Investigating Tool Usage in Ontology Engineering: a Survey

Clarete A. Diniz Gomes[1], Glaice Kelly Q. Monfardini[1,2], Jordana S. Salamon[1], Rosiane da Silva Sangali[3], Isauflânia S. Ribeiro Timoteo[1], Monalessa P. Barcellos[1] and Vítor E. Silva Souza[1,*]

[1] *Ontology & Conceptual Modeling Research Group (NEMO), Federal University of Espírito Santo (UFES), Brazil*
[2] *Federal Institute of Espírito Santo (IFES), Brazil*
[3] *Optimization Laboratory (LabOtim), Federal University of Espírito Santo (UFES), Brazil*

Abstract

The use of ontologies has been acknowledged as a successful approach to solving interoperability and knowledge-related problems. They have also been recognized as valuable for explainability in AI-based systems. Developing ontologies involves a lot of tacit knowledge and requires tool support. Several tools are available and can be used for different purposes to aid ontology engineering (OE) activities. The use of a suitable set of tools contributes to increasing development process performance and improving ontology quality. Aiming at getting evidence about the use of tools to support OE, we conducted a survey with 74 ontology engineers. We investigated which tools have been used, their main characteristics, benefits and difficulties perceived when using them, and properties considered important in OE tools. The results indicate that tools have been widely adopted, but some issues, such as integration problems and scope limitations, still need to be addressed. This paper presents the survey, its results, and discusses the main findings.

Keywords

Tools, Ontology, Ontology Engineering, Survey

1. Introduction

Ontologies allow the formal representation of fundamental concepts interpreted by users, in a way that facilitates communication and understanding of a given domain [1]. They have been recognized as conceptual tools of great importance in Computer Science since the end of the 1960s. In the last fifteen years, there has been an increase in works related to ontologies in various segments of Computer Science [2], mainly in areas such as Data Modeling (conceptual modeling) [3, 4] and Artificial Intelligence [5, 6], and ontologies have become the predominant way to deal with semantic aspects in semantic integration initiatives [7].

Developing ontologies is not trivial. Several ontology engineering (OE) methods have been proposed to aid ontology engineers in developing high-quality ontologies (e.g., [8, 9, 10, 11, 12, 13]). They decompose OE into other processes (or phases) and recommend activities for each one. Although there are differences among OE methods, in general, they consider that OE core activities involve defining the ontology requirements, capturing and developing the ontology conceptualization, implementing the ontology, and evaluating it. OE also relies on tools. They emerged as a form of automation to optimize and accelerate the ontology development process [14] and are crucial to formalizing ontological conceptual models, disseminating knowledge, implementing machine-readable ontologies, and ensuring ontology quality [15].

ONTOBRAS'24: 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.

✉ clarete.gomes@edu.ufes.br (C. A. D. Gomes); monalessa@inf.ufes.br (M. P. Barcellos); vitor.souza@ufes.br (V. E. S. Souza)
🆔 0009-0004-1657-7557 (C. A. D. Gomes); 0000−0002−1559−6201 (G. K. Q. Monfardini); 0000−0002−3209−5273 (J. S. Salamon); 0009-0003-9618-2039 (R. d. S. Sangali); 0009-0001-4415-6264 (I. S. R. Timoteo); 0000−0002−6225−9478 (M. P. Barcellos); 0000-0003-1869-5704 (V. E. S. Souza)

Currently, there is a wide range of tools to support OE. Some are devoted to OE activities (e.g., Protégé[1]), while others were not conceived for OE but can be used in this context (e.g., Visual Paradigm[2]). Selecting the more suitable tools that provide the necessary features is an important decision in the OE process and it may not be easy [16]. Several factors should be taken into account, such as ontology scope, team characteristics (e.g., size, experience), organization context (e.g., policies, preferences), and ontology's intended use. Considering the tool characteristics is also relevant (e.g., availability, features). Some OE methods help tool selection by suggesting some to support their activities [17, 13].

Although there has been an evolution in the functionalities of tools to support ontology development and some tools have become very popular (e.g., Protégé) [18], there is still a need to understand what (combination of) tools offer the best support for the OE process [19]. For that, it is necessary to understand which tools have been used, their characteristics, perceived benefits, and faced difficulties.

To take a step in this direction and gather evidence on this subject, we carried out a survey with 74 ontology engineers to investigate tools used to support OE. The results provide a preliminary picture of the state of practice. With that, we intend to share data and perceptions with other ontology engineers and shine a light on opportunities to advance the research topic.

In summary, the results reinforce the importance of using tools to aid OE. Tools have been widely used, with a predominance of tools for conceptual model development and ontology implementation. Both, OE-specific (e.g., ontology editors) and generic (e.g., text editors) tools have been used and there has been a preference for free tools. Functional suitability, reliability, usability, and availability have been considered the more relevant properties. Ontology development acceleration and ontology quality improvement have been perceived as the main benefits, corroborating the literature [20, 21, 4]. On the other hand, integration problems and scope limitations have been the main obstacles to overcome.

In this paper, we provide an overview of our study, summarize the main findings, and discuss the results. The remainder of the paper is organized as follows: Section 2 presents the theoretical background for the paper; Section 3 presents the study protocol; Section 4 synthesizes the results; Section 5 discusses the results; Section 6 addresses the study limitations; Section 7 discusses related works; and Section 8 presents our final considerations.

2. Background

An ontology is a formal and explicit specification of a shared conceptualization [22]. The conceptualization is an abstract and simplified view of the portion of the world intended to be represented. An important distinction sets apart ontologies as conceptual models, called *reference ontologies*, from ontologies as computational artifacts, called *operational ontologies* [23]. A reference ontology is built aiming at 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 artifacts designed with the focus on guaranteeing desirable computational properties [9].

Engineering ontologies is a complex and collaborative task, which is challenging for both newcomer and experienced engineers [24, 2]. It requires adequate methodological approaches and appropriate tooling support [25]. The literature presents several OE methods to help in this matter (e.g., Methontology [26], Ontology development 101 [27], eXtreme Design [8], SAMOD [10], LOT [13]). 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 the 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 [28].

[1] https://protege.stanford.edu/

[2] https://www.visual-paradigm.com/

Several tools have been developed specifically to support ontology development, such as Protégé, OLED [29], and Onto4AllEditor [4]. Given the similarity of some OE and Software Engineering (SE) activities (e.g., conceptual modeling), SE supporting tools, such as Astah[3] and Visual Paradigm, have also been used to aid ontology development tasks. Such tools can be used off-the-shelf or adapted to suit OE needs through plugins, like the OntoUML Visual Paradigm plugin presented in [30].

According to Gómez-Pérez et al. [31] OE supporting tools can be grouped in six categories: *development tools*, including tools and integrated suites used to build a new ontology from scratch (usually support documentation, export and import to/from different formats and languages, graphical edition, library management); *evaluation tools*, used to evaluate the ontologies and their related technologies; *merge and alignment tools* support merging and aligning different ontologies in the same domain; *annotation tools* help add instances of concepts and relations and maintain (semi)automatically ontology-based markups in Web pages; *querying tools and inference engines*, allowing querying and performing inferences; and *learning tools*, which derive ontologies (semi)automatically from natural language texts, as well as semi-structured sources and databases, by means of machine learning and natural language analysis techniques.

3. Study Design

The study consisted of a survey whose **goal** was to investigate the use of tools in OE to get an overview of this topic in practice. A survey aims to identify the characteristics of a broad population by generalizing data collected from a representative sample of individuals [32]. Surveys are conducted to produce a snapshot of the situation to capture the current status [33]. We chose this method to reach many individuals, which would be harder or unfeasible through interviews or case studies, for example. We followed the process defined in [33], which comprises five activities: *scoping*, when we delimit the scope of the study and establish its goals; *planning*, when the study design is determined; *operation*, which consists of data collection; *analysis and interpretation*, which involves analyzing data to draw conclusions about the research topic; and *presentation and package*, when the results are communicated.

Aligned with the study goal, we defined three main **research questions**: (RQ1) *Which tools have been used and what are their characteristics?*; (RQ2) *What benefits and difficulties have been perceived when using the tools?*; and (RQ3) *Which properties have been considered important in OE tools?*.

The **instrument** used in the study was a form created in Google Forms. It is composed of a consent term for participation in the study and three sections of questions. The first section has eight closed questions to characterize the participants. The second has 11 closed questions related to RQ1 and RQ2. One question allows the participant to complement their answer by providing information in text format. The third section is related to RQ3 and has one closed question plus two open questions to collect further information, comments, and suggestions. In some questions we used ordinal agreement scales but excluded the neutral option to capture meaningful information. The form is available in the study package [34].

The survey **participants** must be a sample of the target population. Therefore, we targeted ontology engineers with knowledge of and experience with ontology development and OE tooling support.

The **procedure** followed in the study consisted of three steps. In the first step, we carried out a small pilot to evaluate the form and the study protocol. We asked an ontology engineer to answer the questionnaire and report response time, problems, and suggestions. Taking his feedback into account, we made minor adjustments to the form. In the second step, we sent messages inviting people to take part in the study. We sent messages to research groups working with ontologies, mailing lists involving researchers and professionals in the field, contact networks at universities, public and private organizations, and authors of the articles selected in an ongoing systematic literature mapping we are performing on the topic. We also requested that those invited forward the invitation to other individuals whom they believed could participate in the study. The final step involved collecting data from the answered questionnaires, representing collected data in tables and graphs, and analyzing it.

[3] https://astah.net

4. Study Execution and Data Synthesis

Invitations to participate in the study were sent in two batches, the first in late December 2023 and the second in late January 2024. We contacted 227 people and received 74 answers until March 1st, 2023, which amounts to a response rate of 32.6%. 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 74 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 [34].

The participants' profile was identified through questions regarding their level and area of education, knowledge of ontologies, and experience in OE. To better contextualize the results, we also asked about the context in which they have worked with ontologies (academia or productive sector), the ontology types they have developed (reference or operational), and how often they have used OE systematic approaches (e.g., SABiO [9], NeOn [11], Methontology [26]). Most participants are Brazilians (50; 67.6%). Others are from Spain (8; 10.8%), Russia (4; 5.4%), Netherlands (3; 4.1%), Argentina (3; 4.1%), France (2; 2.7%), United Kingdom (1; 1.4%), South Africa (1; 1.4%), and Belarus (1; 1.4%). One participant did not provide this information. Most participants have a background in IT, including Computer Science (73%), Computer Engineering (8.1%), Information Systems (10.8%), and Information Science (6.8%) plus one respondent who has his highest level of training in Social Work, but who also has a degree in Computer Science. Most of the participants work in the academic context (70.3%), have worked with OE for five or more years (62.2%), and have high experience in OE (58.1%). 20.3% of the participants have developed only operational ontologies, 32.4% only reference ontologies, and 47.3% both. Most of the participants (60.8%) often use OE systematic approaches and 21.6% always use them. Figure 1 illustrates data about the participants.

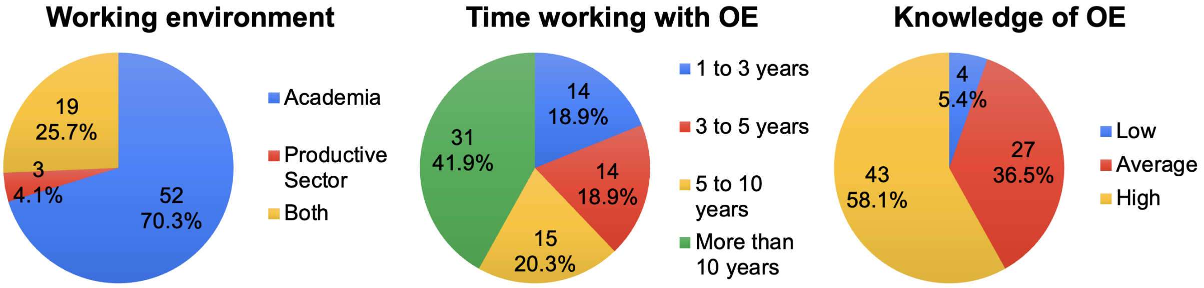

Figure 1: Participants' profile.

In the following, we synthesize collected data by grouping the questionnaire questions into four topics: Use of tools in OE, OE tool characteristics, Perceptions of using OE tools, and Properties of OE tools. The first two topics are related to RQ1, the third to RQ2, and the last one relates to RQ3.

Use of tools in OE: In this topic, we asked which tools the participants have used, how often they have used them, how they found the tools, and in which OE activities they have used OE-specific (i.e., tools addressing OE tasks, such as ontology editors) and generic (i.e., domain-independent tools that address general tasks, such as text editors) tools.

When asked about *how often they use tools* to support OE, the majority of participants (55.4%) informed that they always use tools, 39.2% indicated they use tools most of the time, and 5.4% rarely use them. Concerning the *used tools*, OE-specific tools have been predominant. Protégé (85.1%) has been the most used tool, followed by Visual Paradigm (43.2%), Astah (41.9%), and OntoUML-vp-plugin (36.5%). Protégé is a free, open-source platform that provides a graphical interface for developing ontologies and an architecture with a suite of tools to construct domain models and knowledge-based applications with ontologies [35]. Visual Paradigm and Astah are tools that support creating UML conceptual models and generate code from the models. OntoUML-vp-plugin,[4] in turn, is a plugin designed for OntoUML modelers that adds features to Visual Paradigm to enable the use of OntoUML stereotypes in class diagrams, model checking, and transformation. Other OE-specific tools used by the participants have been OLED [29] (20.3%) and Menthor [36] (16.2%). Both are OntoUML editors that support model building and formal verification, incorporating support for a range of methodological resources for developing, evaluating, and implementing ontologies. Generic tools have also been used, mainly Electronic Spreadsheets (27%), Text Editors (27%), and Slide/Presentation Tools (24.3%). In addition to the tools listed in the figure, others were mentioned by the participants, such as OntoTrack, Onto4AllEditor, Diagram.io, Miro, Widoco, Astrea, OnToology, Hermit, Themis, CmapTools, ONTOFOX, Enterprise Architect, PlantUML, Coggle.it, and yEd Graph Editor.

As for *how the participants discovered the tools*, most of them (82.1%) received recommendations from other people (e.g., friends, work/study colleagues, teachers). 39.2% discovered the tools in courses or classes; 37.8% by searching the Internet; 36.5% have used tools employed where they work, 35.1% have used tools developed by the organization they work; and 17.6% declared to have used tools indicated in OE systematic approaches.

To investigate *tooling support provided to ontology development phases*, we asked the participants to indicate the phases that have been supported by OE tools and the type of used tools (OE-specific or generic). Requirements Specification has been supported mainly by generic tools (89.2% of the participants indicated that have used that kind of tool in this phase). The other phases have had a predominance of OE-specific tools: 64.9% in Ontology Conceptualization and Formalization; 71.2% in Ontology Design, 73% in Ontology Implementation, and 58.1% in Verification, Validation, and Testing. Figure 2 summarizes data about the use of tools in OE.

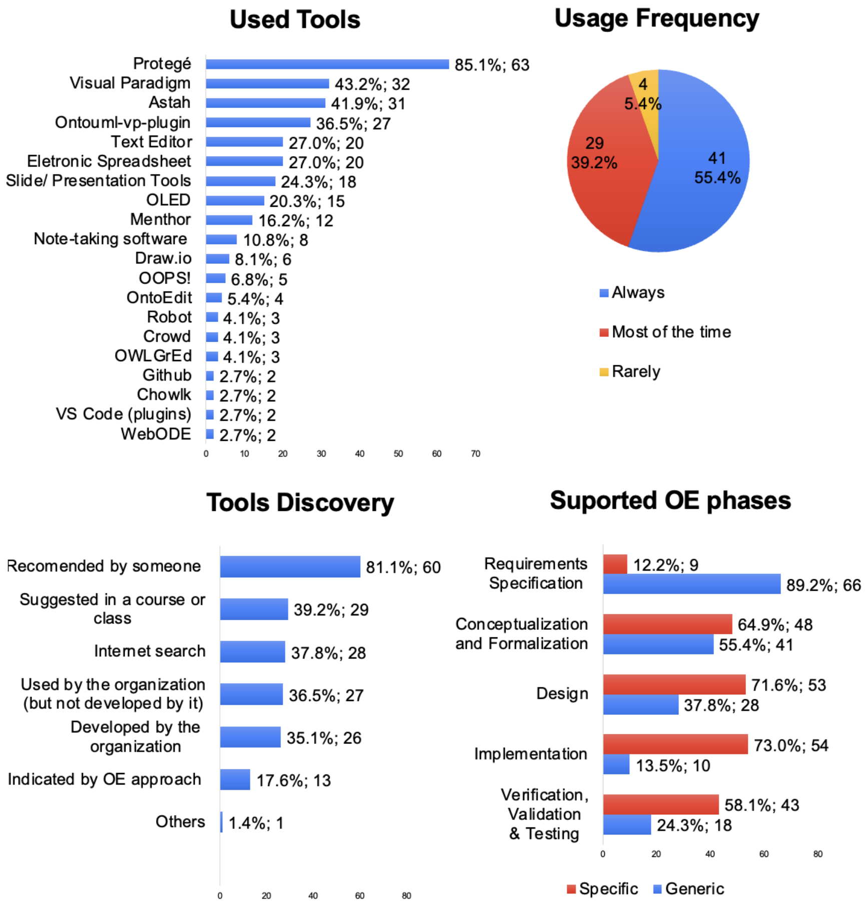

Figure 2: Use of Tools in OE.

[4] https://github.com/OntoUML/ontouml-vp-plugin

OE tools characteristics: In this topic, we asked the participants about the tools' availability, features, and integrated support for OE. Figure 3 summarizes the results. Regarding *availability*, most participants (63.5%) declared that they have used only free tools, 33.8% informed that some of the tools they have used are free, and 2.7% of participants have not used free tools. Concerning *features*, we provided a list of options for the participants to indicate which ones have been offered by the tools they have used. The features most indicated were ontology's concept hierarchy representation (86.5%), ontology visualization (85.1%), diagram creation (82.4%), ontology verification (59.5%), operational ontology implementation (56.8%), and ontology validation (50%). We also asked the participants if the tools have provided *integrated support* for the OE process. 47.3% of participants declared that the tools partially provide integrated support; 41.9% informed that the tools do not provide integrated support, and 10.8% stated that the tools have provided integrated support for the OE process.

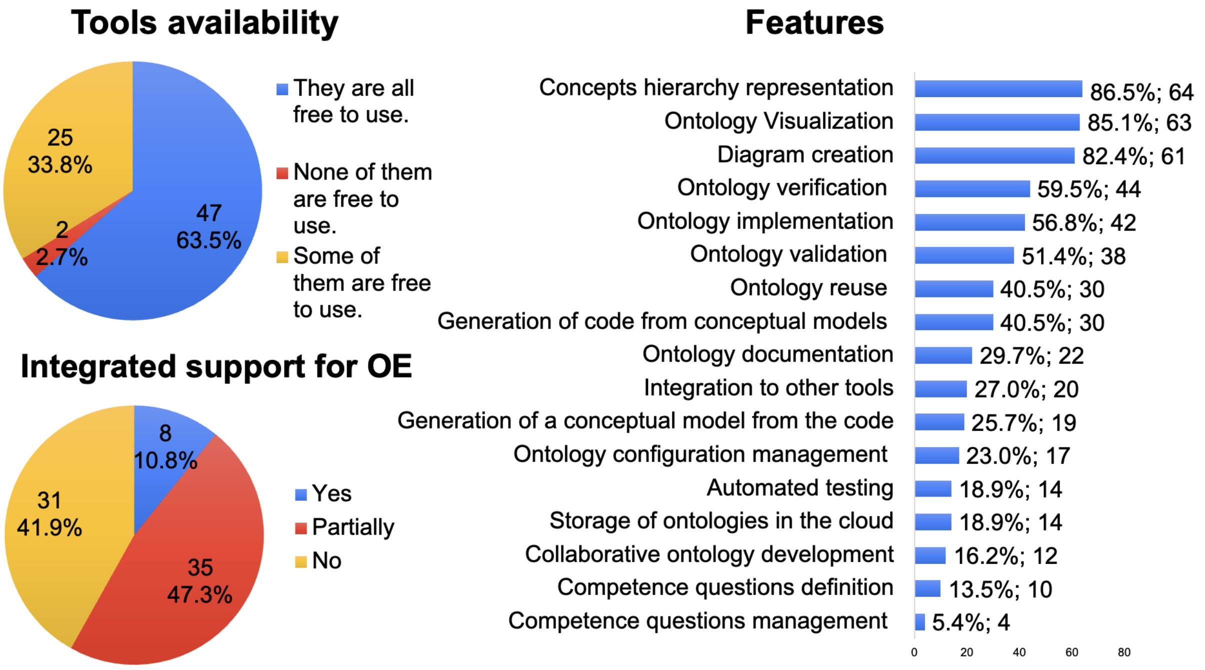

Figure 3: Characteristics of the tools.

Perceptions of using OE tools: In this topic, we asked the participants about their perceptions of using OE tools regarding how useful the tools have been, provided benefits, and faced difficulties. Concerning tools *usefulness*, 43.2% of the participants stated that they have helped a lot; 39.2% answered that they have helped a little; 16.2% declared that they have helped very little; and one participant answered that tools have not helped at all. The participants were asked to justify their answers. We grouped the answers with the same meaning and categorized them. Regarding the reasons why participants have considered tools useful, they highlighted that the tools have helped simplify developing complex ontologies (13.5%), evaluate the ontology (12.2%), automate repetitive tasks (10.8%), visualize and understand the ontology as a whole (8.1%), increase productivity (8.1%), communicate and interact with stakeholders (2.7%), manage knowledge (2.7%) and reuse (2.7%). As for the reasons why participants have not considered the tools useful, they pointed out that the tools are too "fragmented" and the entire OE process cannot be supported by a single tool or a set of integrated tools (29.7%). Additionally, there has been a lack of support, documentation, tutorials, or other educational resources to help them use the tools (17.6%), automation has been limited (5.4%), there have been performance problems (5.4%), lack of free access (1.4%), and complex user interface (1.4%).

We also asked the participants to indicate the benefits and difficulties they have perceived when using the tools. As for *benefits*, they highlighted that the tools accelerate the ontology development process (90.5%) and improve the quality of the produced ontologies (85.1%). Facilitating reuse (66.2%) and

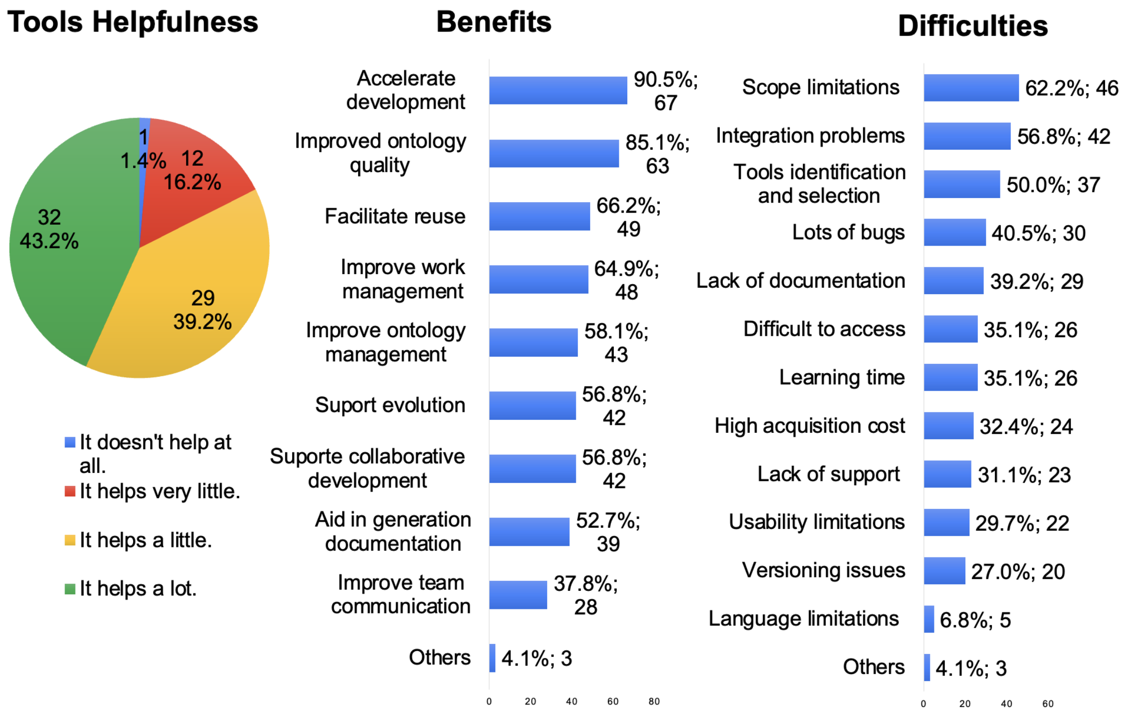

Figure 4: Perceptions of using OE tools.

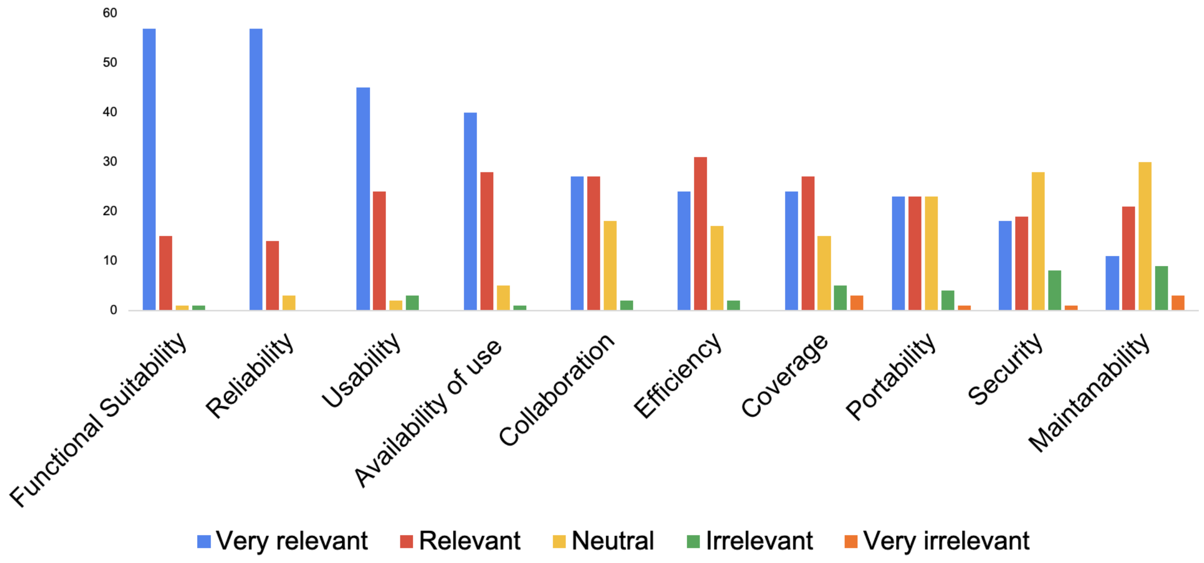

Figure 5: Relevance of OE tools properties.

improving work management (64.9%) have also been acknowledged, among others. As for *difficulties*, the ones more prominent were scope limitations (e.g., the tools support a limited set of OE activities or the provided functionalities have limitations) (62.2%) and integration problems (e.g., the tools do not allow integration with others, the result produced in one tool cannot be exported to another) (56.8%). Identifying and selecting the tools has been difficult for half of the participants. Figure 4 summarizes the participants perceptions.

Properties of OE tool: To investigate which *properties* ontology engineers expect to find in an OE supporting tool, we presented 10 properties (Functional Suitability, Efficiency, Usability, Reliability, Security, Maintainability, Portability, Availability of use, Collaboration, and Coverage) and asked the participants to indicate the level of relevance of each of them. The properties are based on ISO 25010 [37] (we adapted some of them to better fit OE tools). Figure 5 presents the results. Functional Suitability, Reliability, Usability, and Availability of Use were considered by the participants as the most relevant properties. In contrast, Maintainability and Security were considered less relevant (the number of "neutral" plus "irrelevant" plus "very irrelevant" answers is higher than in the other properties). The participants were also asked to indicate other properties they consider important in a tool to support OE. 31 participants answered this question. We grouped the equivalent properties. The main cited properties were Interoperability, Documentation & Support, Version Control, and Reuse.

5. Discussion

Next, we discuss the results by considering the topics identified in Section 4.

Regarding the **use of tools in OE**, around 94.6% of the participants declared that have systematically used tools, which demonstrates the key role tools play in OE. The most used tools were Protégé (85.1%), Visual Paradigm (41.9%), and Astah (36.5%). They support implementing operational ontologies (Protégé) and creating ontology conceptual models (Visual Paradigm and Astah). These are the OE most specific activities and tooling support is crucial to perform them. Thus, it is not a surprise that the most used tools relate to these activities.

By analyzing the ontology types, we noticed that 79.7% of the participants have developed reference ontologies and 67.6% operational ontologies. Although there has been a predominance of reference ontologies, the number of participants that have used Protégé (85.1%) is higher than the ones using Visual Paradigm or Astah (59.5% of the participants have used only one of them or both). We believe that this is because Protégé is free, very popular among ontology engineers, and supported by a strong community of academic, government, and corporate users. In contrast, although Astah and Visual Paradigm are popular, they have similar features, which may divide users (e.g., 23% of the participants have used Visual Paradigm while 17.6% have preferred Astah), and are not free (Astah has a student version and Visual Paradigm has a community version with fewer features than the paid version). OntoUML-vp-plugin was also cited by many participants (36.5%). It helps validate conceptual models created in Visual Paradigm and represented in OntoUML and generates code that implements the corresponding operational ontology. Thus, it provides integrated support to modeling and implementation activities and bridges the ontology conceptual model and its operational version. We believe that tools like this contribute to built-in quality and should be further explored by the OE community. Four participants (5.4%) declared that have used tools rarely. Given that developing ontologies without tool support can be unfeasible, we hypothesize that the participants may have thought that the question referred to OE-specific tools only. That was the first question about tools presented for the participants, thus they were not advised that both, generic and OE-specific tools, should be considered.

By looking at the 10 most cited tools, we notice that six of them are specific tools supporting conceptual modeling and ontology implementation, while four are generic tools. By analyzing in which OE phases the different types of tools have been used, we notice that the more specific the task is, the more specific the tool needs to be. Hence, there has been a predominance of generic tools to support Requirements Specification activities. Given that requirements are usually recorded in text format, generic tools such as text editors and spreadsheets can be helpful. Ontology conceptualization and formalization, design, implementation, and evaluation, in turn, have depended mainly on OE-specific tools. However, the use of generic tools to support ontology conceptualization and formalization (e.g., record concepts definitions and axioms) has also been expressive. Although generic tools have been useful, it may be hard to map information recorded in such tools (e.g., text editors) to the ontology stored in OE-specific tools (e.g., ontology editors). Thus, advances in OE-specific tools to support OE initial phases and connect them to other tools would be welcome.

By analyzing tool support for each OE phase disregarding the types of tools, we noticed that around

92% of the participants have used tools to support Requirements Specification, 86% to support Conceptualization and Formalization, 84% to support Design, 76% to aid in Implementation, and 66% in Verification, Validation, and Testing. It is possible to notice higher values in the three first phases when compared to the last two phases, even though tools are essential to implement operational ontologies. These results are probably because many participants (32.4%) have developed only reference ontologies and, thus, do not implement or test operational ontologies.

There has not been a common structured source of tools (e.g., a catalog, a repository). Most ontology engineers (more than 80%) have discovered the tools mainly through recommendations from other people. This suggests a lack of knowledge of the panorama of existing tools, their main characteristics, features, strengths, weaknesses, and so on. Some ontology engineers (around 17%) have used tools indicated by OE systematic approaches. Given that around 80% of the participants declared that have used such approaches, this result suggests that OE methods have not suggested tools or have suggested limited or unavailable ones, which is in line with observations made in [9].

Concerning **OE tool characteristics**, free tools have been predominant. This result corroborates the literature, which states that free access to the source code and cost savings are among the main characteristics desired in software tools [38, 39, 40]. The features offered by the tools have been diverse. The six most cited features are directly related to OE core phases: diagram creation, concepts hierarchy representation, ontology visualization, ontology implementation, ontology verification, and validation. This demonstrates that the tools have been concerned with providing features to address core OE tasks. Other features have covered support activities, such as reuse, automatic documentation, and configuration management. Features related to requirements (competency questions definition and management) have been the ones less covered by the tools. This is consistent with the findings previously discussed, indicating a lack of support of OE-specific tools for requirements specification. Providing advances to support requirement specification activities and connecting the produced artifacts to outcomes of subsequent phases would be beneficial for the OE community.

Only around 10% of the participants informed that the tools they have used provide integrated support to the OE process. This suggests that despite the range of existing OE tools, they still have not supported the ontology lifecycle in a comprehensive and integrated way. In fact, most OE tools focus on a few activities of the ontology lifecycle, mainly design and implementation [20]. Moreover, the tools often work in an isolated way or with little integration with others. Thus, ontology engineers run them individually, switching between different tools and putting effort into integrating results from different tools [41]. This makes OE activities cumbersome and time-consuming [42], hampering tool support efficiency. Even some widespread OE tools only cover some OE tasks and cannot be used in the sense of a fully-fledged integrated development environment (IDE) [43, 20]. Therefore, there is a lack of integrated suites of tools that address the entire OE process. Analogous to tools that bridge the ontology conceptual model and its operational version, it is important to produce solutions that offer integrated OE support environments or other alternatives (e.g., plugins, APIs) to facilitate the integration of existing tools.

As for the **perceptions of using OE tools**, around 82% of the participants perceive OE tools as helpful. Although this result was in some way expected, almost half of these participants claimed that the tools help just a little. Moreover, around 18% of the participants indicated that the tools help very little or are not helpful. Participants declared that the main problems hindering tool usefulness have been related to integration, reinforcing the issue discussed above, and documentation. Without a proper integration among the several tools used to support OE activities and clear documentation and support to use the tools, the advantages of using them are hampered [42, 19]. On the other hand, the support of the tools in developing complex ontologies, visualizing and evaluating ontologies, as well as automating repetitive tasks have contributed to ontology engineers perceiving the tools as helpful.

The reasons why ontology engineers perceive OE tools as helpful (or not) were declared by the participants in an open question. After that, they were asked about the benefits and difficulties of using OE tools. In general, the results were consistent with the reasons presented by them, providing an overview of the pros and cons of using OE tools. Accelerating the ontology development process and improving the quality of the produced ontologies were the benefits perceived by more participants. These benefits have also been advocated in the literature (e.g., [4, 20, 21]). Hence, the results reinforce them and provide evidence of tools usefulness. Although ontology engineers have several OE tools available, they still have faced difficulties when using them, mainly due to scope limitations and integration problems. Moreover, selecting the tools has been challenging and the tools have presented many bugs. These problems have also been pointed out in the literature (e.g., [19, 44, 45, 20, 43, 46, 42, 41]), revealing that despite the advances in OE tools, there is a need for further improvements. Concerning integration problems, which is a prominent issue revealed in the survey results, it is important to advance works aiming at creating a common workbench for OE that facilitates ontology development during the whole ontology life cycle, also including support to configuration management and ontology evolution [20].

Regarding **properties of OE tools** the four considered more relevant by the participants (Functional Suitability, Reliability, Usability, and Availability) show that ontology engineers expect tools with the capability of doing what they are set out to do (i.e., have the necessary functions to meet the target tasks and objectives [47]), producing correct and reliable results, communicating clearly with the users, and being easily accessed (e.g., free tools). In other words, the tools should correctly do what they are intended to do and it should be easy to access and use them. These properties are perceived mainly from the external perspective (i.e., when ontology engineers are the users of the tools) and are consistent with other results. For example, bugs, limited access, and usability problems were pointed out as difficulties faced by the participants when using the tools, indicating the need for improvements in reliability, availability, and usability and the relevance of such properties. Despite scope limitations having been pointed out by most of the participants as an obstacle to overcome when using the tools, Coverage was not among the top five properties. Security and Maintainability were the properties considered less relevant. Security refers to the ability of the tool to protect data and information. We believe that the participants perceived this property as less relevant because, in general, when developing ontologies, we do not record confidential or sensitive information in the tools. However, protecting the ontology models, concepts, instances, etc. is still necessary, which was recognized by 50% of the participants, who considered this property relevant or very relevant. Maintainability, in turn, concerns the internal perspective, i.e., from who develops the tool. Thus, its relevance would be perceived only by ontology engineers who interact with the tools by playing different roles, user (ontology engineer) and developer (software engineer). The results suggest that this is not the case for most of the participants.

The list of properties we provided to the participants was not exhaustive. To capture other relevant properties, we asked them to cite others they consider relevant. Interoperability, Documentation & Support, Version Control, and (support for) Reuse stand out. All these properties are related to aspects pointed out by the participants in other questions, which reinforces their relevance. Integration problems, lack of documentation & support, and version control issues have been indicated as difficulties faced when using the tools. Reuse, in turn, helps accelerate the development process, which is the main benefit of using tools perceived by the participants. To properly meet ontology engineers expectations and needs, tools should manifest properties referring to desirable quality attributes. The survey results indicate that some of them have not been adequately addressed by existing tools. Therefore, there is an opportunity to explore how these properties should be manifested and develop strategies to achieve them.

6. Threats to Validity

The validity of a study denotes the trustworthiness of the results. Every study has threats that should be addressed as much as possible and considered together with the results. We discuss threats using the classification presented in [33].

Threats related to the constructs involved in the study can affect the results and impact *Construct Validity*. The main threat is the participants misinterpreting the questions. To minimize this threat, after producing the questionnaire, we carried out an internal test in which one of the authors, who did not participate in the questionnaire creation, answered it and pointed out aspects to be improved. After improving the questionnaire, we ran a pilot with an ontology engineer external to our research group to evaluate the questionnaire and response time. Moreover, in the questionnaire, we provided examples and definitions for the participants to better understand how to answer the questions.

Another threat is related to the scale used in some questions. Given that we did not provide any common grounds, different participants may have interpreted terms (e.g., a lot, a little) in different ways. Moreover, the results might have been different if we had not excluded the neutral alternative in some questions. The alternatives of answers provided in each question also represent a threat because they may not cover all the relevant alternatives. To address this threat, when defining the questions and the respective alternatives of answers, we considered results from the systematic literature mapping about tools we are carrying out and, in addition, we included "Others" as an alternative the participants could choose and provide further information. Still regarding construct validity, it is important to be aware that the results reflect the participants' personal experiences, interpretations, and beliefs, and the answers embed subjectivity that could not be captured through the questionnaire.

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 is participants provide inaccurate answers because they think they could be evaluated, since we asked them to inform their email to participate in the study. To mitigate this threat, we informed participants that data would not be evaluated individually and the participants' identities would not be revealed.

Regarding *External Validity*, i.e., to what extent it is possible to generalize the study results, the main threat is the study sample, which comprises 74 ontology engineers and most of them are Brazilians. Ideally, the sample should be larger and the geographic distribution of the study participants more diverse. 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 invite other people freely. Even so, the sample presents limitations. Given that the participants' profile directs influence the results, new studies involving participants with different profiles might produce different results.

Finally, the decisions and data interpretations made by the researchers affect *Reliability Validity*, which regards the influence of the researchers on the results and conclusions. The classification schemas for categorizing data collected in open questions and the classification of data in such categories involve a lot of judgment, as well as analyzing data and getting conclusions. Therefore, other researchers could obtain different results. To minimize this threat, the collected data was initially discussed and analyzed by five authors. Data and analysis results were then presented to the other two authors (more experienced researchers) who reviewed them and discussed the results with the other researchers to refine the conclusions and reach a consensus.

7. Related Work

In the literature there are several works addressing tools for OE. Some of them propose tools, such as WIDOCO [48], a wizard for documenting ontologies, ROBOT [49], a tool to automate workflows, and Chowlk [50], a drawing library and a converter to draw conceptual models and convert them into operational ontologies. Other works compare tools. For example, in [16], Abburu and Babu compared tools (Ontolingua Server, OntoSaurus, OilEd, WebOnto, Protégé, Swoop, TopBraid Composer, WebODE, OntoEdit, and Neon Toolkit) considering a set of features (e.g., versioning, collaborative functions, extensibility, support for large ontologies). Vigo et al. [19], in turn, compared Protégé to its web version, WebProtégé, TopBraid Composer and Swoop against criteria such as situational awareness, ontology overview, search engine, filtering and linking features, ontology retrieval and reuse, among others. Alatrish [51] studied Apollo, OntoEdit, Protégé, Swoop and Top Braid Composer regarding aspects such as software architecture, tool evolution, interoperability, inference services, and usability. Slimani [52] used these same criteria to compare other tools.

A few works have reported surveys investigating OE tools. We highlight the work by Khondoker and Mueller [53], who conducted a survey with 32 practitioners who were asked to choose a tool and perform a modeling task using it. After that, they provide feedback regarding the tool used, considering their perceptions of the tool's user-friendliness, learning time, problems, and their level of satisfaction with the tool, among others. The results showed that although there was a wide range of free tools, the participants chose only a few of them, most OE-specific tools, with Protégé being the predominant.

From the works we found in the literature, we consider [53] the one most related to ours because it consists of a survey with OE practitioners aiming at getting their perceptions about OE supporting tools. In both works, Protégé is the predominant tool and there has been a preference for using OE-specific tools. However, there are some important differences between the work by Khondoker and Mueller [53] and ours. The survey presented in [53] focused on the use of the tools in the conceptual modeling task at hand instead of on the aspects that led the participants to choose the used tools and how they use them. Moreover, in that work, only aspects relevant to the performed modeling task were investigated. In contrast, our work investigates a more comprehensive set of aspects, provides an overview of the use of tools to support OE practice, gathers evidence on this subject, and raises some issues that can be addressed in future research.

8. Final Considerations

This paper presented a survey that investigated the use of tools to support OE. The survey aimed to answer three research questions: (RQ1) Which tools have been used and which are their characteristics?; (RQ2) What benefits and difficulties have been perceived when using the tools?; and (RQ3) Which properties have been considered important in OE tools?

In summary, concerning RQ1, there has been a predominance of OE-specific tools, mainly Protégé, Visual Paradigm, and Astah, which support modeling and implementation activities. Generic tools have been used mainly to support OE initial phases, mostly Requirements Specification. The tools have been discovered mainly through recommendations from other people and there has been a preference for free tools. Tools have not provided adequate integrated support for the OE process. As for RQ2, increasing development process performance and improving ontology quality have been acknowledged as the main benefits. In contrast, integration problems and scope limitations have been the main difficulties faced by ontology engineers. Regarding RQ3, Functional Suitability, Reliability, Usability, and Availability have been considered the most relevant properties.

These results provide a panorama of tool usage in OE and raise some issues that can be addressed in future research. By looking at the pros and cons of using the tools, their weaknesses, and strengths we can set a path to provide advances in the topic. For example, improving the support for ontology requirements specification activities and connecting the produced artifacts to other tools would benefit OE practice. This study can be replicated in other countries, allowing for data comparison and evaluation of its applicability in broader contexts. It is also necessary to produce solutions to enable integrated support for the OE process and deal with scope limitations. Moreover, there is a need to improve the quality of the tools by considering the properties ontology engineers expect to find in them. Therefore, investigating such properties and defining strategies to achieve them would be welcome.

Currently, we are carrying out a systematic mapping of the literature to provide a panorama of the state of the art about OE tools. We intend to analyze the survey and systematic mapping results to obtain a multivocal overview of tool usage in OE and shine a light on the road ahead. Considering the studies' results, we intend to produce a solution (e.g., a catalog, a repository, guidelines) to help ontology engineers search and use tools.

Acknowledgments

We would like to thank everyone who took part in the survey and spread our questionnaire to help us with this study. This research is supported by Coordination for the Improvement of Higher Education Personnel - Brazil (CAPES) - Finance Code 001, and Espírito Santo Research and Innovation Support Foundation (FAPES) - Processes 2023-5L1FC, 2021-GL60J, 2022-NGKM5, and T.O. 1022/2022.

References

[1] M. Uschold, M. Gruninger, Ontologies and semantics for seamless connectivity, ACM SIGMod Record 33 (2004) 58–64.

[2] C. C. Reginato, J. S. Salamon, G. G. Nogueira, M. P. Barcellos, V. E. S. Souza, M. E. Monteiro, R. Guizzardi, A goal-oriented framework for ontology reuse, Applied ontology 17 (2022) 365–399.

[3] D. Oberle, How ontologies benefit enterprise applications 5 (6)(2014), 473–491. doi: 10.3233, SW-130114 (????).

[4] F. M. Mendonça, J. L. Emygdio, L. P. de Castro, E. R. Felipe, Onto4alleditor: a graphic web ontology editor for information science: a graphic web ontology editor for information science, Frontiers of Knowlegde Representation 1 (2021) 70–94.

[5] A. B. Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. García, S. Gil-López, D. Molina, R. Benjamins, et al., Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai, Information fusion 58 (2020) 82–115.

[6] R. Confalonieri, G. Guizzardi, On the multiple roles of ontologies in explainable ai, arXiv preprint arXiv:2311.04778 (2023).

[7] P. S. S. Júnior, M. P. Barcellos, R. de Almeida Falbo, J. P. A. Almeida, From a scrum reference ontology to the integration of applications for data-driven software development, Information and Software Technology 136 (2021) 106570.

[8] E. Blomqvist, K. Hammar, V. Presutti, Engineering ontologies with patterns-the extreme design methodology., Ontology Engineering with Ontology Design Patterns 25 (2016) 23–50. doi:10.3233/978-1-61499-676-7-23.

[9] R. A. Falbo, SABiO: Systematic Approach for Building Ontologies, in: G. Guizzardi, O. Pastor, Y. Wand, S. de Cesare, F. Gailly, M. Lycett, C. Partridge (Eds.), Proc. of the Proceedings of the 1st Joint Workshop ONTO.COM / ODISE on Ontologies in Conceptual Modeling and Information Systems Engineering, CEUR, 2014.

[10] S. Peroni, A simplified agile methodology for ontology development, in: OWL: Experiences and Directions–Reasoner Evaluation: 13th International Workshop, OWLED 2016, and 5th International Workshop, ORE 2016, Bologna, Italy, November 20, 2016, Revised Selected Papers 13, Springer, 2017, pp. 55–69.

[11] 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.

[12] D. Spoladore, E. Pessot, Collaborative ontology engineering methodologies for the development of decision support systems: Case studies in the healthcare domain, Electronics 10 (2021) 1060.

[13] M. Poveda-Villalón, A. Fernández-Izquierdo, M. Fernández-López, R. García-Castro, Lot: An industrial oriented ontology engineering framework, Engineering Applications of Artificial Intelligence 111 (2022) 104755.

[14] T. Tudorache, Ontology engineering: Current state, challenges, and future directions, Semantic Web 11 (2020) 125–138.

[15] S. Rautenberg, J. L. Todesco, A. V. Steil, F. A. Gauthier, Uma metodologia para o desenvolvimento de ontologias, RECEN-Revista Ciências Exatas e Naturais 10 (2008) 237–262.

[16] S. Abburu, G. S. Babu, Survey on ontology construction tools, International Journal of Scientific & Engineering Research 4 (2013) 1748–1752.

[17] R. Iqbal, M. A. A. Murad, A. Mustapha, N. M. Sharef, et al., An analysis of ontology engineering methodologies: A literature review, Research journal of applied sciences, engineering and technology 6 (2013) 2993–3000.

[18] J. Cardoso, The semantic web vision: Where are we?, IEEE Intelligent systems 22 (2007) 84–88.

[19] M. Vigo, S. Bail, C. Jay, R. Stevens, Overcoming the pitfalls of ontology authoring: Strategies and implications for tool design, International Journal of Human-Computer Studies 72 (2014) 835–845.

[20] O. Corcho, M. Fernández-López, A. Gómez-Pérez, Methodologies, tools and languages for building ontologies. where is their meeting point?, Data & knowledge engineering 46 (2003) 41–64.

[21] K. I. Kotis, G. A. Vouros, D. Spiliotopoulos, Ontology engineering methodologies for the evolution

of living and reused ontologies: status, trends, findings and recommendations, The Knowledge Engineering Review 35 (2020) e4. doi:10.1017/S0269888920000065.

[22] R. Studer, V. R. Benjamins, D. Fensel, Knowledge engineering: Principles and methods, Data & knowledge engineering 25 (1998) 161–197.

[23] G. Guizzardi, Conceptualizations, modeling languages, and (meta) models, in: Databases and Information Systems IV: Selected Papers from the Seventh International Baltic Conference, DB&IS'2006, volume 155, IOS Press, 2007, p. 18.

[24] 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.

[25] A. Stadnicki, F. F. Pietroń, P. Burek, Towards a modern ontology development environment, Procedia Computer Science 176 (2020) 753–762.

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

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

[28] O. Corcho, M. Fernandez-Lopez, A. Gomez-Perez, Ontological engineering: what are ontologies and how can we build them?, in: Semantic web services: Theory, tools and applications, IGI Global, 2007, pp. 44–70.

[29] J. Guerson, T. P. Sales, G. Guizzardi, J. P. A. Almeida, Ontouml lightweight editor: a model-based environment to build, evaluate and implement reference ontologies, in: 2015 IEEE 19th international enterprise distributed object computing workshop, IEEE, 2015, pp. 144–147.

[30] C. M. Fonseca, T. P. Sales, V. Viola, L. B. R. da Fonseca, G. Guizzardi, J. P. A. Almeida, Ontology-driven conceptual modeling as a service., in: JOWO, 2021.

[31] A. Gomez-Perez, J. Angele, M. Fernandez-Lopez, V. Christophides, A. Stutt, Y. Sure, et al., A survey on ontology tools, OntoWeb deliverable 1 (2002).

[32] S. Easterbrook, J. Singer, M.-A. Storey, D. Damian, Selecting empirical methods for software engineering research, Guide to advanced empirical software engineering (2008) 285–311.

[33] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, A. Wesslén, Experimentation in software engineering, Springer Science & Business Media, 2012.

[34] C. A. D. Gomes, J. S. Monfardini, Glaice Kelly Q.and Salamon, R. d. S. Sangali1, I. S. R. Timoteo, M. P. Barcellos, V. E. S. Souza, Investigating tool usage in ontology engineering: a survey - protocol & data extraction (2024). doi:10.6084/m9.figshare.25927438.

[35] M. Musen, The protégé project: A look back and a look forward, AI Matters 1 (2015). doi:10.1145/2557001.25757003.

[36] J. L. Moreira, T. P. Sales, J. Guerson, B. F. Braga, F. Brasileiro, V. Sobral, Menthor editor: an ontology driven conceptual modeling platform (2016).

[37] I. O. for Standardization (ISO), Iso/iec 25010:2011 - systems and software engineering – systems and software quality requirements and evaluation (square) – system and software quality models, ISO Standards (2011). URL: https://iso25000.com/index.php/en/iso-25000-standards/iso-25010.

[38] J. Sanders, Linux, open source, and software's future, IEEE software 15 (1998) 88–91.

[39] J. Feller, Perspectives on free and open source software, MIT Press, 2005.

[40] B. Fitzgerald, Has open source software a future? (2005).

[41] L. Zhang, Y. Yu, J. Lu, C. Lin, K. Tu, M. Guo, Z. Zhang, G. Xie, Z. Su, Y. Pan, Orient: Integrate ontology engineering into industry tooling environment, in: The Semantic Web–ISWC 2004: Third International Semantic Web Conference, Hiroshima, Japan, November 7-11, 2004. Proceedings 3, Springer, 2004, pp. 823–837.

[42] A. Alobaid, D. Garijo, M. Poveda-Villalón, I. Santana-Perez, A. Fernández-Izquierdo, O. Corcho, Automating ontology engineering support activities with ontoology, Journal of Web Semantics 57 (2019) 100472.

[43] G. Dziwis, L. Wenige, L.-P. Meyer, M. Martin, Ontoflow: A user-friendly ontology development workflow., in: SemIIM, 2022.

[44] A. Katifori, C. Halatsis, G. Lepouras, C. Vassilakis, E. Giannopoulou, Ontology visualization methods—a survey, ACM Computing Surveys (CSUR) 39 (2007) 10–es.

[45] M. Dzbor, E. Motta, C. Buil, J. Gomez, O. Görlitz, H. Lewen, Developing ontologies in owl: An observational study (2006).

[46] A. Zouaq, R. Nkambou, A survey of domain ontology engineering: Methods and tools, in: Advances in intelligent tutoring systems, Springer, 2010, pp. 103–119.

[47] A. S. Puspaningrum, S. Rochimah, R. J. Akbar, Functional suitability measurement using goal-oriented approach based on iso/iec 25010 for academics information system, Journal of Information Systems Engineering and Business Intelligence 3 (2017) 68–74.

[48] D. Garijo, Widoco: a wizard for documenting ontologies, in: The Semantic Web–ISWC 2017: 16th International Semantic Web Conference, Vienna, Austria, October 21-25, 2017, Proceedings, Part II 16, Springer, 2017, pp. 94–102.

[49] R. C. Jackson, J. P. Balhoff, E. Douglass, N. L. Harris, C. J. Mungall, J. A. Overton, Robot: a tool for automating ontology workflows, BMC bioinformatics 20 (2019) 1–10.

[50] S. Chávez-Feria, R. García-Castro, M. Poveda-Villalón, Chowlk: from uml-based ontology conceptualizations to owl, in: European Semantic Web Conference, Springer, 2022, pp. 338–352.

[51] E. Alatrish, Comparison some of ontology, Journal of Management Information Systems 8 (2013) 018–024.

[52] T. Slimani, Ontology development: A comparing study on tools, languages and formalisms, Indian Journal of Science and Technology 8 (2015).

[53] M. R. Khondoker, P. Mueller, Comparing ontology development tools based on an online survey, World Congress on Engineering 2010 (WCE 2010), London, UK, 2010.