies to acquire knowledge about CQs. The first study was a survey with 63 ontology engineers with experience in CQs ([35]). The survey provided a panoramic picture of how CQs have been used in ontology engineering. In summary, the results showed that CQs have been considered useful and have helped mainly define ontology scope and evaluate ontology conceptualization. However, although there are ontology engineering methods that provide guidelines to define CQs, they are still limited, which causes ontology engineers to face difficulties when writing, using, and managing CQs. The second study is a systematic literature mapping [36] to give us a panorama of the state of the art about the use of CQs in Ontology Engineering in the requirements specification context. The study follow the guidelines defined in [36, 37]. For that, we defined fourteen research questions and selected publications in four steps. First, the search string was applied in the search mechanism of Scopus, Science Direct, ACM, IEEE, Engineering Village, and Web of Science. After that, duplicated publications were removed. Then, we applied the first filter by reading the abstracts of the selected publications and selecting only publications that meet inclusion and exclusion criteria. Finally, we applied the second filter, by reading the full texts of the publications included after the first filter and selecting only publications that meet the criteria. As inclusion criteria, we considered that the publication must address CQs in the ontology requirements specification context. As exclusion criteria, we considered: the publication does not have an abstract; the publication is only an abstract; the publication is a secondary study, a tertiary study, a summary, an editorial or a tutorial; the publication is not written in English; the publication is a copy or an older version of an already selected publication; and the publication full text was not available. The publications returned in the publication selection steps were cataloged and stored in spreadsheets. Data from the selected publications was extracted and organized into a data extraction form oriented to the research questions. We considered paper published until December 2023. In total, 298 publications were analyzed. In the end, 44 publications were selected and data was extracted. We have recorded the systematic literature mapping results in a paper to make the study available soon.

Considering the knowledge obtained so far, we have restructured an OPL, namely the Services OPL (S-OPL) [10] [1]. This experience will result in the first version of the guidelines. However, for creating the guidelines, we need to evaluate if the use of CQs improves pattern selection in OPLs. Thus, we carried out a study with three expert ontology engineers who have already used OPLs represented using OPL-ML [6] and faced problems with pattern selection. To perform the study, we developed a tool to support pattern selection in S-OPL. The tool allows the ontology engineer to analyze the set of CQs covered by the patterns of S-OPL and select the ones they judge necessary to the ontology they intend to develop. According to the CQs that the ontology engineer selects, the corresponding OPL patterns are selected. The tool verifies the selected CQs considering dependency relations between the corresponding patterns and warns the ontology engineer if changes are needed. Having resolved all dependencies, the tool generates the list of selected OPL patterns that can, thus, be integrated to generate the desired model. Figure 1 shows a screenshot of the tool.

In the study, the participants used the tool to select patterns from S-OPL and develop ontologies for two domains, considering information provided in two scenarios presented to the participants. After using the tool and selecting the patterns, the list of patterns selected by each participant was sent to the researchers (this paper's authors), who produced the corresponding conceptual model integrating the

---

[1] S-OPL addresses the central conceptualization of service phenomena. The S-OPL patterns were extracted from UFO-S, a commitment-based reference ontology for services [38], whose conceptualization is based on the establishment and fulfillment of commitments and claims between service participants (used to refer to both service providers and service customers) throughout the service lifecycle. S-OPL focuses on the three main phases of the service lifecycle: service offering, service negotiation and service delivery.



S-OPL Competency Questions

Read the instructions below carefully:

- Analyze the competency questions presented below and select those that you consider relevant to the domain ontology that you wish to develop.
- The patterns will be selected according to the competency questions that you select.
- The selected patterns will appear highlighted on the right side of the screen.
- Information about each pattern can be obtained by clicking on the pattern.
- To help in understanding the competency questions, for each of them, in the information icon (i), examples of answers are presented considering two domains:
  - (a) Issuing an identity card in Brazil.
  - (b) Ordering decorated cakes.

Service Offer:

☑ CQ1. What services are offered? ⓘ
☐ CQ2. Who offers the service? ⓘ
☐ CQ3. To which target audience is a service offered? ⓘ
☑ CQ4. Who is part of the target audience for which a service is offered? ⓘ
⊟ CQ5. What types of potential customers are part of the target audience? ⓘ
  ☑ People ☐ Organizations ☐ Organizational Units
⊟ CQ6. What types of providers can offer the service? ⓘ
  ☐ People ☐ Organizations ☐ Organizational Units
☑ CQ7. What information is described in the offering of a service? ⓘ
☐ CQ8. What are the commitments of the provider of a service offered to its target audience? ⓘ

Service Offering
Service Offering Description
Service Offering Commitment
Service Offering Provider Claims

Person Target Customer
Organization Target Customer
Organizational Unit Target Customer

Figure 1: Screenshot of *S-OPL Tool*

selected patterns (the current version of the tool does not integrate the patterns conceptual models), and sent back to the participants. The participants analyzed the resulting model and answered a questionnaire about their experience in the study. Currently, analysis and interpretation of the collected data are ongoing. After analyzing the results, we will document the first version of the guidelines, considering the knowledge acquired so far. Then, we will evolve S-OPL and restructure other OPL considering the study results and the defined guidelines and will conduct a new study with a large and more varied sample of ontology engineers. After that, we will refine the guidelines considering the study results.

In the *Demonstration* step, we will use guidelines to restructure or create an OPL and, thus, in the *Validation* step, the guidelines will be applied by a third party. Finally, in the *Communication* step, which occurs parallel to other steps, we will present the research results by publishing papers and the complete research project will be documented in the doctorate thesis.

Acknowledgments

This research is supported by the Coordination for the Improvement of Higher Education Personnel - CAPES Brazil (Finance Code 001) and the Espírito Santo Research and Innovation Support Foundation - FAPES (Processes 2023-5L1FC, 2022-NGKM5, 2021-GL60J, and T.O. 1022/2022).

References

[1] O. Noppens, T. Liebig, Ontology patterns and beyond: towards a universal pattern language, in: Proceedings of the 2009 International Conference on Ontology Patterns-Volume 516, 2009, pp. 179–186.

[2] M. Poveda-Villalón, M. C. Suárez-Figueroa, A. Gómez-Pérez, Reusing ontology design patterns in a context ontology network, in: Proceedings of the 2nd International Conference on Ontology Patterns-Volume 671, 2010, pp. 35–52.

[3] R. Falbo, M. Barcellos, F. Ruy, G. Guizzardi, R. Guizzardi, Ontology pattern languages, in: Ontology Engineering with Ontology Design Patterns, IOS press, 2016, pp. 133–159.

[4] R. de Almeida Falbo, M. P. Barcellos, J. C. Nardi, G. Guizzardi, Organizing ontology design patterns as ontology pattern languages, in: The Semantic Web: Semantics and Big Data: 10th International Conference, ESWC 2013, Montpellier, France, May 26-30, 2013. Proceedings 10, Springer, 2013, pp. 61–75.

[5] A. Gangemi, V. Presutti, Ontology design patterns, in: Handbook on ontologies, Springer, 2009, pp. 221–243.

[6] G. K. Quirino, M. P. Barcellos, R. A. Falbo, Opl-ml: A modeling language for representing ontology pattern languages, in: Advances in Conceptual Modeling: ER 2017 Workshops AHA, MoBiD, MREBA, OntoCom, and QMMQ, Valencia, Spain, November 6–9, 2017, Proceedings 36, Springer, 2017, pp. 187–201.

[7] F. B. Ruy, R. A. Falbo, M. P. Barcellos, G. Guizzardi, G. K. Quirino, An iso-based software process ontology pattern language and its application for harmonizing standards, ACM SIGAPP Applied Computing Review 15 (2015) 27–40.

[8] R. d. A. Falbo, F. B. Ruy, G. Guizzardi, M. P. Barcellos, J. P. A. Almeida, Towards an enterprise ontology pattern language, in: Proceedings of the 29th Annual ACM Symposium on Applied Computing, 2014, pp. 323–330.

[9] M. P. Barcellos, R. de Almeida Falbo, V. Frauches, Towards a measurement ontology pattern language., in: ONTO. COM/ODISE@ FOIS, 2014.

[10] R. A. Falbo, G. K. Quirino, J. C. Nardi, M. P. Barcellos, G. Guizzardi, N. Guarino, A. Longo, B. Livieri, An ontology pattern language for service modeling, in: Proceedings of the 31st Annual ACM Symposium on Applied Computing, 2016, pp. 321–326.

[11] R. Dandan, S. Desprès, J. Nobécourt, Oafe: An ontology for the description of elderly activities, in: 2018 14th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS), IEEE, 2018, pp. 396–403.

[12] E. P. Bontas, M. Mochol, R. Tolksdorf, Case studies on ontology reuse, in: Proceedings of the IKNOW05 International Conference on Knowledge Management, volume 74, Citeseer, 2005, pp. 345–353.

[13] M. Katsumi, M. Grüninger, What is ontology reuse?, in: Formal Ontology in Information Systems: Proceedings of the 9th International Conference (FOIS 2016), volume 283, IOS Press, 2016, pp. 9–22.

[14] F. Bushmann, K. Henney, D. C. Schmidt, The neon methodology for ontology engineering, in: Pattern-Oriented Software Architecture, volume 5, John Wiley & Sons, 2007.

[15] R. d. A. Falbo, G. Guizzardi, A. Gangemi, V. Presutti, Ontology patterns: clarifying concepts and terminology, in: Proceedings of the 4th Workshop on Ontology and Semantic Web Patterns, volume 1188, 2013.

[16] E. Blomqvist, A. Gangemi, V. Presutti, Experiments on pattern-based ontology design, in: Proceedings of the fifth international conference on Knowledge capture, 2009, pp. 41–48.

[17] NEMO, Ontology patterns and ontology pattern languages, 2024. URL: https://nemo.inf.ufes.br/en/projetos/patterns-and-pattern-languages/.

[18] R. d. A. Falbo, Sabio: Systematic approach for building ontologies., in: 1st Joint Workshop ONTO.COM / ODISE on Ontologies in Conceptual Modeling and Information Systems Engineering. Fois, 2014.

[19] M. Uschold, M. Gruninger, Ontologies: Principles, methods and applications, The knowledge engineering review 11 (1996) 93–136.

[20] V. Presutti, E. Blomqvist, E. Daga, A. Gangemi, Pattern-based ontology design, Ontology Engineering in a Networked World (2012) 35–64.

[21] M. C. Suárez-Figueroa, C. Pradel, N. Hernandez, Verifying ontology requirements with swip, Informatica, 2012.

[22] C. Shimizu, K. Hammar, P. Hitzler, Modular ontology modeling, Semantic Web 14 (2023) 459–489.

[23] M. C. Suárez-Figueroa, A. Gómez-Pérez, M. Fernández-López, The neon methodology for ontology engineering, in: Ontology engineering in a networked world, Springer, 2011, pp. 9–34.

[24] M. Fernández-López, A. Gómez-Pérez, N. Juristo, Methontology: from ontological art towards ontological engineering, American Asociation for Artificial Intelligence (1997).

[25] A. Gómez-Pérez, Evaluation of ontologies, International Journal of intelligent systems 16 (2001) 391–409.

[26] N. F. Noy, D. L. McGuinness, et al., Ontology development 101: A guide to creating your first ontology, 2001.

[27] A. Gangemi, A. S. Lippolis, G. Lodi, A. G. Nuzzolese, automatically drafting ontologies from competency questions with frodo, in: Towards a Knowledge-Aware AI, IOS Press, 2022, pp. 107–121.

[28] V. Nejković, N. Petrović, Ontology development approach adopting analogy and competency questions, in: Conference on Information Technology and its Applications, Springer, 2023, pp. 288–297.

[29] K. Hammar, Ontology design patterns: Adoption challenges and solutions, in: Second International Workshop on Semantic Web Enterprise Adoption and Best Practice in conjunction with the 11th European Semantic Web Conference (ESWC 2014), Anissaras, Greece, May 26, 2014, EUR-WS, 2014, pp. 23–32.

[30] M. A. Yasvi, R. Mutharaju, Odpreco-a tool to recommend ontology design patterns., in: WOP@ ISWC, 2019, pp. 71–75.

[31] B. Lantow, K. Sandkuhl, V. Tarasov, Selecting content ontology design patterns for ontology quality improvement., in: ILOG@ BIR, 2013, pp. 68–79.

[32] B. Lantow, K. Sandkuhl, V. Tarasov, Ontology reuse., in: KEOD, 2015, pp. 163–170.

[33] A. R. Hevner, A three cycle view of design science research, Scandinavian journal of information systems 19 (2007) 4.

[34] K. Peffers, T. Tuunanen, M. A. Rothenberger, S. Chatterjee, A design science research methodology for information systems research, Journal of management information systems 24 (2007) 45–77.

[35] G. K. Q. Monfardini, J. S. Salamon, M. P. Barcellos, Use of competency questions in ontology engineering: A survey, in: International Conference on Conceptual Modeling, Springer, 2023, pp. 45–64.

[36] B. Kitchenham, S. Charters, et al., Guidelines for performing systematic literature reviews in software engineering, 2007.

[37] B. Kitchenham, L. Madeyski, D. Budgen, SEGRESS: Software Engineering Guidelines for REporting Secondary Studies, IEEE Transactions on Software Engineering 49 (2023) 1273–1298. doi:10.1109/TSE.2022.3174092.

[38] J. C. Nardi, R. de Almeida Falbo, J. P. A. Almeida, G. Guizzardi, L. F. Pires, M. J. van Sinderen, N. Guarino, C. M. Fonseca, A commitment-based reference ontology for services, Information systems 54 (2015) 263–288.