# An Approach for Developing an Ontology: Learned from Business Model Ontology Design and Development

Ahadi Haji Mohd Nasir[1], Mohd Firdaus Sulaiman[2], Liew Kok Leong[3], Ely Salwana[4], Mohammad Nazir Ahmad[5]
Institute of Visual Informatics, Universiti Kebangsaan Malaysia (UKM), Bangi, Selangor, Malaysia[1, 2, 3, 4, 5]
Infrastructure University Kuala Lumpur (IUKL), Kajang, Selangor, Malaysia[5]

***Abstract*—Ontology, serving as an explicit specification of conceptualization, has found widespread applications across various fields. Business Model Ontology (BMO) stands out as a prominent ontology, especially in the domains of business and entrepreneurship. This study employs the narrative literature review method to delve into the Ontology Development Method (ODM). By identifying commonalities among various ODMs and drawing insights from the BMO, the study proposes a Unified Ontology Approach (UOA) as an alternative ODM. The UOA is derived by combining the common characteristics and key steps of various ODMs, aiming to streamline the ontology development process and enhance its effectiveness. Through an extensive analysis of existing methodologies, this research contributes to the field by offering a consolidated perspective on ODMs. The study findings shed light on the strengths and weaknesses of different approaches, facilitating informed decision-making for ontology developers. Furthermore, the discussion explores the implications of adopting the UOA in practical applications, emphasizing its potential to improve ontology quality, interoperability, and adaptability across diverse domains. In conclusion, this paper advocates for the adoption of the UOA as a comprehensive and flexible framework for ontology development. By synthesizing the strengths of existing ODMs and insights from the BMO, the UOA offers a promising avenue for advancing the field of ontology development and driving progress in various domains and applications.**

***Keywords*—Ontology; Ontology Development Method (ODM); Business Model Ontology (BMO); Unified Ontology Approach (UOA)***

## I. Introduction

Ontology, a term originally rooted in philosophy, is defined as an explicit specification of a conceptualization [1]. This concept has been extensively utilized in various fields, including computer science, software engineering, and business. Within the realm of information science, an ontology is characterized as a formal representation of knowledge. It encompasses a set of concepts within a specific domain and the relationships that bind these concepts together [2].

Ontology Development Methodologies (ODMs) are methodologies used to create formal specifications of terms and their relations within a specific domain. These methods facilitate information sharing and reuse across various domains, as highlighted by Gokhale et al. [3]. The process of developing an ontology is multifaceted and iterative, requiring meticulous attention and time [4].

Notably, there's no one-size-fits-all methodology for ontology development, as pointed out by Walisadeera et al., [5] and Yu [4]. This sentiment is echoed by Noy et al. [2] who emphasize that ontology development doesn't adhere to a rigid set of rules or a universally correct approach. The design and development of an ontology are influenced by several factors. These include the nature of the domain in question, the intended application of the ontology, and the ontology developer's perspective [2]. Thus, Ontology development necessitates a harmonious blend of technical expertise and innovative problem-solving, making it both an art and a science.

In the current landscape, a multitude of ontologies have been developed utilizing a wide array of ODMs. These include, but are not limited to, Ontology Development 101, OntoSpec, UPON Lite, Methontology, NeON methodology, Uschold and King Methodology and the new ODMs, Linked Open Terms (LOT) Methodology and Agile Ontology Engineering Methodology (AgiSCont). Each of these methodologies offers unique techniques and perspectives for ontology creation, thereby contributing to the richness and diversity of the field.

Nevertheless, the selection of methodology is not a universally applicable determination. It is influenced by a variety of factors such as the intended application, potential extensions, and the specific use case of the ontology in question. Although it is agreeable that there are varieties of ontological development approaches, the desired outcomes are all the same - developing an ontology. With so many methodologies available, each with its own strengths and nuances, the question arises - is it feasible, or even desirable, to standardize these approaches? Is the approach to ontology solely confined to the use of a single ODM? Or, can it be an amalgamation of various ontology development methods to develop an ontology? Perhaps, by drawing inspiration from a particular ontology development project, a new common version of the ODM could be established. Ultimately, the answer may lie in finding a balance between maintaining methodological diversity and establishing common guidelines to ensure quality and interoperability across different ontologies.

In essence, this paper aims to contribute to the discourse on ontology development methodology by proposing a Unified Ontological Approach (UOA) that integrates the strengths of various ODMs with inspiration drawn from the success and principles of the BMO.

## II. Methodology

This study employs a Narrative Literature Review (NLR) method to provide a comprehensive and interpretative synthesis of existing literature related to ODM. The NLR is a useful method to synthesize a complex and emerging field [6] as well as better suited to addressing a topic in broader ways [7].

The research was carried out in several stages. Initially, a comprehensive literature review was conducted to identify and understand various ODMs. This involved a systematic search of databases and journals for relevant articles, followed by a thorough reading and analysis of these articles.

The environment for this research was prepared by creating a database of all the identified ODMs. This database served as the primary resource for the study. Data was produced through a detailed analysis of the identified ODMs. This involved identifying common steps and practices across different ODMs and drawing insights from the Business Model Ontology (BMO), a globally acclaimed ontology for business setup.

The data validation method involved cross-referencing the identified common steps and practices with the principles of the BMO. This ensured that the synthesized methodology was not only based on the strengths of various ODMs but also aligned with the successful principles of the BMO.

## III. Literature Review

### A. *Understanding Ontology: Definition, Applications, and Importance*

Ontology, in its broadest sense, is the philosophical study of existence or the nature of being, as described by Simon [8]. Salatino et al. [9] further elaborate on this concept, defining ontology as a collection of concepts and categories within a specific subject area or domain that outlines their properties and the relationships between them.

In the realm of computer and information science, the term "ontology" assumes a slightly different meaning. As explained by Gruber [10], in this context, an ontology is an artifact created with a specific purpose - to facilitate the modeling of knowledge about a certain domain, whether it's based on reality or a hypothetical scenario. It provides a specialized vocabulary for formulating statements, which can serve as either inputs or outputs for knowledge agents, such as a software program. Simply put, an ontology can be viewed as a framework that outlines the key concepts, relationships, and other distinctions that are crucial for modeling a domain.

According to Guarino [11], ontologies can significantly impact the main components of an information system, including information resources, user interfaces, and application programs. They provide an effective solution for capturing common knowledge [12] and sharing it [13]. Therefore, ontologies serve as a vital tool for reasoning about entities within a variety of domains and can be effectively employed to describe these domains.

In addition, ontologies are used for several practical reasons. They help in sharing a common understanding of the structure of information, enable the reuse of domain knowledge, make domain assumptions explicit, separate domain knowledge from operational knowledge, and analyze domain knowledge [2].

Fernández-López et al. [14] expand the use of ontologies beyond their traditional applications, leveraging them to enhance communication, collaboration, and decision-making among various stakeholders and systems. The shared understanding facilitated by ontologies can be instrumental in promoting effective communication, fostering collaboration, and guiding decision-making processes across different stakeholders and systems. Such applications underscore the significance and versatility of ontologies in numerous fields, with a particular emphasis on information and computer science.

### B. *Endurant versus Perdurant in Ontology Engineering*

The concept of ontologies is systematized through endurants and perdurants, philosophical positions that address how objects persist over time. Endurants, as defined by Huang [15], are entities that persist wholly at any specific temporal juncture, such as physical objects. Conversely, perdurants, as described by [15], are entities that possess temporal segments and persist through a continuum of time, such as events or processes.

The application of endurant and perdurant can impact the development of ontologies [15], [16]. The significance of perdurant and endurant ontology lies in their ability to categorize entities based on their relationship to time, playing a prominent role in top-level ontologies in information science. The necessity of incorporating both perspectives in an ontology depends on the specific requirements of the domain being modeled [17].

Despite its significance, there are limitations in the application of endurants and perdurants. Huang [15] states that the distinction may not be consistently represented linguistically. Additionally, there is a lack of standard tools to develop perdurant ontologies, suggesting that the distinction may not be adequately supported by existing ontological frameworks [18]. Furthermore, Johansson [19] argued that robust top-level ontologies classifying particulars may need to rely on taxonomic principles other than the endurant-perdurant distinction.

In conclusion, the distinction between endurantism and perdurantism in ontology provides a valuable framework for understanding the temporal aspects of entities. The use of both perspectives offers advantages in accurate representation, modeling structural and dynamic aspects, and supporting diverse applications. However, challenges such as inconsistent linguistic representation and a lack of standard tools for perdurant ontologies exist. The universal applicability and representation of this distinction in linguistic and ontological systems raise questions, emphasizing the need for careful consideration based on specific domain requirements. The choice between endurantism, perdurantism, or a combination should align with the domain's needs for effective knowledge representation. These insights can be instrumental in guiding the design and development of future ontologies.

### C. Business Model Ontology: What We Can Learn?

The Business Model Ontology (BMO), a notable study developed by Alex Osterwalder in 2004, offers valuable insights into the design and development of ontologies. The BMO, which is often applied in the form of the Business Model Canvas (BMC), serves as a strategic management tool that facilitates the description and design of a company's business model [20]. As argued by Holdford et al. [21], this tool has been widely accepted and recognized universally for describing and designing a business enterprise model due to its simplicity, practicality, and effectiveness in guiding the formation of a complete business model [20], [22].

The BMO is structured around nine building blocks: key partners, key activities, key resources, value proposition, customer relationships, channels, customer segments, cost structure, and revenue streams which are simplified by Osterwalder [23] in an ontology framework depicted in Fig. 1 below:

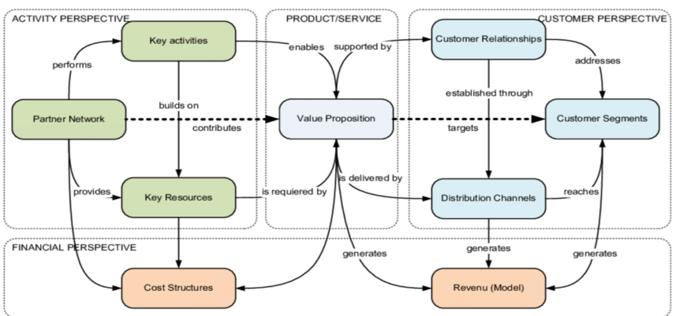


Fig. 1. Business model ontology [23].

Osterwalder [23] does not specifically mention the ODM he employs. The development of the BMO involves six steps starting with a comprehensive literature review. He conducted a comprehensive literature review on the existing definitions and frameworks of business models, as well as the relevant theories and concepts from various disciplines. Based on the information gathered from the literature review, he then proposed a conceptual business model based on the frame-based representation paradigm. The conceptual model is formalized using Web Ontology Language (OWL), which is a standard language for creating ontologies on the Semantic Web. The formalization process involved defining the classes, properties, and axioms of the ontology, as well as the rules and constraints for its instantiation. The validity and usefulness of BMO are evaluated by applying it to several case studies of real-world e-businesses. The evaluation criteria included the completeness, consistency, expressiveness, and simplicity of the ontology, as well as its ability to support analysis and design tasks. The BMO is then documented and the user guide and glossary for the ontology are also published. Based on the building blocks of BMO and depending on how one interprets the concepts and relations in the ontology, BMO is more inclined towards capturing the endurant aspects of a business model, since it focuses on the static and structural elements that define the value proposition, the customer segments, and the business logic.

From an ontology design and development perspective, the BMO provides a shared language and structure that aids entrepreneurs in identifying opportunities for business model innovation [24]. It offers a structured framework for representing and understanding the complexity of business models, systematically encapsulating the fundamental aspects of business models, including the relationships between various components such as actors, resources, and the transfer of resources between actors [21], [25]. This aligns with Osterwalder's [23] original aim for the BMO, which was to formalize the key elements of a business model using an ontology, thereby facilitating the development of sophisticated methods for requirement elicitation and computer-based tools for business model design and analysis.

Moreover, the BMO contributes to enhancing interoperability by providing a common language and framework for describing business models, a feature that is crucial for collaboration and integration between different organizations and systems [25]. Holdford et al., [21] added that this shared framework encourages collaboration and integration between organizations, making it easier to understand and plan business models at a more strategic level. Upward et al. [26] supported this argument and opined that the BMO can be particularly valuable in the context of emerging trends such as business model innovation, digital transformation, and the sharing economy, where the ability to understand and compare different business models is crucial. They further note that the BMO's capability to integrate concepts from strategy, business processes, and information systems underscores the potential for ontologies to bridge interdisciplinary domains and provide a holistic view of complex phenomena such as business models.

In essence, the BMO demonstrates the use of ontology as a common language and framework for describing business models, a feature that is essential for communication, collaboration, and integration between different organizations and systems [25], [26]. Furthermore, the BMO's structured approach enhances modeling capabilities, enabling the representation of both structural and dynamic aspects of business models, leading to a more comprehensive understanding of business phenomena [21], [26].

In conclusion, the BMO provides invaluable insights into the process of understanding, defining, and innovating business models using ontology. Its structured approach and common language foster effective communication and collaboration, establishing it as an effective knowledge representation instrument. The lessons gathered from the BMO underscore the potential of ontology in augmenting the comprehension and innovation of business models across various domains. The BMO insights therefore serve as a testament to the transformative power of ontology in reshaping the understanding of business models and beyond.

### D. Overview of Several ODMs

There are several ODMs available, each with its unique approach to ontology development. An overview of six commonly used ODMs namely Ontology Development 101 (OD101), OntoSpec, Upon Lite, Methontology, NeON methodology, Uschold and King Methodology and the recently developed ODMs, Linked Open Terms (LOT) Methodology and Agile Ontology Engineering Methodology (AgiSCont) will be briefly discussed in this section.

*1) Ontology Development 101 (OD101):* Ontology Development 101 (OD101), introduced by Noy et al. [2] serves as a fundamental guide for novice ontology designers embarking on their first ontology creation. It provides basic-level information on the terms and concepts in a domain by using wine classification as an example. The method employs an iterative approach, beginning with an initial rough draft of the ontology, followed by subsequent revisions and refinements. OD101 as outlined by Noy et al. [2] comprises three key steps: 1) Defining concepts in the domain (classes): This involves identifying the key concepts or classes that are relevant to the domain of interest. 2) Arranging the concepts in a hierarchy (subclass-superclass hierarchy): This key step involves organizing the identified concepts or classes into a hierarchical structure, often in the form of a subclass-superclass hierarchy. 3) Defining which attributes and properties (slots) classes can have and constraints on their values: In this step, the attributes and properties that each class can have are defined, along with any constraints on their values. However, Nie et al. [27] pointed out a limitation of OD101. While it provides a basic guide for creating initial ontologies, it may not offer a comprehensive overview of diverse domains, which is essential for comparing and benchmarking different environments. OD101 primarily considers two similar environments, leading to a lack of definition that allows for comparison and benchmarking against each other. Despite this, OD101 remains important for creating ontologies, which are widely used across various application domains such as biomedical [28] and natural disaster management [29].

*2) OntoSpec:* According to to Kassel [30], OntoSpec is a micro-level Ontology Development Methodology (ODM) that emphasizes formalization aspects. It utilizes highly structured natural language as a specification mode, aiding the builder with the ontological knowledge modeling step, upstream of the formal representation and knowledge implementation steps. Similar to Ontology Development 101 (OD101), OntoSpec involves an iterative process comprising four key steps: 1) identifying the entities (concepts and relations), 2) modeling the properties characterizing the entities through successive refinements, 3) formalizing the ontology using a formal representation language, and 4) Evaluating and validating the ontology [30]. OntoSpec provides a modeling framework that allows ontology builders to define conceptual entities (concepts and relations) composing the ontology through successive refinements. The general principle of OntoSpec involves identifying ever more precise roles defined in a generalization/specialization taxonomy, while considering the structure of the properties in question. OntoSpec is independent of formal representation languages, which makes its definitions universally understandable. This allows domain experts and future users of the ontology to collaborate with the builder in evaluating the modeling choices and the quality of the resulting definitions. However, OntoSpec's use of semi-informal language may limit its applicability in contexts that require strict formalization or the use of specific formal representation languages [30]. It focuses on the details of formalization rather than the broader process of ontology development [31]. Despite this, OntoSpec remains a valuable ODM widely used across various application domains such as neurology [32] and business process [33].

*3) UPON Lite:* UPON Lite, as described by Nicola et al. [31], is a methodology for rapid ontology engineering, derived from the Unified Process for Ontology building (UPON). It is designed to be accessible to domain experts, with minimal intervention from ontology engineers, and focuses on delivering formal ontology. As noted by Lille et al. [34], this method consists of key steps: 1) building the domain terminology lexicon, 2) associating domain terms with descriptions and possible synonyms, 3) organizing the domain terms in an ISA hierarchy, and 4) producing a formally encoded ontology that contains conceptual knowledge collected in the previous steps. UPON Lite's main advantage according to Nicola et al. [31] is its ability to allow a wide base of users, typically domain experts, to construct an ontology largely without the help of ontology engineers. Only in the last step, after domain content is elicited, organized, and validated, the ontology engineers intervene is needed to deliver a final ontology formalization before releasing it to users. This approach provides a well-defined enrichment to each preceding step and disintermediates ontology engineers, making it easier and faster for end-users to create usable ontologies with more efficient collaboration between domain experts and ontology engineers. However, UPON Lite does have some limitations. Lille et al. [34] highlighted that one of the key limitations is that it may not offer a comprehensive overview of diverse domains, which is essential for comparing and benchmarking different environments. UPON Lite primarily considers two similar environments, leading to a lack of definition that allows for comparison and benchmarking against each other. Another limitation underscored by Lille et al. [34] is that despite the existing scientific literature reports on practical applications of UPON Lite in several domains, the detailed elaboration of the development process is limited. This could potentially limit its reproducibility in an actual business context. Despite these limitations, UPON Lite continues to be relevant for Ontology Engineering, particularly for domain experts due to its ease of use and reduced dependence on ontology engineers. It is used across various application domains such as smart building [35] and social networks [36].

*4) Methontology:* According to Fernandez Lopez et al. [37], Methontology an ODM that stresses the importance of reusing and reengineering existing ontologies and knowledge resources. Methontology provides a systematic approach to ontology development, which lead to the creation of more effective and easier to maintain over time ontologies [37]. Fernandez Lopez et al. [37] state that this ODM proposes a set of guidelines and best practices for identifying and evaluating existing ontologies, determining how they can be reused or reengineered to meet the needs of a new ontology development project. Fernandez Lopez et al. [37] outline the best practices in Methontology include reusing existing ontologies, carefully capturing domain concepts and relationships, using formal language, evaluating the ontology's quality, and thorough documentation. The six key steps in Methontology include: 1) identifying the purpose of the ontology and its intended uses, 2) capturing and building the ontology, 3) implementing and testing the ontology, 4) Evaluating the ontology, 5) documenting the ontology after each phase and 6. Maintaining and evolving the ontology [37]. Fernandez Lopez et al. [37] highlight that these steps are not necessarily sequential and can occur concurrently or iteratively. Evaluations should occur throughout the process to ensure continuous improvement of the ontology [38]. Due to its comprehensive and systematic approach to ontology development, which includes various phases such as requirements elicitation and analysis, conceptualization, integration, implementation, and maintenance, Methontology, therefore, is often used for developing heavyweight ontology [3], [39]. Fernandez Lopez et al. [37] argue that the comprehensive approach of Methontology leads to high-quality ontologies that are well-aligned with the needs of the intended users and easier to maintain over time). Nonetheless, despite its breadth, Methontology has limitations. It necessitates more time and effort than other, less comprehensive ODMs [37], [38], [39]. Nevertheless, Methontology continues to be a well-established and influential ODM. It has been successfully utilized in diverse fields, including chemistry [38] and legal [40].

*5) NeOn methodology:* The NeOn Methodology, as explained by Suárez-Figueroa et al. [41], [42], is a scenario-based ODM that focuses on the construction of ontology networks. It promotes collaborative ontology development and emphasizes the reuse and re-engineering of knowledge resources. Unlike other methodologies that enforce a strict workflow, NeOn offers flexibility, accommodating a range of scenarios including reengineering, alignment, modularization, localization, and integration with non-ontological resources [43]. According to Suárez-Figueroa et al. [41], the NeOn Methodology framework is built upon four pillars: The NeOn Glossary, ontology-building scenarios, methodological guidelines, and guidelines for ontology evaluation and evolution. It involves six main steps: 1) ontology requirements specification, 2) ontology analysis, 3) ontology design, 4) ontology development, 5) ontology evaluation, and 6) ontology evolution. These steps are scenario-driven and can be customized to meet the specific characteristics and requirements of each scenario [42]. Interestingly, the NeOn Methodology framework is flexible and customizable based on the specific needs of ontology engineers and software developers for different scenarios. This adaptability as noted by Suárez-Figueroa et al. [42] is a key strength of the NeOn Methodology, making it suitable for a wide range of ontology engineering contexts. However, Suárez-Figueroa et al. [40] highlighted that a limitation of this methodology is that it does not explicitly state all the steps to be performed, and its application can be time-consuming. Despite this, the NeOn Methodology has been flexibly applied in various domains such as education [44] and tourism [45].

*6) Uschold and king methodology:* The Uschold and King Methodology developed by Uschold et al. [46] is an ODM that emphasizes the systematic development of ontologies which includes identifying the appropriate content, relationships, and structuring for the ontology, as well as establishing a process for ontology development and evaluation [47]. The methodology is centered on four distinct steps: 1) identifying the purpose, 2) building the ontology, 3) evaluating the ontology, and 4) documenting the ontology. It provides a set of techniques, methods, and principles for each phase to produce high-quality ontologies [46]. Uschold et al. [46] outlined the key steps in this ODM which include identifying the purpose and scope of the ontology, building the ontology, evaluating the ontology's quality, consistency, and completeness, and documenting the ontology. This ODM involves a comprehensive and systematic approach, making it particularly suitable for complex domains where precision and detail are required [39], [48]. However, despite its comprehensive nature, it lacking in terms of the need for motivating scenarios to guide the construction process, limited user engagement throughout the ontology creation process, and potential inadaptability to all ontology development requirements [39], [48]. Nonetheless, this ODM remains a valuable tool for Ontology Engineering, particularly for domain experts, due to its ease of use and reduced dependence on ontology engineers. It has been successfully applied in various domains, including e-government [49] and education [50].

*7) LOT (Linked Open Terms) methodology:* The LOT (Linked Open Terms) Methodology, as described by Poveda-Villalón et al. [51] is a method for developing ontologies and vocabularies focusing on industry projects. It emphasizes alignment with software development, integrating ontology development into the software industry to promote interoperability between different systems by providing well-documented and consistent standards for information exchange and reuse [51]. Unlike other methodologies that enforce a strict workflow, LOT allows for the adoption of the method in different contexts and needs, offering flexibility, and accommodating a range of scenarios including requirements specification, ontology implementation, ontology publication, and ontology maintenance. According to Poveda-Villalón et al. [51] the LOT Methodology framework is built upon four pillars: 1) the LOT Glossary, 2) ontology-building scenarios, methodological guidelines, and 3) guidelines for ontology evaluation and evolution. It involves four main steps: 1) ontology requirements specification, 2) ontology implementation, 3) ontology publication, and 4) ontology maintenance. These steps are scenario-driven and can be customized to meet the specific characteristics and requirements of each scenario. Interestingly, the LOT Methodology framework is flexible and customizable based on the specific needs of ontology engineers and software developers for different scenarios. This adaptability as noted by Poveda-Villalón et al. [51] is a key strength of the LOT Methodology, making it suitable for a wide range of ontology engineering contexts and aims to serve as a reference framework that can be tailored to meet the specific needs of each project and context. This methodology however does not explicitly state all the steps to be performed, and its application can be time-consuming. According to Poveda-Villalón et al. [51], this, limitation is inherent in any methodology that aims to be flexible and adaptable to different contexts. Despite this limitation, the LOT methodology has been flexibly applied in various domains such as VICINITY, DELTA, BIMERR, and Ciudades Abiertas, demonstrating its potential for use in various contexts.

*8) Agile ontology engineering methodology (AgiSCOnt):* The Agile Ontology Engineering Methodology (AgiSCOnt), as explained by Spoladore et al. [52] is a novel approach that supports organizations in collaborative ontology development. It is a recent ODM developed to support ontologists, especially novices, through the ontology development workflow in an iterative, customizable, and flexible manner while promoting collaboration with domain experts [52]. Similar to LOT, AgiSCOnt is designed to accommodate differing levels of technical expertise among ontology engineers and it is highly iterative, customizable, and flexible, allowing ontologists to tailor the approach to their particular needs and contexts [52]. There are five main steps involved in AgisCOnt: 1) Defining the scope and objectives of the ontology, 2) Selecting the ontological language and expressivity, 3) Identifying the most appropriate Ontology Design Patterns (ODPs) for the domain, 4) Building the ontology iteratively and, 5) Evaluating the ontology against use case scenarios. The possible limitation of AgiSCOnt is that some of the steps are not comprehensively described, resulting in some level of subjectivity in the implementation of the methodology, which can potentially lead to inconsistencies. However, due to its adaptability and flexibility, this method has been applied favorably across domains of knowledge.

*9) Other ODMs:* In addition to the above-established and new ODMs, there are also studies on ontology development using a customized ODM. Youcef et al. [53] introduced an ODM founded on two philosophically grounded foundational ontologies, UFO [54], [55] and DEMO [56] to offer a clear and consistent representation of domain knowledge for virtual reality training in ophthalmology known as OntoPhaco. This ODM spans crucial phases to create reusable, localized, and shareable ontologies for the domain through IWs. There are five key steps involved: 1) Pre-conceptualization- select domain, scope, and range), 2) Conceptualization – analyze and construct classes, relationships, and axioms, 3) Implementation - encode ontology in a knowledge representation language, 4) Ontology evaluation - assess suitability for intended use, 5) Ontology maintenance - review and improve the structure, expand the scope, and refine documentation. Based on the comprehensiveness of OntoPhaco developed using this ODM, its application demands significant expertise and effort and may not be well-suited for less-defined or dynamic domains.

In contrast, Sattar et al. [57] advocate for an enhanced ODM rooted in the Design Science Research Methodology (DSRM), comprising six steps: 1) requirement identification, 2) conceptualization, 3) implementation, 4) evaluation, 5) documentation, and 6) maintenance. This improved ODM underscores collaborative ontology development practices, aligns ontologies with business goals, and integrates agile development principles. It applies to any domain involving IWs. Both ODMs share heavyweight characteristics, embodying a rigorous and comprehensive approach, as suggested by Femi Aminu et al. [58], rendering them more suitable for the development of intricate ontologies.

In a nutshell, although both studies employ a customized ODM to cater to a specific requirement of their ontology development works, the characteristics and the steps involved are not too distinct from common and established ODMs.

### E. ODMs Categorization

Studer et al. [59] argue that ontologies vary in formality and coverage of formal language elements, leading to the categorization as lightweight or heavyweight. This categorization often focuses on the ontology (the product) rather than categorizing the methodology employed to develop it. Most ODMs do not explicitly mention their categorization as either lightweight or heavyweight, except for the Linked Open Terms (LOT) methodology [51], which directly reveals that they are lightweight ontology, their characteristic can be analyzed to determine their category. According to Corcho [60], the difference between lightweight and heavyweight ontologies is determined based on their formalization degree and completeness of the included components. Studer et al. [59] contend that a lightweight ontology provides basic-level information on the terms and concepts in a domain, while a heavyweight ontology explicitly represents more complex relationships, such as part-whole relationships and inheritance hierarchies. Similarly, Corcho [60] and Fernandez-Lopez et al. [14] describe lightweight ontologies as less formal and include fewer formal axioms and constraints, whilst heavyweight ontologies are more formal and have many formal axioms and constraints. The distinguishing characteristics between lightweight and heavyweight are suggested by Lassila et al. [61] presented in Fig. 2. It shows that ontologies can vary in their degree of formality and expressivity, ranging from very lightweight, almost casual ontologies to heavyweight ontologies with many formal rules and restrictions.

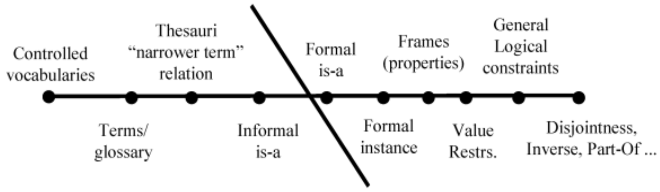


Fig. 2. Lightweight vs heavyweight ontologies characteristics [59].

Lassila et al. [61] opine that lightweight ontologies are easier to understand and share, and that they can grow into useful ontologies through a process of natural selection. In contrast, they state that heavyweight ontologies are more complex, comprehensive, and formal.

Therefore, after synthesizing the characteristics of the ontology and various ODMs discussed briefly in Section 2.4, it can be concluded that in developing a lightweight ontology, the employed ODMs tend to be simpler and more accessible. Conversely, for heavyweight ontologies, the ODMs used are more detailed and comprehensive [58].

Based on the above arguments, the common characteristic distinguishing lightweight and heavyweight ODMs therefore can be summarized in Table I below:

In Section D, eight ODMs have been briefly reviewed. Based on the common characteristics of the lightweight and heavyweight summarized in Table I above, the six ODMs therefore can be categorized as elucidated in Table II below:

In conclusion, discerning whether an ODM is lightweight, heavyweight or a combination of both is imperative in any ontology development project, as it influences resource allocation, project planning, skill assessment, scope definition, usability assurance, flexibility evaluation, reusability consideration, and cost understanding, ensuring the project aligns effectively with its objectives and requirements, ultimately contributing to the development of an effective ontology. Despite distinct characteristics and the key steps involved, all methods share the common goal of creating structured ontologies.

TABLE I. COMMON CHARACTERISTICS DISTINGUISHING LIGHTWEIGHT AND HEAVYWEIGHT ODMS

| Category | Common Characteristics |
|---|---|
| Lightweight | ▪ Provide basic-level information on terms and concepts in a domain.<br>▪ Are less formal, involving fewer formal axioms and constraints.<br>▪ Tend to be simpler, more accessible, and suitable for novice ontology designers.<br>▪ Emphasize ease of use, accessibility for domain experts, and rapid ontology engineering. |
| Heavyweight | ▪ Explicitly represent complex relationships, such as part-whole relationships and inheritance hierarchies.<br>▪ Are more formal, with many formal axioms and constraints.<br>▪ Take a comprehensive and systematic approach to ontology development.<br>▪ Involve detailed and extensive processes, including requirements elicitation, conceptualization, integration, implementation, and maintenance.<br>▪ Are suitable for complex domains where precision and detail are required. |

TABLE II. CATEGORIZATION OF EIGHT ODMS DISCUSSED IN SECTION D

| ODM | Category | Characteristic | Key Step |
|---|---|---|---|
| Ontology Development 101 (OD101) | Lightweight | ▪ Basic-level information<br>▪ Iterative approach<br>▪ Suitable for novices | 1. Defining concepts in the domain (classes)<br>2. Arranging the concepts in a hierarchy (subclass-superclass hierarchy)<br>3. Defining which attributes and properties (slots) classes can have and constraints on their values |
| OntoSpec | Lightweight | ▪ Emphasizes formalization<br>▪ Less formal<br>▪ Uses structured natural language | 1. Identifying the entities (concepts and relations) composing the ontology<br>2. Modeling the properties characterizing the entities through successive refinements<br>3. Formalizing the ontology using a formal representation language<br>4. Evaluating and validating the ontology |
| UPON Lite | Lightweight | ▪ Rapid engineering<br>▪ Accessible to domain experts<br>▪ Minimal intervention from ontology engineers during the ontology development process | 1. Building the domain terminology lexicon<br>2. Associating domain terms with descriptions and possible synonyms<br>3. Organizing the domain terms in an ISA hierarchy<br>4. Producing a formally encoded ontology that contains conceptual knowledge collected in the previous steps |
| Methontology | Heavyweight | ▪ Involves a systematic approach<br>▪ Suitable for a more complex ontology development | 1. Identifying the purpose of the ontology and its intended uses<br>2. Capturing and building the ontology |

| | | | |
|---|---|---|---|
| | | | 3. Implementing and testing the ontology<br>4. Evaluating the ontology<br>5. Documenting the ontology<br>6. Maintaining and evolving the ontology |
| NeOn Methodology | Combination | ▪ Scenario-based<br>▪ Flexible processes<br>▪ Combines lightweight and heavyweight characteristics | 1. Ontology requirements specification<br>2. Ontology analysis<br>3. Ontology design<br>4. Ontology development<br>5. Ontology evaluation<br>6. Ontology evolution |
| Uschold and King Methodology | Heavyweight | ▪ Involves a systematic approach<br>▪ Includes comprehensive phases<br>▪ Suitable for a more complex ontology development | 1. Identifying the purpose and scope of the ontology<br>2. Building the ontology<br>3. Evaluating the ontology's quality, consistency, and completeness<br>4. Documenting the ontology |
| LOT (Linked Open Terms) Methodology | Lightweight | ▪ Focuses on flexibility and adaptability.<br>▪ Involves a detailed view of ontology requirements specification.<br>. | 1. Ontology requirements specification<br>2. Ontology implementation<br>3. Ontology publication<br>4. Ontology maintenance |
| Agile Ontology Engineering Methodology (AgiSCont) | Lightweight | ▪ Focuses on flexibility and adaptability.<br>▪ Supports iterative development.<br>▪ Fits the various ontology activities into the phases of the Scrum agile methodology | 1. Defining the scope and objectives of the ontology<br>2. Selecting the ontological language and expressivity<br>3. Identifying the most appropriate Ontology Design Patterns (ODPs) for the domain<br>4. Building the ontology iteratively<br>5. Evaluating the ontology against use case scenarios |

### F. Synthesizing BMO and ODMs

Synthesizing BMO and ODMs involves recognizing BMO's simplicity and widespread use globally. Although Osterwalder [23] does not explicitly mention the ODM he employs in developing the BMO, the characteristics embedded in BMO, such as simplicity, logic, and ease of comprehension, align closely with the characteristic of a lightweight ontology, establishing a universal language and framework for articulating and scrutinizing business models [25]. Osterwalder [23] approach, steering clear of too many rules or complexity, is very similar to lightweight ODM [14], [59], [60].

Leveraging from BMO, we can derive lessons for ODMs, emphasizing the importance of simplicity, accessibility, and comprehensibility, particularly for novice users. By embracing the straightforward and logical methodology as employed in BMO, the development of more widely applicable, reliable, and actionable ontologies can be facilitated. The BMO's global acceptance underscores that a lightweight ontology can be effective and useful for capturing the essential aspects of a complex domain, without imposing unnecessary complexity or constraints. Chungyalpa et al. [25] further posit that BMO also shows that a lightweight ontology can be easy to use and understand, even for non-experts, by using a graphical notation and a clear structure. BMO also exemplifies the adaptability and extensibility inherent in lightweight ontologies, allowing customization and integration with other ontologies or models.

## IV. Proposed Approach

### A. Key Takeaways from BMO

BMO serves as a paradigmatic lightweight ontology, offering inspiration for the development of ODMs characterized by simplicity, accessibility, and comprehensibility, without compromising reliability and actionability. Standardizing certain aspects, including notation, structure, and evaluation criteria, while permitting flexibility for customization such as terminology, granularity, and integration options, may strike a harmonious balance between uniformity and adaptability in ODMs. The key takeaways from BMO are summarized in Table III below:

TABLE III. The Key Takeaways From BMO

| Key Takeaways from BMO | Description |
|---|---|
| 1. Simplicity and accessibility | BMO demonstrates that a lightweight ontology can be user-friendly for non-experts. |
| 2. Comprehensibility and reliability | BMO's logical structure highlights the importance of reliable and actionable ontologies. |
| 3. Balancing uniformity and adaptability | Standardizing elements while allowing customization aims to balance uniformity and adaptability. |
| 4. Global acceptance as a model | BMO's global acceptance suggests that lightweight ontologies can capture diverse business models. |

### B. Uniformity of the Ontological Approach based on BMO and Various ODMs

To achieve a harmonious balance between simplicity, accessibility, and comprehensibility, without compromising reliability and actionability, a unified approach should assimilate common characteristics from diverse ODMs as the best practices. These characteristic are extracted from BMO and the various ODMs discussed in Section II (D). Table IV below summarizes the common elements that can be integrated as the best practices into a UOA.

A Unified Ontological Approach (UOA) should strive to strike a balance between common elements and adaptability, acknowledging that ontology development is both an art and a science. This balance ensures that the approach remains versatile across diverse domains [2]. This proposed approach, with its emphasis on a harmonious balance between commonality and adaptability, aspires to foster a shared understanding and effective communication within the ontology development community [23]. The integration of insights from BMO and various ODMs paves the way for a more versatile, efficient, and collaborative ontology development approach, with the potential to benefit a multitude of application domains.

TABLE IV. SUMMARY OF THE COMMON CHARACTERISTICS EXTRACTED FROM BMO AND THE VARIOUS ODMS

| Common element | Description |
|---|---|
| 1. Iterative process | Adopt an iterative process inspired by BMO and ODMs, which allows continuous refinement and adaptation to evolving domain requirements. |
| 2. Consistent notation system | Choose a consistent notation system based on a formal language, such as OWL or RDF, which enables unambiguous representation and reasoning of ontological knowledge. |
| 3. Flexible formalization | Apply a flexible formalization approach that supports both domain experts and ontology engineers in building ontologies from scratch or reusing existing ones, such as UPON Lite, LOT and AgiSCOnt. |
| 4. Reusability and reengineering | Follow principles of reusability and reengineering, as suggested by Methontology, which enhance efficiency and effective maintenance of ontologies over time. |
| 5. Scenario-driven development | Utilize scenario-driven development principles from the NeOn Methodology, LOT and AgiSCont which enable customization and collaboration in ontology engineering based on common situations, such as reusing, reengineering, merging, localizing, and integrating ontologies and non-ontological resources. |
| 6. Comprehensive Structure | Provide a comprehensive and systematic structure that addresses complex domains with precision and detail, such as Uschold and King Methodology. |

### C. Towards a Unified Ontological Approach

Identifying the common characteristics, and key steps of each ODM, and an insight learned from the BMO is vital towards a UOA. In this section, all these aspects will be synthesized and integrated to form a UOA as other instances of ODM.

*1) Common characteristics:* In the context of the proposed UOA to ontology development, a seamless integration of common characteristics from various ODMs and insights from BMO is essential. The integration of these elements aims to capitalize on the strengths of different methodologies while addressing their specific limitations, ensuring a comprehensive and versatile approach. Based on Table IV, Fig. 3 below illustrates the common element framework of the ODMs.

A brief explanation of the six common elements framework as illustrated in Fig. 3 is outlined below:

*a) Iterative process:* At the core element of ODM is an iterative process. This element draws inspiration from the BMO proposed by Osterwalder [23] and OD101 introduced by Noy et al. [2], OntoSpec [30], UPON Lite [31], Methontology [37], NeOn Methodology [42], [43], and Uschold and King Methodology [46] acknowledging the significance of continuous refinement and adaptation to evolving domain requirements. ODM is not fixed, but rather dynamic and adaptable, as they reflect the changing needs and challenges of ontology engineering practice as argued by Elhassouni et al. [62]. The iterative nature according to Noy et al. [2] and Espinoza et al. [63] ensures that the ontology remains dynamic, responsive to changes, and refined over time, aligning with the evolving nature of various application domains.

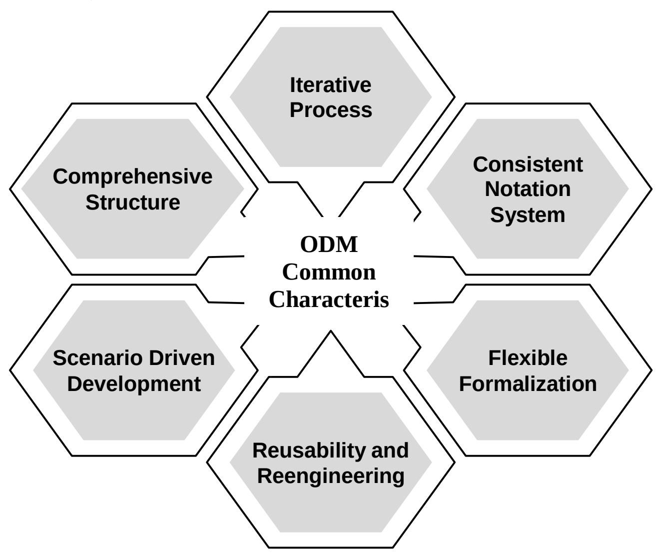


Fig. 3. The common characteristics framework of the ODM.

*b) Consistent notation system:* The adoption of a consistent notation system is imperative element for universal understanding of the ontology without sacrificing formality [64]. In turn, it will help different stakeholders, such as domain experts and ontology engineers to understand the ontological knowledge clearly [65]. A consistent notation system is based on a formal language, such as OWL, RDF, or SKOS, used in different ODMs, such as in BMO [23], NeOn Methodology [42] and UPON Lite [34]. This formal language, according to Gruber [64] and Fernandez Lopez [65] ensures that the ontology can be effectively communicated across diverse stakeholders, including both domain experts and ontology engineers, facilitating clear and unambiguous representation and reasoning of ontological knowledge.

*c) Flexible formalization:* To accommodate both domain experts and ontology engineers, the common element applies a flexible formalization approach, inspired by UPON Lite [34], LOT [51] and AgiSCOnt [52]. This element allows for a balance between precision and accessibility, catering to the diverse expertise levels of stakeholders involved in the ontology development process. The flexibility in formalization means that the ontology can be expressed in different levels of detail and formality, depending on the needs and preferences of the users and the application domain [65]. Verbert et al. [66] and Schlenoff [67] argue that flexible formalization will ensure widespread applicability across various domains, making the approach more inclusive and adaptable.

*d) Reusability and reengineering:* Derived from Methontology, the next common element is reusability and reengineering [37]. Fernandez-Lopez et al. [14] and Villazon-Terrazas et al. [68] underscore the importance of leveraging existing ontologies and knowledge resources, promoting efficiency and effective maintenance over time by avoiding redundant efforts. By integrating insights from various ODMs, this element strives to streamline the development process and enhance the quality of ontologies through systematic reuse [62], [69].

*e) Scenario-driven development:* Scenario-driven development principles are another important common element adopted from NeOn Methodology. This element, according to Suárez-Figueroa et al. [42] enables customization based on specific scenarios while promoting collaborative ontology construction. Recognizing the varied contexts in which ontologies are applied, scenario-driven development enhances the relevance and applicability of the ontology in real-world situations [66], [70].

*f) Comprehensive structure:* Aligned with the Uschold and King Methodology [46], a comprehensive and systematic structure in ontology development is another vital element in ODM. This element, as emphasized by Fernandez-Lopez [14] and Verbert et al. [66] is particularly crucial for addressing the intricacies of complex domains, ensuring that the ontology captures the structural and dynamic aspects with precision and detail. The comprehensive structure enhances the depth of representation, contributing to a more nuanced understanding of business phenomena and other complex domains [65], [66].

*2) Common steps:* In Table II, apart from the categorization of the ODMs and their characteristics, there is also a summary of the key steps involved in each ODM. Based on the various key steps employed in ODM, several common steps are identified among the lightweight and heavyweight ODMs that could be unified. The common steps of the ODM based on all six ODMs summarized in Table II are depicted in Fig. 4 below:

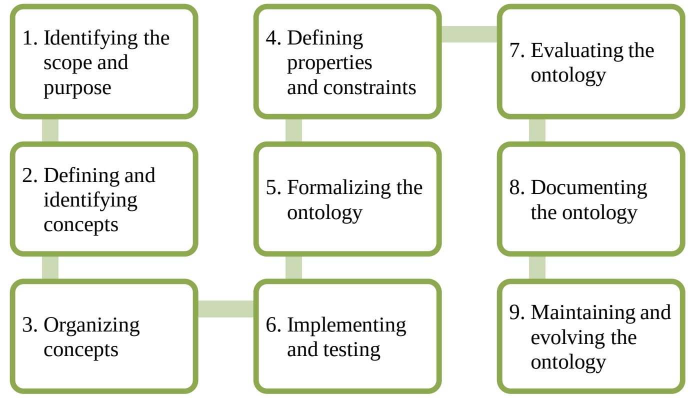


Fig. 4. The common steps of the ODM.

A brief explanation of the six common steps depicted in Fig. 4 is expounded below:

*a) Identifying the purpose and scope:* Identifying the purpose and scope is the key starting point in any ontology development [71], [72]. This step involves understanding why the ontology is being built and what it will be used for. It also includes defining the scope of the ontology, i.e., what concepts it will cover and what level of detail it will provide. Identifying the purpose and scope will ensure that the right ontology is developed in the right way in accordance with the intended user's needs. This process is crucial as it provides direction and focus to the ontology development process, helps to avoid unnecessary work, ensures that the resulting ontology is useful and relevant to its intended users, makes the ontology development project more manageable, and helps define its role in the larger ecosystem of ontologies.

*b) Defining and identifying concepts:* Defining and identifying concepts is a vital step in ontology development. This step includes identifying the main concepts (or classes) that exist in the domain that the ontology is covering [71], [73]. These concepts are the building blocks of the ontology. This step is crucial as it lays the foundation for the ontology. At this stage, an upper ontology can be applied to provide a set of general concepts that can be used to define your specific domain concepts. The use of an upper ontology helps ensure consistency and persistence in the usage of terms, which is crucial for the accuracy and completeness of the identified and defined concepts. The most prominent role of formal ontologies, such as an upper ontology, is to provide a skeleton or common system for ontologies to be developed, provide rich semantics for knowledge representation systems, and enhance ontological adequacy and accuracy. This approach has been demonstrated by Sattar et al. [74] and Youcef et al. [53]. The accuracy and completeness of the identified and defined concepts directly impact the usefulness and applicability of the ontology. Therefore, considerable time and effort are often spent on this step to ensure that the ontology accurately represents all relevant concepts in the domain [2], [71], [73].

*c) Organizing concepts:* Once the main concepts have been identified, the next common step in ontology development is organizing them in a hierarchical structure.

Organizing concepts in a hierarchy helps to show the relationships between different concepts [75]. Here, the upper ontology can guide the structuring of relationships between the domain-specific concepts. It's important to preserve the meaning of higher-level ontology terms during this process. The use of an upper ontology in this step is part of organizing the design and development of ontologies under a common framework. It provides a more coherent and easy navigation as users move from one concept to another in the ontology structure. It also makes the ontology easy to extend as relationships and concept matching are easy to add to existing ontologies [72]. This step is important as it structures the ontology in a way that reflects the inherent structure of the domain. It also facilitates the understanding and use of ontology by providing a clear and intuitive organization of the concepts [75]. Considerable time and effort are often spent on this step to ensure that the ontology accurately represents the relationships among the concepts in the domain.

*d) Defining properties and constraints:* Defining properties and constraints is another vital common step in ontology development. This step involves identifying the properties (or slots) that each concept can have and defining any constraints on these properties [73]. This step is crucial as it adds detail and specificity to the concepts in the ontology. By defining properties and constraints, the ontology can represent not just what concepts exist in the domain, but also what characteristics those concepts have and how they are related to each other. Moreover, defining properties and constraints is essential for the ontology's usability [2], [73]. They allow for more precise queries and more detailed answers, making ontology a more powerful tool for understanding and navigating the domain.

*e) Formalizing the ontology:* After defining properties and constraints, the formalization of the ontology takes place. This common step in ontology development implies taking the concepts, hierarchies, properties, and constraints that have been identified and formalizing them using a formal representation language [2]. This makes the ontology machine-readable and allows it to be used by other software applications. The formal representation language used for this purpose needs to be machine-readable, allowing the ontology to be understood and used by other software applications. This is particularly important in the context of the Semantic Web, where ontologies play a key role in enabling machines to understand and process the vast amounts of data available on the Web [76]. Formalizing the ontology is imperative as it will ensure that the knowledge it represents is explicit, unambiguous, and readily accessible to both humans and machines. This process is key to unlocking the full potential of ontologies as tools for knowledge representation and management [77].

*f) Implementing and testing:* Once the ontology has been formalized, the next step is the implementation and testing. This standard step involves implementing the ontology in a software application and then testing it to make sure it works as expected. This might involve checking that the ontology correctly represents the domain it is intended to cover and that it provides the expected results when used in a software application. The implementation and testing phase is not a one-time process. As the domain of interest evolves and new knowledge is acquired, the ontology may need to be updated and re-tested to ensure that it continues to accurately represent the domain [78]. Although implementing and testing an ontology might sound complex, it is a necessary process that ensures the ontology is correctly integrated into a software application and functions as expected.

*g) Evaluating the ontology:* The subsequent step is evaluation, the critical phase in the ontology development. This is where the identification of the drawbacks took place. In this step, the identified issues will be resolved before the ontology is used, thereby increasing its reliability and usefulness. This common step involves evaluating the quality, consistency, and completeness of the ontology. This might involve assessing that the ontology accurately represents the domain it is intended to cover, and that it doesn't contain any inconsistencies or gaps [73], [79]. Considerable time and effort is needed on this step to ensure the evaluation process is comprehensively conducted to ensure the ontology fit for its purpose.

*h) Documenting the ontology:* In this step, the completed ontology will be documented. The document includes a description of the ontology's purpose and scope, an explanation of the concepts, hierarchies, properties, and constraints it contains, and instructions on how to use the ontology [73]. This document serves as a manual instruction to guide the users and developers in using the ontology correctly. It can eliminate ambiguities or confusion among its users. Therefore, documenting the ontology comprehensively is an imperative step in ontology development as it will facilitate communication and collaboration by providing a common understanding of the ontology [2], [73].

*i) Maintaining and evolving the ontology:* The final step is maintaining and evolving the ontology. This common step involves updating and refining the ontology as needed. This might involve adding new concepts, properties, or constraints, modifying existing ones, or reorganizing the hierarchy of concepts [80]. This step is critically needed as some of the domains like technology are rapidly evolving where new concepts may emerge frequently that need to be added to the ontology. Therefore, consistently maintaining and evolving the ontology will ensure that the ontology is updated and ultimately remains accurate and relevant over time [80], [81].

## V. RESULT

In this section, the common characteristics and key steps, synthesized from the review of various Ontology Development Models (ODMs) and Business Model Ontologies (BMOs) that have been extensively discussed in the previous sections, are integrated. This integration results in a Unified Ontological Approach (UOA) framework. The proposed framework combines the strengths of both lightweight and heavyweight ODMs, drawing inspiration from the success and principles of the BMO.

This UOA aims to facilitate the ontology development process by providing a step-by-step guide. The common characteristics, integrated into the key steps, will serve as best practices that can be adopted or adhered to at each step.

The proposed UOA framework is illustrated in Fig. 5 below and a brief explanation follows afterward.

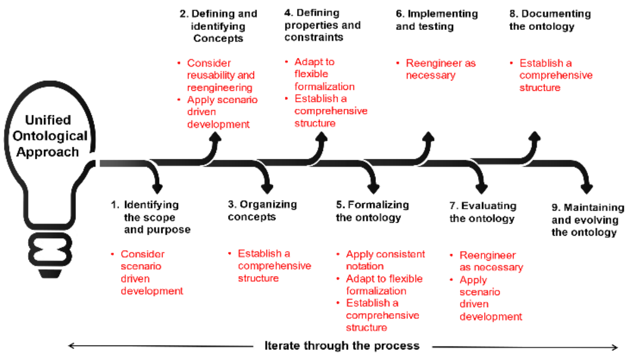


Fig. 5. The proposed unified ontological approach (UOA) framework.

Fig. 5 above showcases the proposed UOA framework. The common steps gathered from the synthesis of various ODMs are organized sequentially to guide the entire ontology development process. The word written in red is the common characteristic which can also be referred to as the best practices to be adopted in each step. To better understand the meaning of each common characteristic in the context of this framework, a simple explanation can be found in Table V below:

TABLE V. BRIEF EXPLANATION OF THE COMMON CHARACTERISTICS

| Common Characteristics (Best Practices in this Framework Context) | Explanation |
|---|---|
| Consider Scenario Driven Development | Using specific, real-world examples to drive the development process |
| Consider Reusability | Using existing ontologies or parts of ontologies in the creation of a new ontology |
| Reengineering as necessary | Modifying the ontology based on new insights or changes in the domain |
| Establish a Comprehensive Structure | Creating a well-organized, detailed, and complete representation of the domain of interest |
| Adapt to Flexible Formalization | The ability to adapt and modify the ontology as needed, while still maintaining its structure and integrity |
| Apply Consistent Notation | Using a standard notation system to ensure that the ontology is understandable and interoperable |
| Iterate throughout the process | Repeatedly go through the steps of a process, making improvements each time based on what was learned in previous iterations |

At the bottom of the framework, there are also fine two-way arrows stating iterate through the process which means that all steps in the framework can be revisited and refined as needed, allowing for continuous improvement and refinement of the ontology.

## VI. Discussion

The proposed UOA for ontology development, as outlined through the integration of various ODMs and insights from the BMO invented by Osterwalder [23], presents a novel framework with both promising strengths and notable considerations. This brief discussion aims to critically examine the implications of this unified approach, shedding light on its strengths, addressing potential limitations, and identifying avenues for future research.

### A. Strengths

*1) Synergy of diverse methodologies:* The UOA leverages the strengths of diverse ODMs, which have been recognized for their ability to guide the process of constructing, deploying, and maintaining ontologies [82]. The iterative process, a key aspect of many ODMs, allows for continuous refinement and adaptation to evolving domain requirements [2], [83]. This iterative approach is also a fundamental aspect of the BMO which was developed specifically to represent business models and provide a comprehensive representation of a business [25]. By adopting this iterative process, according to Pittet et al. [84], the approach fosters a dynamic and responsive ontology development process. This ensures that the ontology remains relevant and up-to-date, adapting to changes in the domain of interest [73]. Therefore, the UOA effectively combines the strengths of both BMO and ODMs to create a robust and flexible ontology development process.

*2) Formal clarity and precision:* The adoption of a consistent notation system as one of the characteristics of the proposed UOA ensures unambiguous representation and reasoning of ontological knowledge. These formal languages provide a standardized way to represent and reason about ontological knowledge, ensuring that the representation is unambiguous [85]. This characteristic can be seen in BMO. According to Chungyalpa et al. [25] BMO uses a common notation to represent different aspects of a business, ensuring unambiguous representation and reasoning of ontological knowledge. This aligns with the argument made by Shukla et al. [85] about the importance of formal languages like OWL, UML, or RDF in ensuring unambiguous representation. According to Norris et al. [86], this consistency in notation enhances clarity in communication across different stakeholders, from domain experts to developers and end users. It ensures that everyone has a shared understanding of the ontological structures, which is key for effective collaboration and successful ontology development [2].

*3) Flexibility in formalization:* The incorporation of a flexible formalization approach as one of the best practices in the proposed UOA is influenced by methodologies such as UPON Lite, LOT and AgiSCOnt which addresses the needs of both domain experts and ontology engineers [34]. Lille et al. [34] added that this approach is oriented towards reduced dependence on ontology engineers, ensuring ease of use for the development of application ontologies. The flexibility in formalization is an important characteristic as it allows for the construction of ontologies from scratch or the reuse of existing ones [87]. Fernandez-Lopez et al. [43] argue that this promotes inclusivity and adaptability across diverse domains. In the context of BMO, flexibility is imperative as it allows the ontology to adapt to the diverse and evolving needs of businesses. Therefore, the flexible formalization approach effectively combines the strengths of both new ontology construction and existing ontology reuse to create a robust and flexible ontology development process.

*4) Efficiency through reusability:* The emphasis on reusability and reengineering in some of the steps in the proposed UOA, drawing from methodologies such as Methontology, contributes to the efficiency and effective maintenance of ontologies over time [37]. Leveraging existing ontologies and knowledge resources mitigates redundancy and streamlines the ontology development process [88]. This is evidenced in the BMO, which is designed to be reusable, allowing it to be applied across various business scenarios and domains [25]. This reusability not only mitigates redundancy but also streamlines the ontology development process, contributing to the efficiency of the ontology.

*5) Scenario-driven customization:* The utilization of scenario-driven development principles in the proposed UOA adopted from the NeOn Methodology facilitates customization and collaboration in ontology engineering based on common situations. This scenario-driven approach enhances the relevance and applicability of ontologies in real-world contexts [70]. This is similar to how BMO can be used to describe and analyze different business scenarios. For instance, BMO can be used to model different aspects of a business such as value proposition, customer segments, channels, customer relationships, revenue streams, key resources, key activities, key partnerships, and cost structure. These aspects can be seen as different scenarios in a business context. Therefore, applying scenario-driven customization allows for greater flexibility and relevance in development.

*6) Comprehensive structural representation:* Aligned with the Uschold and King Methodology, the UOA also emphasizes a comprehensive and systematic structure in the ontology development steps. This ensures the nuanced representation of both structural and dynamic aspects, which is particularly beneficial for addressing complexities in various application domains [78]. The BMO exemplifies the adoption of the structured approach to describe and analyze the business model as noted by Chungyalpa et al. [25]. It allows for the representation of complex business structures and dynamics systematically and comprehensively, similar to how the Uschold and King Methodology is applied in ontology development [47].

Table VI below compares the strength of the proposed UOA against the existing ODMs:

TABLE VI. COMPARISON OF UOA AND EXISTING ODM

| Aspect | Existing ODMs | Proposed UOA |
|---|---|---|
| Synergy of Diverse Methodologies | Each ODM has its own focus and limitations. | Integrates strengths from various ODMs, leveraging diverse methodologies for robust ontology development. |
| Formal Clarity and Precision | Varies in emphasis on formalization. | Ensures unambiguous representation and reasoning of ontological knowledge through consistent notation and formal languages. |
| Flexibility in Formalization | Flexibility ranges across ODMs. | Adopts a flexible formalization approach, allowing for construction from scratch or reuse of existing ontologies, promoting inclusivity and adaptability |
| Efficiency Through Reusability | Reusability is emphasized in some ODMs. | Emphasizes reusability and reengineering for efficiency, leveraging existing ontologies and knowledge resources to streamline development. |
| Scenario-Driven Customization | Scenario-driven approaches vary. | Utilizes scenario-driven principles for customization, enhancing relevance and applicability in ontology engineering based on common situations. |
| Comprehensive Structural Representation | Varies in depth of structural representation. | Emphasizes comprehensive and systematic structure in ontology development, ensuring nuanced representation of both structural and dynamic aspects for complex domains. |

## B. *Limitations*

The proposed UOA approach may also possess several limitations as briefly described below:

*1) Learning curve and expertise:* The adoption of a unified approach, which integrates elements from various methodologies, may introduce a learning curve and require expertise in multiple ODMs. This could potentially pose a challenge for practitioners who may need to familiarize themselves with different methodologies. However, the inclusion of the BMO as a practical example could facilitate comprehension and understanding among the users.

*2) Potential overhead in formalization:* The insistence on a consistent notation system and formalization, while enhancing precision, may introduce an additional overhead in terms of complexity. This may particularly impact users less familiar with formal languages, potentially limiting the accessibility of the approach.

*3) Applicability in highly specialized domains:* While the proposed UOA strives for versatility, its effectiveness in highly specialized domains with unique ontological requirements remains to be thoroughly examined. Certain domains may necessitate tailored methodologies not fully addressed by the integrated elements.

As the study of ODM is dynamic and rapidly growing, future research could focus on the enhancement of the proposed unified approach by adding the relevant steps and best practices towards a more holistic approach. The insights could also be taken to other renowned ODMs, apart from the ODMs discussed in this study.

### C. Future Work

The UOA framework is ready to be applied in real-world scenarios, particularly in the creation of the Information Dashboard Design Ontology (IDDO). This practical implementation will be used as a test environment to evaluate the efficiency and success of the UOA framework in directing the process of developing ontologies.

## VII. Conclusion

This study presents a Unified Ontological Approach (UOA), which is proposed through the integration of common characteristics and steps found in Ontology Development Methods (ODMs). The paper commences with a comprehensive discussion of ontology, its significance, and its applications. It also briefly touches upon the notation of endurant and perdurant elements in ontology, providing a general overview of these elements' existence within ontology. The study further reviews the BMO to glean insights into its development process and to learn from its widespread usage. An in-depth examination of several ODMs is also conducted to gain a succinct understanding of each method's characteristics, steps, and applicability. The paper briefly discusses the characterization of ODMs, specifically lightweight and heavyweight, to shed light on their suitability for various ontology development projects. Leveraging the insights gathered from this comprehensive study process, common characteristics, and key steps are identified. These elements are then synthesized and organized to form the proposed unified ontological framework, drawing from the insights of ODMs and the BMO. This synthesis forms the key contribution of this study.

## Acknowledgment

This research is supported by KDEB Waste Management Grant, Vo.t. No: ZG-2023-001. We highly appreciate the enormous support received for this research project.

## References

[1] T. R. Gruber, "Toward principles for the design of ontologies used for knowledge sharing," International Journal of Human Computer Studies, vol. 43, pp. 907–928, 1995.

[2] N. F. Noy and D. L. Mcguinness, "Ontology Development 101: A Guide to Creating Your First Ontology," 2001. [Online]. Available: www.unspsc.org

[3] P. Gokhale, S. Deokattey, and K. Bhanumurthy, "Ontology Development Methods," DESIDOC Journal of Library & Information Technology, vol. 31, no. 2, pp. 77–83, Mar. 2011, doi: 10.14429/djlit.31.2.860.

[4] A. C. Yu, "Methods in biomedical ontology," Journal of Biomedical Informatics, vol. 39, no. 3. pp. 252–266, Jun. 2006. doi: 10.1016/j.jbi.2005.11.006.

[5] A. I. Walisadeera, A. Ginige, and G. N. Wikramanayake, "User centered ontology for Sri Lankan farmers," Ecol Inform, vol. 26, no. P2, pp. 140–150, Mar. 2015, doi: 10.1016/j.ecoinf.2014.07.008.

[6] A. Basheer, "The art and science of writing narrative reviews," International Journal of Advanced Medical and Health Research, vol. 9, no. 2, p. 124, 2022, doi: 10.4103/ijamr.ijamr_234_22.

[7] C. Baethge, S. Goldbeck-Wood, and S. Mertens, "SANRA—a scale for the quality assessment of narrative review articles," Res Integr Peer Rev, vol. 4, no. 1, Dec. 2019, doi: 10.1186/s41073-019-0064-8.

[8] P. M. Simon, "Ontology," Encyclopedia Britannica.

[9] A. A. Salatino, T. Thanapalasingam, A. Mannocci, A. Birukou, F. Osborne, and E. Motta, "The computer science ontology: A comprehensive automatically-generated taxonomy of research areas," Data Intell, vol. 2, no. 3, pp. 379–416, Jul. 2020, doi: 10.1162/dint_a_00055.

[10] T. Gruber, "Ontology," Encyclopedia of Database Systems, no. 6. Addison Wesley, 2009. [Online]. Available: http://www.w3.org/TR/owl-features/[8http://suo.ieee.org/

[11] N. Guarino, "Formal Ontology in Information Systems," IOS Press, 1998. [Online]. Available: http://www.csc.liv.ac.uk/~pepijn/legont.html

[12] L. Zhou and Q. Wang, "Knowledge discovery and modeling approach for manufacturing enterprises," in 3rd International Symposium on Intelligent Information Technology Application, IITA 2009, 2009, pp. 291–294. doi: 10.1109/IITA.2009.46.

[13] T. Hoppe, H. Eisenmann, A. Viehl, and O. Brigmann, "Shifting from data handling to knowledge engineering in aerospace industry," 2017.

[14] M. Fernández-López and A. Gómez-Pérez, "Overview and analysis of methodologies for building ontologies," Knowledge Engineering Review, vol. 17, no. 2. pp. 129–156, 2002. doi: 10.1017/S0269888902000462.

[15] C.-R. Huang, "Endurant vs Perdurant: Ontological Motivation for Language Variations," 2016. [Online]. Available: https://www.researchgate.net/publication/307597775

[16] R. M. Colomb and M. N. Ahmad, "A perdurant ontology for interoperating information systems based on interlocking institutional worlds," Appl Ontol, vol. 5, no. 1, pp. 47–77, 2010, doi: 10.3233/AO-2010-0074.

[17] S. D. Hales and T. A. Johnson, "Endurantism, perdurantism and special relativity," 2003.

[18] A. Madani Mohamed, R. M. Colomb, H. Ahmed, and R. E. Ahmed, "Perdurant Ontology Using Conceptual Dependency Theory." [Online]. Available: www.tripadisor.co.uk/hotel-Review

[19] I. Johansson, "Qualities, Quantities, and the Endurant-Perdurant Distinction in Top-Level Ontologies.," 2005. [Online]. Available: https://www.researchgate.net/publication/220927024

[20] S. Lorenz, B. Heigl, M. Palmié, and P. Oghazi, "From business models for public actors to public service provision models: Extending the business model concept to the public sector," Technol Forecast Soc Change, vol. 201, Apr. 2024, doi: 10.1016/j.techfore.2024.123273.

[21] [21] D. A. Holdford, V. M. Pontinha, and T. D. Wagner, "Using the Business Model Canvas to Guide Doctor of Pharmacy Students in Building Business Plans," Am J Pharm Educ, vol. 86, no. 3, pp. 243–253, 2022.

[22] [22] T. Otsuka, R. Ishizaki, T. Ahamed, and R. Noguchi, "Three-layer business model canvas of oil-water separation equipment in restaurants and food processing factories," Energy Nexus, vol. 13, Mar. 2024, doi: 10.1016/j.nexus.2024.100273.

[23] A. Osterwalder, "The business model ontology: A proposition in a design science research," Universite de Laussanne, Lausanne, 2004.

[24] A. M. Braccini, "Defining Cooperative Business Models for Inter-Organizational Cooperation," International Journal of Electronic Commerce Studies, vol. 3, no. 2, pp. 229–249, Dec. 2012, doi: 10.7903/ijecs.1093.

[25] W. Chungyalpa, B. Bora, and S. Borah, "Business Model Ontology (BMO): An Examination, Analysis, and Evaluation," Journal of Entrepreneurship & Management, vol. 5, no. 1, 2016, doi: 10.21863/jem/2016.5.1.023.T. J. Bright, E. Yoko Furuya, G. J. Kuperman, J. J. Cimino, and S. Bakken, "Development and evaluation of an ontology for guiding appropriate antibiotic prescribing," J Biomed Inform, vol. 45, no. 1, pp. 120–128, Feb. 2012, doi: 10.1016/j.jbi.2011.10.001.

[26] A. Upward and P. Jones, "An Ontology for Strongly Sustainable Business Models: Defining an Enterprise Framework Compatible With

Natural and Social Science," Organ Environ, vol. 29, no. 1, pp. 97–123, Mar. 2016, doi: 10.1177/1086026615592933.

[27] W. Nie, K. De Vita, and T. Masood, "An ontology for defining and characterizing demonstration environments," J Intell Manuf, 2023, doi: 10.1007/s10845-023-02213-1.

[28] T. J. Bright, E. Yoko Furuya, G. J. Kuperman, J. J. Cimino, and S. Bakken, "Development and evaluation of an ontology for guiding appropriate antibiotic prescribing," J Biomed Inform, vol. 45, no. 1, pp. 120–128, Feb. 2012, doi: 10.1016/j.jbi.2011.10.001.

[29] C. H. Chou, F. M. Zahedi, and H. Zhao, "Ontology for developing web sites for natural disaster management: Methodology and implementation," IEEE Transactions on Systems, Man, and Cybernetics Part A:Systems and Humans, vol. 41, no. 1, pp. 50–62, Jan. 2011, doi: 10.1109/TSMCA.2010.2055151.

[30] G. Kassel, "Integration of the DOLCE top-level ontology into the OntoSpec methodology," 2005. [Online]. Available: https://hal.science/hal-00012203

[31] A. De Nicola and M. Missikoff, "A lightweight methodology for rapid ontology engineering," Commun ACM, vol. 59, no. 3, pp. 79–86, Feb. 2016, doi: 10.1145/2818359.

[32] B. Gibaud et al., "NeuroLOG: sharing neuroimaging data using an ontology-based federated approach," 2011. [Online]. Available: http://neurolog.i3s.unice.fr/doku.php

[33] M. Turki, G. Kassel, I. Saad, and F. Gargouri, "A Core Ontology of Business Processes Based on DOLCE," J Data Semant, vol. 5, no. 3, pp. 165–177, Sep. 2016, doi: 10.1007/s13740-016-0067-2.

[34] N. De Lille and B. Roelens, "A Practical Application of Upon Lite for the Development of a Semi-Informal Application Ontology," 2021.

[35] J. B. Koh, "Metadata Models and Methods for Smart Buildings," University of California San Diego, California, 2020. [Online]. Available: https://escholarship.org/uc/item/36n891k9

[36] H. Sebei, M. A. Hadj Taieb, and M. Ben Aouicha, "SNOWL model: social networks unification-based semantic data integration," Knowl Inf Syst, vol. 62, no. 11, pp. 4297–4336, Nov. 2020, doi: 10.1007/s10115-020-01498-5.

[37] M. Fernandez Lopez, A. Gomez-Perez, and N. Juristo, "METHONTOLOGY: From Ontological Art Towards Ontological Engineering," 1997. [Online]. Available: www.aaai.org

[38] M. Fernandez Lopez, A. Gomez-Perez, J. P. Sierra, and A. P. Sierra, "Building a Chemical Ontology Using Methontology and the Ontology Design Environment," IEEE Intelligent Systems, pp. 37–45, 1999. [Online]. Available: http://www-ksl.stanford.edu:5915

[39] M. Keet, An Introduction to Ontology Engineering, 1.5. Cape Town: LibreTexts, 2020. [Online]. Available: https://LibreTexts.org

[40] O. Corcho, M. Fernandez Lopez, A. Gomez-Perez, and A. Lopez-Cima, "Building Legal Ontologies with METHONTOLOGY and WebODE," R. Benjamins, P. Casanovas, J. Breuker, and A. Gangemi, Eds., Berlin: Springer, 2005, p. 247.

[41] M. Fernández-Lépez, M. C. Suárez-Figueroa, and A. Gómez-Pérez, "Ontology development by reuse," in Ontology Engineering in a Networked World, Springer Berlin Heidelberg, 2012, pp. 147–170. doi: 10.1007/978-3-642-24794-1_7.

[42] M. C. Suárez-Figueroa, A. Gómez-Pérez, and M. Fernández-López, "The NeOn Methodology framework: A scenario-based methodology for ontology development," Appl Ontol, vol. 10, no. 2, pp. 107–145, Sep. 2015, doi: 10.3233/AO-150145.

[43] M. C. Suárez-Figueroa, A. Gómez-Pérez, and M. Fernández-López, "The NeOn Methodology for Ontology Engineering," in Ontology Engineering in a Networked World, Springer Berlin Heidelberg, 2012, pp. 9–34. doi: 10.1007/978-3-642-24794-1_2.

[44] J. Clemente, J. Ramírez, and A. De Antonio, "A proposal for student modeling based on ontologies and diagnosis rules," Expert Syst Appl, vol. 38, no. 7, pp. 8066–8078, Jul. 2011, doi: 10.1016/j.eswa.2010.12.146.

[45] C. Lamsfus, D. Martin, Z. Salvador, A. Usandizaga, and A. Alzua-Sorzabal, "Human-Centric Ontology-Based Context Modelling In Tourism," Paseo Mikeletegi, 2009. [Online]. Available: http://aisel.aisnet.org/mcis2009/64

[46] M. Uschold and M. King, "Towards a Methodology for Building Ontologies," 1995.

[47] M. Uschold and M. Gruninger, "Ontologies: principles, methods and applications," 1996. [Online]. Available: http://www.aaii.oz.au/.

[48] M. Bravo, L. F. H. Reyes, and J. A. Reyes Ortiz, "Methodology for ontology design and construction," Contaduria y Administracion, vol. 64, no. 4, pp. 1–24, 2019, doi: 10.22201/FCA.24488410E.2020.2368.

[49] J. V. Fonou Dombeu and M. Huisman, "Semantic-Driven e-Government: Application of Uschold and King Ontology Building Methodology for Semantic Ontology Models Development," International journal of Web & Semantic Technology, vol. 2, no. 4, pp. 111–20, Oct. 2011, doi: 10.5121/ijwest.2011.2401.

[50] H. N. Abed, A. Y. C. Tang, and Z. C. Cob, "An ontology-based search engine for postgraduate students information at the ministry of higher education portal of Iraq," in International Conference on Intelligent Systems Design and Applications, ISDA, IEEE Computer Society, Oct. 2014, pp. 69–73. doi: 10.1109/ISDA.2013.6920710.

[51] M. Poveda-Villalón, A. Fernández-Izquierdo, M. Fernández-López, and R. García-Castro, "LOT: An industrial oriented ontology engineering framework," Eng Appl Artif Intell, vol. 111, May 2022, doi: 10.1016/j.engappai.2022.104755.

[52] D. Spoladore, E. Pessot, and A. Trombetta, "A novel agile ontology engineering methodology for supporting organizations in collaborative ontology development," Comput Ind, vol. 151, Oct. 2023, doi: 10.1016/j.compind.2023.103979.

[53] B. Youcef, M. N. Ahmad, and M. Mustapha, "Ontophaco: An ontology for virtual reality training in ophthalmology domain - A case study of cataract surgery," IEEE Access, vol. 9, pp. 152347–152378, 2021, doi: 10.1109/ACCESS.2021.3126697.

[54] G. Engelberg, M. Fumagalli, A. Kuboszek, D. Klein, P. Soffer, and G. Guizzardi, "Towards an Ontology-Driven Approach for Process-Aware Risk Propagation," Dec. 2022, doi: 10.1145/3555776.3577795.

[55] G. Guizzardi, A. Botti Benevides, C. M. Fonseca, D. Porello, J. A. Paulo Almeida, and T. Prince Sales, "UFO: Unified Foundational Ontology," IOS Press, 2021. [Online]. Available: http://purl.org/krdb-core/model-repository/.

[56] J. L. G. Dietz and H. B. F. Mulder, Enterprise Ontology A Human-Centric Approach to Understanding the Essence of Organisation. Cham, Switzerland: Springer, 2020. [Online]. Available: http://www.springer.com/series/8371

[57] A. Sattar, M. N. Ahmad, E. S. M. Surin, and A. K. Mahmood, "An Improved Methodology for Collaborative Construction of Reusable, Localized, and Shareable Ontology," IEEE Access, vol. 9, pp. 17463–17484, 2021, doi: 10.1109/ACCESS.2021.3054412.

[58] E. Femi Aminu, I. O. Oyefolahan, M. Bashir Abdullahi, and M. T. Salaudeen, "A Review on Ontology Development Methodologies for Developing Ontological Knowledge Representation Systems for various Domains," International Journal of Information Engineering and Electronic Business, vol. 12, no. 2, pp. 28–39, Apr. 2020, doi: 10.5815/ijieeb.2020.02.05.

[59] R. Studer, V. R. Benjamins, and D. Fensel, "Knowledge Engineering: Principles and methods," Data Knowl Eng, vol. 25, no. 1–2, pp. 161–197, 1998, doi: 10.1016/S0169-023X(97)00056-6.

[60] O. Corcho, "Ontology based document annotation: Trends and open research problems," Int J Metadata Semant Ontol, vol. 1, no. 1, pp. 47–57, 2006, doi: 10.1504/IJMSO.2006.008769.

[61] O. Lassila and D. Mcguinness, "The role of frame-based representation on the Semantic Web," 2001. [Online]. Available: https://www.researchgate.net/publication/229101030

[62] J. ElHassouni and A. El Qadi, "Ontology engineering methodologies: State of the art," in International Conference on Bing Data and Internet of Things, Springer International Publishing, 2021, pp. 59–72.

[63] A. Espinoza, E. Del-Moral, A. Martínez-Martínez, and N. Alí, "A validation & verification driven ontology: An iterative process," Appl Ontol, vol. 16, no. 3, pp. 297–337, 2021, doi: 10.3233/AO-210251.

[64] T. R. Gruber, "A Translation Approach to Portable Ontology Specifications," 1993.

[65] M. Fernández-López, “Ontological Engineering: With Examples from the Areas of Knowledge Management, E-Commerce and the Semantic Web,” 2004. [Online]. Available: https://www.researchgate.net/publication/230771114

[66] K. Verbert, J. Klerkx, M. Meire, J. Najjar, and E. Duval, “Towards a global component architecture for learning objects: An ontology based approach,” in 13th International Worl Wide Web Conference (WWW2004), 2004, pp. 223–231.

[67] C. Schlenoff, “Ontology formalisms: What is appropriate for different applications?,” in 9th Workshop of Performance Metrics for Intelligent Systems - perMIS ’09, 2009, pp. 180–187. doi: 10.1145/1865909.1865947.

[68] B. Villazón-Terrazas and A. Gómez-Pérez, “Reusing and re-engineering non-ontological resources for building ontologies,” in Ontology Engineering in a Networked World, Springer Berlin Heidelberg, 2012, pp. 107–145. doi: 10.1007/978-3-642-24794-1_6.

[69] J. Rogushina, A. Gladun, and R. Valencia-Garcia, “Reuse of ontological knowledge in open science:models,sources,repositories,” in International Conference on Technologies and Innovation, Springer Nature Switzerland, 2023, pp. 157–172.

[70] M. Zipfl, N. Koch, and J. M. Zöllner, “A Comprehensive Review on Ontologies for Scenario-based Testing in the Context of Autonomous Driving,” Apr. 2023, [Online]. Available: http://arxiv.org/abs/2304.10837

[71] S. Parlar, “Ontologies: In Detail- How to Develop an Ontology?,” Analytics Vidhya.

[72] H. Bencharqui, S. Haidrar, and A. Anwar, “Ontology-based Requirements Specification Process,” in E3S Web of Conferences, EDP Sciences, May 2022. doi: 10.1051/e3sconf/202235101045.

[73] R. Rudnicki, “Best Practices of Ontology Development,” 2016. [Online]. Available: http://www.w3.org/2001/sw/wiki/Tools

[74] A. Sattar, A. K. Mahmood, M. N. Ahmad, and E. Salwana, “Issues in Designing Ontology for Waste Management: A Systematic Review,” 2020. Available: https://www.researchgate.net/publication/360167161

[75] E. Sunagawa, K. Kozaki, Y. Kitamura, and R. Mizoguchi, “A Framework for Organizing Role Concepts in Ontology Development Tool: Hozo,” 2005. [Online]. Available: http://wordnet.princeton.edu/

[76] A. F. Donfack Kana and A. H. Abubakar, “Formalizing ontology operations for semantic web under uncertainty using vague graph approach,” 2020. [Online]. Available: https://www.researchgate.net/publication/367201342

[77] A. Ruiz-Iniesta and O. Corcho, “A review of ontologies for describing scholarly and scientific documents,” 2008. [Online]. Available: http://purl.org/spar/c4o

[78] M. Uschold, “Building Ontologies: Towards a Unified Methodology,” 1996.

[79] L. Obrst, W. Ceusters, I. Mani, S. Ray, and B. Smith, “The evaluation of ontologies toward improved semantic interoperability,” in Semantic Web: Revolutionizing Knowledge Discovery in the Life Sciences, vol. 9780387484389, Springer US, 2007, pp. 139–158. doi: 10.1007/978-0-387-48438-9_8.

[80] M. Pietranik and A. Kozierkiewicz, “Methods of managing the evolution of ontologies and their alignments,” Applied Intelligence, vol. 53, no. 17, pp. 20382–20401, Sep. 2023, doi: 10.1007/s10489-023-04545-0.

[81] N. Matentzoglu et al., “Ontology Development Kit: A toolkit for building, maintaining and standardizing biomedical ontologies,” Database, vol. 2022, 2022, doi: 10.1093/database/baac087.

[82] Y. Alfaifi, “Ontology Development Methodology: A systematic review and case study,” in 2nd International Conference on Computing and Information Technology (ICCIT) , Tabuk, Saudi Arabia, 2022, pp. 446–450.

[83] [S. Peroni, “A Simplified Agile Methodology for Ontology Development,” 2017. [Online]. Available: http://www.sparontologies.net/

[84] P. Pittet, C. Cruz, and C. Nicolle, “Guidelines for a dynamic Ontology: Integrating Tools of Evolution and Versioning in Ontology,” 2012. [Online]. Available: http://semanticweb.org/wiki/Main_Page

[85] B. Shukla, S. K. Khatri, and P. K. Kapur, “Deep analysis for the development of RDF, RDFS and OWL ontologies with protege,” in Proceedings of 3rd International Conference on Reliability, Infocom Technologies and Optimization, Noida, India, 2014, pp. 1–6. doi: 10.1109/ICRITO.2014.7014747.

[86] E. Norris, J. Hastings, M. M. Marques, A. N. F. Mutlu, S. Zink, and S. Michie, “Why and how to engage expert stakeholders in ontology development: insights from social and behavioural sciences,” Journal of Biomedical Semantics, vol. 12, no. 1. BioMed Central Ltd, Dec. 01, 2021. doi: 10.1186/s13326-021-00240-6.

[87] N. K. Y. Leung, S. K. Lau, J. P. Fan, and N. Tsang, “Reuse existing ontologies in an ontology development process- an integration-oriented ontology development methodology,” in The 13th International Conference on Information Integration and Web-based Applications and Services, ACM, 2011, pp. 174–181.

[88] K. I. Kotis, G. A. Vouros, and D. Spiliotopoulos, “Ontology engineering methodologies for the evolution of living and reused ontologies: Status, trends, findings and recommendations,” Knowledge Engineering Review, vol. 35, 2020, doi: 10.1017/S0269888920000065.