A Review on Ontology Modularization Techniques - A Multi-Dimensional Perspective

Andrew LeClair, Alicia Marinache, Haya El Ghalayini, Wendy MacCaull, and Ridha Khedri

Abstract—In the past two decades, the use of ontologies has grown accompanied by a diversity in ontological representations and applications to more comprehensive domains. Knowledge engineers have found it expeditious to break down large (monolithic) ontologies to work with smaller fragments. Ontology modularization is the process of extracting a fragment, or ”module”, from an ontology, based on predefined requirements. Due to both the diversity in ontological representations and motivations for modularizing, the body of research on ontology modularization techniques has become extremely large and may be intimidating to the novice ontology researcher. The objective of the paper is to present a comprehensive, albeit high-level, review of ontology modularization techniques. A systematic literature review covering January 1st 2000 to July 31st 2020 was performed to find and classify papers on ontology modularization techniques. The techniques exhibiting certain properties with respect to several features were assessed, and the modularization techniques were classified with a multi-dimensional perspective. The classifications are intended to guide one to a suitable modularization process in accordance with the requirements. The limitations of ontology modularization techniques are highlighted in the conclusion, and characteristics of a desirable framework for an ontology representation that would be best-suited for modularization are presented.

Index Terms—Ontology Design, Knowledge Reuse, Knowledge Engineering, Knowledge Management.

1 Introduction

In the early 2000's, as fields such as knowledge engineering and management saw a rise in popularity, ontologies were widely used as a means to conceptualize the domain knowledge [1]. The Semantic Web [2] became a driving force for this movement, as it demonstrated the potential of ontologies in both their ability to conceptualize a domain, as well as to reason with and learn from it [2]. An ontology defines the concepts, relationships, and other details that are relevant for modeling a domain. In general, a domain contains a set of closely related concepts, thus it can be expected that an ontology will be designed in the same likeness. This approach results in ontologies that are possibly difficult to process or maintain due to the complexity arising from the number or the interconnectedness of the concepts and relations within. Such an ontology is said to be *monolithic* [3].

One way to effectively utilize a monolithic ontology is by breaking it into smaller components (called *modules*) through a process of *modularization* [4], [5]. Modularization allows for more efficient query answering and other reasoning tasks [6], manageable maintenance of the ontology [7], or reconciliation of multiple ontologies into one [8]. It is a means to transform monolithic ontologies into smaller components that are more manageable, so having efficient modularization techniques is crucial to the continued utilization of ontologies. The way the ontology can be broken into modules is limited by many factors, including how the ontology is represented and what the intent of the modularization process is. Different classifications of ontology modularization and examples are provided in Sections 2.4 and 2.5.

The literature related to ontology modularization encompasses a variety of techniques due to the diversity of knowledge representation techniques, and the desired properties of the resulting modules. For the novice ontology user, choosing the appropriate modularization technique is frequently very difficult. In particular, the user needs to know if the technique considered is compatible with their ontology (e.g., attempting to apply a data-focused technique to an ontology composed of solely the concept hierarchy).

In this work, we provide a systematic review of ontology modularization techniques with the objective of providing guidance to the (possibly novice) ontology researcher. The compendium resulting from this investigation can assist the reader in deciding which technique is best suited to their purpose for modularization. The techniques are classified using pairs of dimensions (i.e., characteristics) involving data gathered to answer the research questions posed by the authors. The paper is intended to be a gateway into the complex field of ontology modularization, and is not intended to address some complexities of the techniques used. This work does not aim to survey papers discussing the evaluation or engineering of modular ontologies. Instead the focus is on evaluating techniques that modularize an existing ontology.

The outline of the remainder of the paper is as follows. Section 2 standardizes the vocabulary that will be used throughout this review, and introduces concepts needed to

- *A. LeClair, A. Marinache, and R. Khedri are with the Department of Computing and Software, McMaster University, ON, Canada. E-mail: leclaial@mcmaster.ca, marinaam@mcmaster.ca, khedri@mcmaster.ca*
- *H. Ghalayini is with the Faculty of Applied Science & Technology, Sheridan College, Oakville, ON, Canada. E-mail: haya.elghalayini@sheridancollege.ca*
- *W. MacCaull is with the Department of Computer Science, St. Francis Xavier University, Antagonish, NS, Canada. E-mail: wmaccaul@stfx.ca*

understand certain modularization techniques. The guidelines used to conduct this review are outlined in Section 3. The results with respect to the research questions are presented in Section 4. A discussion that assesses the classes of modularization techniques with respect to module features, classifies the techniques based on pairs of data gathered to answer the the research questions, and explores the current limitations of modularization is presented in Section 5. In Section 6, existing surveys and reviews on the field of ontology modularization are discussed and compared to this work. Section 7 presents our concluding remarks.

2 Preliminaries and Definitions

In this section, we present the definitions for the key terms used in the paper. The terms are first introduced as they are understood by the authors of the investigated papers. To avoid confusion with term usage, we also provide the understanding of the term that will be used in this paper. Any differences between the two are explicitly stated.

2.1 Ontology

Ontologies are used to represent and reason about a domain. We subscribe to the understanding that an ontology is a formal specification of a conceptualization of a domain [9], [10]. There are several formalisms used to represent the domain, such as graphs (e.g., [11], [12], [13]), mathematical structures (e.g., [14], [15], [16], [17]), and logics (e.g., [18], [19], [20]). In addition, there is no consensus to whether an ontology only includes the concepts and relations of the domain (e.g., [21]), or if it also includes the instances of concepts (e.g., [22]).

In this paper, we do not place any restrictions on what constitutes an ontology with regards to the inclusion of concept instances (or data), or to the formalism used. Thus, so long as the authors of the reviewed literature claim to use an ontology, it is evaluated in this work. The specifics of the nature of the ontology (such as the formalism used and the inclusion of data) is recorded for comparison.

2.2 OWL vs DL

One formalism[1] used for representing ontologies is Description Logic (DL) [24]. DL is a collection of first-order logic fragments, each with a distinct expressivity. Web Ontology Language (OWL)[2] [25] is a computational logic-based language developed for the Semantic Web, and currently is widely used as the implementation language for ontologies. Within OWL there exist several sublanguages, such as OWL Full, OWL Lite, OWL DL. However, only OWL Lite and OWL DL can be expressed with a respective DL fragment. Therefore, in this paper we do not consider OWL and DL to be interchangeable. Instead, we keep the distinction made by the author of the investigated paper. If their modularization technique is defined with an ontology implemented in OWL, we do not assume it to also operate on the respective corresponding fragments of DL, and vice versa. We explicitly separate OWL techniques from DL techniques unless the authors state that their technique uses both, then we record it as both. Finally, we only record the version of OWL that the author of the paper utilizes, whether it be OWL1 or OWL2. We do not make assumptions with respect to a technique that operates on an OWL1 ontology to also operate on an OWL2 ontology or vice versa.

1. There are several other formalisms, such as Frame-based logic [23], however Description Logic is distinguished due to its wide usage in the field.

2. Although OWL is a language for writing ontologies rather than a logic or theory for formalizing them, the wide usage of it within the field of ontologies imposes that it be a class of its own for this work.

2.3 Module

An (ontology) module is understood as a component of an ontology, which contains a subset of concepts and relations from the original ontology [4].

This definition has been expanded by several authors to further restrict what constitutes a module. With these restrictions, the authors aim to help ensure the resulting modules are indeed more easily maintained or processed than the original entity as a whole. Examples of added restrictions are self-encapsulation [26], or that the produced module conforms to the (author's) definition of an ontology [20].

Here, we consider a module only to be a component that is extracted from an ontology through some modularization technique. Any properties that are further exhibited by the module produced by the technique (such as self-encapsulation) are recorded for discussion.

2.4 Modularization

Modularization is the process of determining a suitable set of ontological components to form a module. The components that populate the module depend on what is needed to achieve the intended purpose, while fulfilling the definition of a module. For example, if the purpose of the module is solely to answer a query, then it should only be composed of the necessary concepts and relations that can answer the considered query [27]. Modularization can be classified as either ontology partitioning or module extraction depending on what the output is [20].

Ontology partitioning is the process of separating an ontology into a set of modules such that the union of all modules covers the original ontology [20]. Traditionally this set would be composed of disjoint modules (e.g., [11]), however in some papers the term 'partitioning' has been relaxed to not require that the modules be disjoint (e.g., [27], [28]). The word *decomposition* is also commonly used, such as in papers [18], [27]. The process described for decomposition is analogous to partitioning. Alternate to partitioning, module extraction is the process of creating *some* modules (one or many) that cover a desired set of concepts and axioms guided by some input query. We consider both ontology partitioning and module extraction techniques.

An example that depicts both module extraction and ontology partitioning is shown in Figure 1, taken from [29]. In the solid-line oval, there is the concept of Physical Chemistry and all concepts related to it via a taxonomical relationship. Through the module extraction technique, one can extract these concepts and their relationships as a module. On the other hand, we note a similar module can be formed containing the concept Physics and all its

subconcepts, shown in the dashed-line oval. The concepts in the Physics module are disjoint from the ones in the Physical Chemistry module. These two disjoint modules can be determined via an ontology partitioning modularization technique.

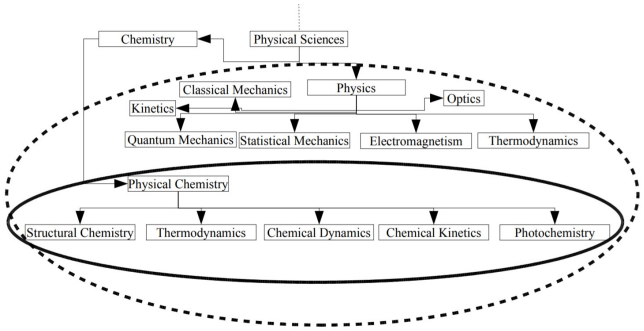


Fig. 1. Ontology Module Extraction vs. Partition (From [29])

2.5 Modularization Approaches

The strategy for modularizing an ontology, which we call the modularization approach, can be classified as a graphical, logical, or hybrid.

A graphical modularization approach (e.g., [11]) produces modules by interpreting the ontology as a graph, and applying algorithms to extract related (or sufficiently related) concepts. For example, in [11] Ghafourian et al. partition an ontology by determining the similarity of concepts, and applying weights to edges to allow for clustering. Graphical modularization approaches are often intuitive and employ statistical methods to determine similarities of concepts or clusters, and require the ontology to have a graphical representation (i.e., a set of vertices and a set of edges). The example shown in Figure 1 illustrates a graphical approach.

A logical modularization approach (e.g., [7]) aims to preserve the knowledge captured within the ontology, based on a given input seed [30]. In other words, given a set of concepts or axioms (the seed), the module should contain the (minimal amount of) concepts, relations, and axioms to ensure that (1) the knowledge about the seed is preserved within the module and (2) no knowledge about the seed can be obtained from the ontology that cannot be obtained from the module only.

An example, taken from [30], of a logical approach modularization is shown in Figure 2. Two ontologies are presented, a medical research projects ontology $\mathcal{P}$, and a medical terms ontology $\mathcal{Q}$. From $\mathcal{Q}$, we wish to extract a module containing the knowledge about the Cystic_Fibrosis and Genetic_Disorder concepts in $\mathcal{P}$. Thus, the input seed is the set of aforementioned concepts. The module must preserve the knowledge about the seed from the medical term ontology. The technique starts with extracting the axioms that that mention the terms in the seed. This results in a module $\mathcal{Q}1 = \{M1, M4, M5\}$. However, a crucial dependency is lost if both $M2$ and $M3$ are removed, while $M5$ brings no useful knowledge to $\mathcal{P}$ regarding the seed. Therefore, a logical approach will extract a module such as $\mathcal{Q}1 = \{M1, M2, M4\}$. Using this module, one can acquire the knowledge about the seed concepts as though they were using $\mathcal{Q}$.

	Ontology of medical research projects $\mathcal{P}$:
P1	Genetic_Disorder_Project $\equiv$ Project $\sqcap$ $\sqcap$ $\exists$has_Focus.Genetic_Disorder
P2	Cystic_Fibrosis_EUProject $\equiv$ EUProject $\sqcap$ $\sqcap$ $\exists$has_Focus.Cystic_Fibrosis
P3	EUProject $\sqsubseteq$ Project
	Ontology of medical terms $\mathcal{Q}$:
M1	Cystic_Fibrosis $\equiv$ Fibrosis $\sqcap$ $\exists$located_In.Pancreas $\sqcap$ $\sqcap$ $\exists$has_Origin.Genetic_Origin
M2	Genetic_Fibrosis $\equiv$ Fibrosis $\sqcap$ $\sqcap$ $\exists$has_Origin.Genetic_Origin
M3	Fibrosis $\sqcap$ $\exists$located_In.Pancreas $\sqsubseteq$ Genetic_Fibrosis
M4	Genetic_Fibrosis $\sqsubseteq$ Genetic_Disorder
M5	DEFBI_Gene $\sqsubseteq$ Immuno_Protein_Gene $\sqcap$ $\sqcap$ $\exists$associated_With.Cystic_Fibrosis

Fig. 2. Module Extraction: Logical Approach (From [30])

A hybrid approach will utilize aspects of both the other approaches. An example of such an approach is in [31], where LeClair et al. utilize the structure of the ontology – articulated around a Boolean lattice – to extract the module, but utilize logical aspects to discuss the knowledge preservation. In this case, the logical aspects are discussed via the Boolean algebra and how the module's Boolean algebra relates to the ontology's Boolean algebra.

2.6 Conservative Extension and Interpolation

As stated, logical modularization techniques rely on the preservation of knowledge within a module. Such a module is said to be conservatively extended by the ontology (or likewise, the ontology is a conservative extension of the module).

Conservative extension is a notion from mathematical logic, often used in fields such as proof theory. In [32], we find the (proof theoretic) definition of conservative extension as follows:

***Definition* 2.1.** Let L and L' be logics. We call L' a (proof theoretic) conservative extension of L provided that all formulas of L are formulas of L' and that, for all formulas A of L, A is a theorem of L iff A is a theorem of L'.

In addition to the definition provided, there exists a stronger notion of conservative extension called *model theoretic* conservative extension. It is presented as follows:

***Definition* 2.2.** Let T and T' be theories. We call T' a (model theoretic) conservative extension of T if every model of T can be extended to a model of T'.

As seen in [33], both definitions of conservative extension are used to define an ontological module. The formulas that are spoken of in the definition of proof theoretic conservative extension are, when considering DL, the Terminological Box (T-Box) sentences. It is common that the proof for showing that the ontology conservatively extends the module is done in two steps, as exemplified by [26].

The first step, referred to as *local correctness* corresponds to determining if "all formulas of L are formulas of L'". The second part, referred to as *local completeness* corresponds to determining if "for all formulas A of L, A is a theorem of L iff A is a theorem of L'".

Similar to the definition of a module using the notion of conservative extension is the definition of a module using the notion of an interpolant. The interpolant is extracted through *interpolation*, and unlike determining a module via a conservative extension, a module is found by forgetting a signature. A foundational work for interpolation can be found in the paper by Konev et al. [34], where a Σ-interpolant is defined as follows:

Definition 2.3. Let $\mathcal{T}$ be an $\mathcal{ELH}^r$-TBox, Σ a finite signature, and $\mathcal{L}$ a set of first order sentences. If $\mathcal{T}_\Sigma$ is a finite set of $\mathcal{L}_\Sigma$-sentences such that $\mathcal{T} \models \varphi$ for all $\varphi \in \mathcal{T}_\Sigma$, then $\mathcal{T}_\Sigma$ is

- a concept Σ-interpolant of $\mathcal{T}$ in $\mathcal{L}$ if $\mathcal{T} \equiv^C_\Sigma \mathcal{T}_\Sigma$;
- an instance Σ-interpolant of $\mathcal{T}$ in $\mathcal{L}$ if $\mathcal{T} \equiv^i_\Sigma \mathcal{T}_\Sigma$;
- a query Σ-interpolant of $\mathcal{T}$ in $\mathcal{L}$ if $\mathcal{T} \equiv^q_\Sigma \mathcal{T}_\Sigma$.

The properties of conservative extension and interpolation utilize the same theory regarding Σ-inseperability.

For the purpose of this work, any paper that utilized either Definition 2.1, 2.2, or 2.3 to describe a module was characterized as a logical approach to modularization.

3 Guidelines for Review Process

This survey was performed following the guidelines outlined by Kitchenham [35]. This systematic process allows for the verification of the completeness of the review, and promotes the repeatability of its results. We outline the process as follows.

In Section 3.1, we introduce the research questions that ultimately form a checklist to evaluate the papers for data. In Section 3.2, we describe the search process. This includes the database query, guided by the research questions, and the initial filtration of papers. Following this, in Section 3.3, we present the inclusion and exclusion criteria that is used to objectively determine what papers are to be included to the survey. Then, in Section 3.4, we introduce the snowballing process used to acquire relevant papers that were not found via the initial search process. Following this, in Section 3.5, we present the checklist used to extract the relevant data from each paper. Finally, in Section 3.6, we communicate how the extracted data is relevant to the initial research questions and the results are tabulated.

For each step that requires an evaluation of a paper or extraction of data, a pairing of authors (of this survey) was formed. Each pair was determined such that an author could not assess the same paper for two different steps of the survey.

3.1 Research Questions

The research questions that are addressed by this review are the following:

- RQ1. What are the existing motivations for the modularization of ontologies?
- RQ2. What are the approaches for modularizing ontologies?
- RQ3. How much activity regarding ontology modularization has there been since 2000?
- RQ4. What are the limitations of the current research?

In addressing RQ1 and RQ2, we considered papers concerning the modularization of an ontology rather than tools or properties of modularization. With respect to RQ3, we limited the scope of the literature review to the papers published from January 2000 to July 2020. We limited the search to 2000 to avoid re-assessing literature that has already been discussed in previous surveys or reviews.

We posed these specific questions to comprehensively evaluate the limitations of the existing modularization techniques (RQ4).

- RQ4.1 Is the proposed technique limited to a specific formalism?
- RQ4.2 Is the technique limited to a specific domain of application?
- RQ4.3 What properties is the technique limited to optimizing, and how are the properties measured?

3.2 Search Process

The search process was a semi-automated search from the Compendex and Inspec databases, accessed via Engineering Village [36], and covered the period January 2000 to July 2020. The year 2000 was considered as a reasonable start date as it was at this time that the modern understanding of ontologies began to form. For instance, OWL, the predominant language for writing ontologies, was introduced in 2004 [25]. Since this time, surveys for specific areas of ontology modularization were made. For instance, [37] is a survey paper from 2009 that surveys modularization techniques for ontologies specifically for the biomedical domain, while [38] is a survey from 2016 that investigates logical modularization techniques. Works that are foundational to modularization techniques occur after 2000, such as [34], where Konev et al. use interpolation. The databases used were chosen for the comprehensive collection of engineering, science, and technical papers. The papers were selected for their inclusion of a modularization technique for ontologies.

After experimenting with several search queries to ensure that a query returned a variety of techniques known to the authors, the query used for the search was:

(((Decomp* OR Modulari* OR Partition* OR (Module NEAR Extract*))) AND (Ontolog* OR ("Description Logic") OR Taxonomy OR Hierarchy) wn TI) NOT ((Proceedings OR Conference OR Forum) wn TI) AND (((Decomp* OR Modulari* OR Partition* OR (Module NEAR Extract*)) AND (Ontolog* OR ("Description Logic") OR Taxonomy OR Hierarchy) AND (Knowledge OR Vocabulary OR Signature OR Semantic OR (Logic NEAR Based)) NOT (Philosophy OR Automata)) wn AB)

The query includes several synonyms for modularization, such as decomposition and partition. It also includes several keywords used in lieu of ontology such as taxonomy, hierarchy, and description logic. It was specified that these keywords be in the title and abstract to ensure that it was a paper that focused on this area of research rather

than be partially related. Keywords were added to greatly reduce the search space, such as restricting philosophy and automata from being within the abstract. This is due to the number of papers which discussed the philosophy of modularization, or automata which utilized modularization. These were eliminated because these papers, although significant in the application or philosophy of modularization, did not discuss the techniques that are the focus of this work, or introduce a new technique.

Due to the uncontrolled keywords assigned to the papers, and the method in which Engineering Village produces the results, it returned many papers that were tangentially related to the input search terms. As a result, an initial filtration was necessary to remove the papers which did not include the key terms in either the title or abstract. To conduct this, a pair of authors (of this survey), screened the set of papers returned from the query, removing any papers that did not meet the above criteria. This step led to the reduction of the number of papers to 320.

3.3 Inclusion and Exclusion Criteria

The papers that made it through the filter process in Section 3.2 were then further evaluated by a second pair of authors. They determined if the paper had the information within the scope of the research questions. Papers that were published between January 1st 2000 and July 31st 2020, and proposed a technique or algorithm for modularization and applied it to a particular ontology were therefore included.

Papers that discussed the process of designing or creating a modular ontology or exclusively investigated properties to assess modules (rather than investigating the modularization techniques themselves) were excluded. Duplicates of papers were also excluded, with the most up-to-date version used.

3.4 Snowballing

The papers that met the inclusion criteria of Section 3.3 formed the primary set of papers to be included to this survey. An additional search referred to as *backward snowballing* [39] followed. Titles in the bibliography of each primary set paper was inspected by a third pair of authors to determine if a paper was of potential interest. Those agreed upon by this pair of authors were then added to a set of candidate papers that would then go through the inclusion and exclusion process as described above. Papers that are a repeat of one in the primary set were discarded. The snowballing process is non-recursive, so once all the papers have had the inclusion and exclusion criteria applied to it, it ends. The purpose of this task is to acquire papers that were not found by the query, but are indeed relevant to this work.

For techniques that were published by a series of papers over several years, the latest version was recorded. This was to avoid having numerous citations for the same technique.

3.5 Data Collection

After completing the steps outlined in Sections 3.2 to 3.4, 69 papers remained. Of these 69 papers, a fourth pair of authors (of this survey) extracted the data relevant for answering the research questions outlined in Section 3.1. The data extracted is itemized as follows:

- The motivation(s) for modularization;
- The definition of a module, and the associated properties of the module;
- The ontology representation formalism and its expressivity (if applicable);
- The modularization approach of the technique;
- The part(s) of the ontology that the modularization technique utilizes;
- The limitations of the modularization process.

3.6 Data Analysis

The data was tabulated to show the following:

- The motivation(s) of modularization (addressing RQ1);
- The approach of the technique for modularization (addressing RQ2);
- The component(s) of the ontology utilized in modularization (addressing RQ2);
- The year the paper was published (addressing RQ3);
- The ontology formalization that the technique was used on (addressing RQ4.1);
- The domain of application for the modularization technique, and whether the technique was limited to that domain (addressing RQ4.2);
- The module properties and what was defined as "good" by the author(s) of the reviewed papers (addressing RQ4.3).

4 Results

In this section, we present the results of the search organized by the research question that the collected data answers. The research questions are addressed sequentially in the following sections: Section 4.1 investigates existing ontology modularization motivations, Section 4.2 investigates the existing approaches, and Section 4.3 investigates the research activity since 2000. The final question of limitations in the field is explored through the subquestions. The limitations regarding ontology formalisms is presented in Section 4.4.1, followed by techniques that are domain-dependent in Section 4.4.2, and the properties of the produced modules in Section 4.4.3. Finally the constraints on the scope of the review are presented in Section 4.5.

4.1 Motivations for Ontology Modularization

Table 1 presents the results regarding the motivations for modularizing an ontology. From the 69 papers that were evaluated, four recurring motivations were found: engineering, reasoning, knowledge hiding, and alignment. In the evaluated papers, the motivation provided may not have used any of the terms above, however it is similar enough that we considered them to be the same. An example of this is a paper that cites the motivation to be ontology matching is included under ontology alignment. The reason for this was to avoid having multiple motivations, each with potentially one cited paper, giving the appearance that this paper cannot be compared to any other. Some papers

TABLE 1
Motivations for Ontology Modularization

Motivation	Paper List
Engineering	[6], [7], [11], [21], [27], [31], [40], [41], [42], [43], [44], [45], [46], [47], [48], [49], [50], [51], [52], [53], [54], [55], [56], [57], [58], [59], [60], [61], [62], [63]
Reasoning	[11], [18], [21], [27], [30], [44], [46], [47], [48], [49], [53], [54], [61], [64], [65], [66], [67], [68], [69], [70], [71], [72], [73], [74]
Knowledge Hiding	[31], [44], [52], [57], [58], [63], [65], [69], [75], [76], [77], [78]
Alignment	[7], [8], [11], [18], [21], [28], [30], [40], [41], [42], [43], [44], [45], [46], [47], [50], [51], [64], [66], [68], [75], [79], [80], [81], [82], [83], [84], [85], [86], [87], [88], [89], [90], [91], [92], [93], [94], [95], [96], [97]

TABLE 2
Modularization Technique Approaches

Approach	Paper List
Logical	[7], [18], [21], [27], [30], [34], [41], [42], [47], [51], [57], [58], [59], [61], [62], [63], [67], [70], [74], [75], [77], [85], [92]
Graphical	[6], [8], [11], [28], [40], [43], [44], [45], [46], [49], [50], [52], [53], [54], [56], [60], [64], [66], [68], [69], [71], [72], [73], [76], [78], [79], [80], [81], [83], [84], [86], [87], [88], [89], [90], [91], [94], [95], [96], [97]
Hybrid	[31], [48], [55], [65], [82], [93]

TABLE 3
Components used for the Modularization Process

Component	Paper List
Data	–
Concepts	[7], [8], [11], [18], [21], [27], [28], [30], [34], [40], [41], [42], [43], [44], [45], [46], [47], [50], [51], [52], [53], [54], [56], [57], [58], [59], [60], [61], [62], [63], [64], [66], [67], [68], [69], [70], [71], [72], [73], [74], [75], [76], [77], [78], [79], [80], [81], [82], [83], [84], [85], [86], [87], [89], [90], [91], [92], [93], [94], [95], [96]
Both	[6], [31], [48], [49], [55], [65], [88]

cited multiple motivations, and thus are listed under each motivation.

The papers citing *engineering* as a motivation include those that aim to produce modules so that the ontology is more refined, easier to maintain, or more usable. The purpose is to modularize an ontology so that it better exhibits software engineering principles, such as high cohesion and low coupling. Through the expression of these engineering principles, the ontology has an improved life-cycle as it is more easily maintained, or may be more efficiently stored and used. For instance, in [7], modularization is used specifically to extract significantly smaller components that are more manageable and easier to use.

Reasoning as a motivation for modularization is characterized by the desire to extract modules from the ontology to improve the effectiveness of existing reasoning techniques. The rationale of these papers typically can be classified as either optimizing the reasoning over the entire ontology by parallelizing the algorithm over each of the modules, or extracting a module to reduce the reasoning space (the number of concepts and relations). The goal is that the smaller modules result in more tractable reasoning tasks. For instance, in [74], modularization is used to extract components that preserves the knowledge of the concepts and relations within the module. The smaller size results in a quicker, more tractable reasoning process, and the preservation of the knowledge ensures that the results of the reasoning process are correct and consistent with respect to the ontology.

Knowledge hiding is motivated by the personalization or access control of the ontology. This includes the similar motivations of knowledge extraction and abstraction. The shared goal of these motivations is to identify a part of the ontology, via query or groupings of similar concepts, and display only those concepts and relations to the user. The reason for showing only one part of the ontology can be for security or privacy, ensuring only certain concepts or relationships are provided to the user, for usability by providing a smaller, light-weight piece of the ontology, or for personalization by allowing the user to select a smaller view of the ontology that they wish to interact with. Unlike the modules extracted for purposes of Engineering, which seek to permanently improve the maintainability of the ontology, a knowledge hiding module is typically 'one-and-done'. The module is produced to answer the needs of the user (e.g., a query) and is then discarded. It is not saved or stored with the ontology. For instance, in [57], modularization is used to 'forget' a part of the ontology. The module will not contain any knowledge about these forgotten concepts, and thus the user is interacting with a restricted view of the ontology.

The motivations of reuse, matching, and merging are all classified under *alignment*. As a whole, this class of modularization motivations deals with identifying a module in an ontology that can be used in another ontology (or compared to concepts in another ontology). An example of such a motivation is modularizing an ontology to determine any equivalences to another separate ontology. Instead of comparing entire ontologies, ultimately, modules of the two ontologies can be compared to one another. For instance, in [11], modularization is used to create smaller resuable components that can be imported into another ontology.

From Table 1, it can be seen that alignment is the most frequent motivation for ontology modularization, whereas knowledge hiding is the least.

4.2 Existing Modularization Approaches

In Tables 2 and 3, the modularization techniques are classified based on which type of approach it is, and which components are utilized to produce the module(s).

In Table 2, the papers that were reviewed classify the technique as either a logical approach, a graphical approach, or an approach that is a combination, or "hybrid", of the other two. The reader is reminded that a logical approach operates with the goal of extracting a module that is conservatively extended by the original ontology or is an interpolant, as defined in Section 2.6. Graphical approaches utilize graph theory to create traversals that determine the concepts and relations that form the module, or statistical methods to identify clusters of similar concepts. These traversals are customizable, allowing the user to specify a relation or a 'distance' from an input concept to determine how the module is formed. The final category

TABLE 4
Publication Year of Paper

Year	Paper List
2000 - 2004	[56]
2005 - 2009	[6], [30], [34], [42], [55], [57], [60], [61], [65], [78], [84], [85], [92], [96], [97]
2010 - 2014	[27], [44], [48], [51], [63], [82], [83], [18], [21], [43], [50], [80], [62], [64], [70], [79], [81], [11], [28], [41], [45], [49], [47], [58], [59], [67], [68], [74], [75]
2015 - 2020	[7], [8], [31], [40], [46], [52], [53], [54], [66], [69], [71], [72], [73], [76], [77], [86], [87], [88], [89], [90], [91], [93], [94], [95]

TABLE 5
Ontology Formalization

Ontology Formalization		Paper List
OWL	OWL-DL	[28]
	OWL-Lite	[51], [63], [85]
	OWL (unspecified)	[11], [21], [30], [42], [50], [54], [64], [66], [69], [72], [82], [88], [89], [91], [93]
	OWL2	[48], [62]
Description Logic	$\mathcal{SHQ}$	[58]
	$\mathcal{SHIQ}$	[18], [67], [70], [92]
	$\mathcal{SHOIQ}$	[61]
	$\mathcal{SHOIN}(\mathcal{D})$	[55]
	$\mathcal{E}-\mathcal{SHIQ}$	[47], [75]
	$\mathcal{SROIQ}$	[7]
	$\mathcal{EL}(\mathcal{I})$	[57]
	$\mathcal{ALC}$	[41], [59], [77]
	$\mathcal{ALCI}$	[74]
	$\mathcal{ALCHI}$	[27], [65]
Graph	Tree	[40], [76], [79], [80], [95]
	Directed	[6], [8], [43], [46], [49], [53], [56], [60], [68], [71], [73], [81], [83], [84], [86], [94], [96], [97]
	Weighted	[44], [45], [78]
Other Mathematical Structure		[31], [52], [87], [90]

employs aspects of both logical and graphical approaches. From Table 2, it is evident that graphical approaches are discussed more frequent, with 40 of the papers identifying as using it. We find that 23 papers claim to use a logical approach, and only 6 papers can be classified as using a hybrid approach.

Table 3 classifies the techniques with respect to which part of the ontology is used for the modularization process into three groups: techniques that only use the concepts used in the ontology, those that use only the data (or instances), or techniques that use a combination of both the concepts and data. Our usage of the terms 'data' and 'concepts' is due their general and wide usage in the literature that is surveyed, and whether it is related to the DL community or to other communities. For instance, an ontology modularization technique that utilizes a DL T-Box is categorized as a technique that uses concepts. Similarly, a technique that utilizes graph vertices from a graph-based ontology, which does not have a T-Box, is also categorized as a technique that uses concepts. Likewise, an ontology modularization technique that utilizes a DL Assertional Box (A-Box) is categorized as a technique that uses data. From Table 3, we see that a large majority of techniques utilize only the concepts of the ontology. A small fraction utilize both the concepts and the data, and no paper employed a technique that utilized only the data.

4.3 Activity within the Ontology Modularization Research Field

Table 4 organizes the modularization techniques based on the year they were published. It shows that the field of modularization is inactive prior to the year when DL and OWL enter the scene, in 2003-2004. It can be seen that the research for ontology modularization became much more active starting in 2010.

As we mentioned in Section 3.4, only the latest publication of a modularization technique was included.

4.4 Limitations of the Ontology Modularization Field

This section contains the results for the research questions 4.1, 4.2, and 4.3.

4.4.1 Representations for Modularized Ontologies

With respect to whether the techniques proposed are limited to a specific structure or formalism (RQ4.1), we refer to Table 5, from which it can be seen that the ontology representations utilized for modularization are evenly represented across OWL, DL, and graph theory. However, taking into account that DL ontologies are often written in an appropriate language of OWL, the imbalance between graph-based methods and DL can be observed.

For the remainder of this paper, when discussing ontologies represented as a graph, it is implied that the graph is connected. None of the papers examined considered disconnected graphs, and as discussed by Wouters et al. in [98], an ontology that is not connected is considered invalid, although each disconnected subgraph can be a valid ontology.

It is also observed that the majority of ontologies written in OWL do not specify the sub-language they use; only one paper specifically utilized OWL2.

4.4.2 Domain of Application for Modularization Techniques

Regarding the limitation of a modularization technique to a specific domain, as stated in (RQ4.2), we observe that the technique in [71] exhibits such a limitation. In this paper, the technique proposed by Pati et al. is guided by Gene Ontology (GO). If the modularized ontology did not contain concepts that also existed in the external ontology, the technique could not be used to the full potential. The technique proposed in [50] by Wennerberg et al. uses Foundational Model of Anatomy (FMA) for existing text corpus. However, there is no dependency on specifically FMA. A text corpus from a different domain could be incorporated and made to operate. Thus, although this technique would require some work to operate in a new domain, it is not considered to be limited by the domain of application such as that in [71]. The other techniques reviewed were demonstrated on a specific domain (using a case study), yet were shown to be applicable to any domain as long as the ontology is formalized in the same way.

TABLE 6
Properties by the Modularization Process

Property	Paper List
Conservative Extension & Interpolation	[7], [21], [30], [41], [42], [57], [58], [59], [61], [62], [63], [70], [74], [75], [77], [85], [92]
Structural Proximity	[6], [28], [31], [40], [43], [52], [54], [56], [66], [71], [72], [78], [79], [80], [82], [84], [86], [91], [96], [97]
Semantic Similarity	[8], [11], [44], [45], [46], [49], [50], [53], [54], [60], [64], [65], [68], [69], [73], [76], [81], [82], [83], [87], [88], [89], [90], [93], [94], [95], [97]
Disjointedness	[18], [27], [47], [48], [51], [55], [67]

4.4.3 Properties of the Produced Modules

The properties described in Table 6 are measured using varying metrics; the definitions for the metrics are provided below and can be found in the papers cited. The property of *conservative extension* is understood as defined in Section 2.5. For the remainder of this paper, when it is stated that a module exhibits conservative extension, it is to say the ontology (the module is extracted from) conservatively extends said module. A *locality*-based module is one which is conservatively extended by the ontology, but is not guaranteed to be minimal, i.e., contain only the essential axioms as defined by Grau et al. in [30]. Although conservative extension is a useful property for defining modules, it is not possible to develop a tractable modularization technique based on it as it is a highly undecidable problem [92]. Instead, the notion of conservative extension is used to motivate the significance of locality-based techniques, which are decidable and usable in practice for less expressive DL fragments. For this reason, when evaluating modules from a logical modularization approach, often times locality is mentioned.

Structural proximity properties measure the quality of a module based on the preservation of the ontological structure. The method of measurement depends on the metric used, yet all aim to produce a numerical means of communicating the relationship between the module and parent ontology. *F-Measure* is a widely-used property to assess the quality of a module by using an average that is calculated using two metrics called *precision* and *recall*. Precision measures the number of relations found in the module that are also found in the ontology, while recall measures the number of relations from the ontology that are also in the module [84]. *Singular value decomposition* is a linear algebraic approach that utilizes the adjacency matrix of the ontology to identify points to modularize on [43], [99]. *Compactness factor* is a repertoire of measurements for the interconnectedness of the concepts within a module and includes lexical similarity, cohesion, and euclidean distance [28].

Semantic similarity properties aim to measure the module compared to the ontology using how well the module maintains the *context* (or understanding) of the ontology. The metrics used for measurement typically try to value modules that maintain semantically related concepts, or relations that are considered rich in meaning. *Neighborhood random walk distance* is the implementation of the random walk process that describes successive steps within the system. It is used to determine how 'close' vertices are, based on how many random steps it takes to traverse from one vertex to another. *Threshold value* is an arbitrary value set by the user to decide what concepts or relations are candidates to form a module. An example is the threshold value representing the minimum weight that an edge must have to be considered as a part of a module [44]. *Degree of centrality* uses the number of connections on a node to determine its importance to the entire ontology, and also to other concepts. *Lexical regularity* is a metric used to determine the similarity of concepts using their name (or label). *Relevant knowledge query* uses the relations to measure the relatedness of concepts, and is also guided by the user's interactions or choices. Lastly, $\mathcal{O}$*-Separability* is used to determine the role assertions that are non-essential in keeping the consistency of the ontology. The ontology partitioning process is guided by the $\mathcal{O}$-Separable role assertions.

The properties of structural proximity and semantic similarity are closely related, as shown in [80]. In their work, a similarity measurement is ultimately used to determine the structural proximity of classes. Ultimately, the categories we use were created based on what the module was evaluated on: the proximity of concepts or number of retained concepts/relations, or the significance of their relatedness or proximity. The reviewed papers were then categorized based on what property the module produced would best exhibit according to the author.

Finally, the property of *disjointedness* measures if the modules produced are independent from each other. That is to say, they do not share concepts or axioms with each other, or a module cannot be produced via the combination of other modules. The *uniqueness of signature* is the measure of how independent and elementary a module is. The aim is to create modules that are not a union or combination of other modules, and contain a minimal amount of axioms.

To address the properties that the modularization techniques aim to maximize (RQ4.3), the papers were classified based on a property or set of properties used to guide the modularization process. Further discussion regarding the properties may be found in Section 5. However, it can be seen that modules built using the notion of semantic similarity are the most predominant.

4.5 Constraints on the Scope of the Study

The results tabulated were based on the language and results of the authors of the papers used. A comparison of how effective the techniques are at the task they set out to do was not made because from the data collected, it is not possible to compare the modules produced from different techniques to determine which technique is objectively better.

We point out the analysis of the techniques that operated on OWL and DL were classified under the formalism explicitly given by the author. Since we do not assume a technique that operates on DL to also operate on a respective OWL sub-language (and vice versa), for some techniques under these classifications, it may be that they also operate under the other (e.g., a DL-based modularization technique may also function for the respective OWL sub-language, although it is not classified as such).

5 Discussion

In this section, we assess and discuss the modularization techniques using the results that were found. In the first subsection we assess classes of modularization techniques, with respect to features of the modules they produce. In the second subsection we classify the papers based on pairs of dimensions (motivations; the role data (if any); the formalism used; the underlying modularization approach and the properties of the produced modules). In the third, we examine the limitations of the research of ontology modularization. In the fourth, characteristics of a desirable framework for an ontology representation are provided.

5.1 Qualitative Assessment of Classes of Modularization Techniques

In this section, we assess the various classes of modularization techniques with respect to some features. A class is composed of the techniques that produce modules that exhibit a specific property. In the remainder of this section, we refer to a class of techniques by the name of the property that the produced modules exhibit. For example, the class for semantic similarity refers to the class of techniques that produces modules exhibiting a property associated with semantic similarity.

In Table 7 the classes of techniques (referred to by the property exhibited by the module(s) they produce) are subjectively assessed based on four different features: applicability, versatility, knowledge preservation, and customizability. Of these four features, knowledge preservation and customizability were chosen due to their prominence in existing frameworks for evaluating modules [100], [101]. The features of applicability and versatility were chosen because they exhibit valuable features of the class of technique as explained below. The four features are qualitatively assessed using a scale of very high (++), high (+), neutral (+/−), low (−), or very low (−−).

Applicability describes the degree to which limitations or restrictions that exist within the class of techniques. Examples of limitations or restrictions include the concerns of decidability of the techniques, the necessity to pre- or post-processing the ontology, or the inability to successfully modularize certain ontologies. *Versatility* describes the limitations of a class of techniques that is applied to specific ontology formalisms. For example, a class of techniques where all the techniques are dependent on the formalism used will be rated lower than a class of techniques that contains techniques that operate on multiple different formalizations. *Knowledge preservation* assesses the degree to which the class of techniques can produce modules that exhibit properties associated with the preservation of knowledge. A class with a higher rating indicates that the majority of the techniques within the class have notions of preserving knowledge, whereas a lower rated class does not. *Customizability* describes the degree of control a user will have with the techniques of a class. Several aspects of control were considered, such as the ability to determine the inclusion (or exclusion) of specific concepts/relations to the module, the ability to determine the size or number of modules, and the ability to prioritize the inclusion of specific concepts or relations over others.

For the applicability feature, the class for conservative extension was rated the lowest (−−) and the class for structural proximity was rated the highest (++). The reason for the low rating of the class for conservative extension techniques is the potential undecidability of the task, and the need to relax the minimality property (thus resulting in locality-based modules). Although the acquisition of locality-based modules is more computationally feasible, the technique may result in modules so large they forsake the original intent of modularization (as seen with GALEN and People [30]). Additionally, it is difficult to determine the complexity of these techniques as they are so highly dependent on the fragment of DL they are used on, e.g., in [102], Del Vescovo et al. only evaluated $\mathcal{SROIQ}$ ontologies. This complexity (or difficulty to easily determine the complexity) is not found within the class for structural proximity, which utilize graph theory or clustering methods to produce the modules, such as in [66], [80]. In addition, this class of techniques is extremely intuitive and easy-to-use, requiring minimal forethought from the user before applying a technique to their ontology. The class for semantic similarity was rated good (+) due to its similarity of the techniques to structural proximity techniques, yet with an additional layer of refinement. For example, Ghafourian et al. proposes a technique where the edges of the ontology can be assigned weights that guide the modularization process [11]. This requires an additional effort by the user, and in some cases (such as the technique proposed by Cirella and Gu, [76]) requires a 'pre-processing' of the ontology before the modularization occurs. The final class for disjointedness was rated neutral (+/−) due to a drastic spectrum of complexities. For instance, in [18], Del Vescovo et al. proposes atomic decomposition which is a modularization approach that utilizes locality-based modules, and thus will suffer the same issues as the class of conservative extension techniques. However, there also exists signature decomposition, presented by Konev et al. in [27], which is shown to be promising for lightweight DL and no harder than subsumption checking for more expressive fragments of DL.

For the versatility feature, the class for conservative extension was rated low (−) as the techniques of this class are designed for and tested solely on DL-based ontologies. Following a similar reasoning, the techniques of the class for structural proximity are designed for and tested on graphs, thus the class for structural proximity was rated the same (−) for this feature. The class for semantic similarity was determined to be highly versatile (++) for this feature because there exist techniques that apply it to graphs, DL-based ontologies, and OWL. The versatility of the class for semantic similarity is also demonstrated by the variety of property metrics, shown in Table 6. The class for disjointedness, although similar to that for conservative extension in that it is largely applied to DL-based ontologies, was rated neutral (+/−) for this feature since the techniques are not as dependent on the fragment of DL used.

For knowledge preservation, since the primary motivation for the class for conservative extension is the extraction of modules that guarantee the properties of local completeness and local completeness, the class is rated very high (++). The class for structural proximity, due to their heuristic

TABLE 7
Assessment of Modularization Techniques with respect to Module Feature

Class of Property \ Feature	Applicability	Versatility	Knowledge Preservation	Customizability
Class of Conservative Extension	−−	−	++	+/−
Class of Structural Proximity	++	−	−−	+
Class of Semantic Similarity	+	++	+/−	++
Class of Disjointedness	+/−	+/−	−	−

nature, cannot make the same guarantees. This inability to guarantee knowledge preservation is expressed in [84] by Doran et al., where they state "[. . .] the process of extracting an ontology module will result in semantic information contained in the parent ontology not being transferred to the ontology modules." Some steps of the modularization process, such as in [80] by Ghazvinian et al., include a pruning process that strives to resemble a "more specific and well-defined module", without the formalization of what a well-defined module is. It scores as very low (−−) for this feature as a result of their heuristic goal of extracting a module that is "good enough" without formalization. The class for semantic similarity ranks as neutral (+/−) for this feature because the techniques have a notion of preserving some sort of information or knowledge within an ontology. In [11], Ghafourian et al. suggests that weights can be assigned to the various types of edges to allow certain relations to be considered more important than others when modularized. Thus, relations deemed of utmost importance can almost entirely be preserved. However, this still does not provide the level of formality found in the class for conservative extension. Finally, the class for disjointedness is rated low (−) as a majority of techniques prioritize the property of disjointedness over preserving knowledge. For example, unlike the techniques in the class for conservative extension, the signature decomposition technique proposed by Konev et al. is less concerned with the axioms that can be generated and preserved, and is instead concerned with the splitting of the signature [27].

The final feature analyzed is customizablility, and the class for conservative extension is rated neutral (+/−) because additional concepts and relations must be included to preserve the property of conservative extension. Although the user is able to customize the process by choosing both the seed signature and the approach for the technique ($\bot$-, $\top$-, and $\bot\top$*-modules), he/she has no control over the number of concepts or relations that will be included. The class for structural proximity is ranked high (+) due to the variety of techniques including customizability at multiple steps of the modularization process. For example, in addition to having the ability to determine which concept(s) the modularization process will be based on, in the technique proposed by Noy and Musen, the users are able to determine the relations or depth of traversal to include, thus, controlling the size of the module [6]. The class for semantic similarity is rated very high (++) because customizations similar to those found within the class for structural proximity are allowed. The ability to prioritize the semantic importance of specific ontological elements also increases the customizability. Examples of prioritizing are assigning of weights to edges in [11] by Ghafourian et al., and manipulating a semantic significance metric that determines the inclusion (or exclusion) of concepts in [46] by Bi et al.. The class for disjointedness is rated low (−) for reasons similar to those from the class for conservative extension, together with the observation that in contrast to locality-based modules, there do not exist multiple approaches of techniques for disjointedness. The user is able to select a seed signature, however, beyond that, the techniques strive for creating modules that are entirely disjoint from each other with no user interaction.

5.2 Classifications of Modularization

This section classifies the modularization techniques with respect to five pairs of dimensions. Section 5.2.1 classifies the motivations with respect to properties of modules produced, as well as with respect to the formalism used. Section 5.2.2 classifies the component used with respect to the approach used. Section 5.2.3 classifies the ontology formalism with respect to the modularization approach. Section 5.2.4 classifies the module property with respect to the ontology formalism used.

5.2.1 Classifications of Modularization Motivations

In this section, we classify the techniques according to the motivations for modularizing in Tables 8 and 9. From these tables, we can see that regardless of the motivation the user may have for modularization, there is a technique that will accomplish the task.

The table highlights and presents the versatility of semantic similarity discussed in the previous section. Thus, if the user is concerned with modularizing their ontology using the property of semantic similarity, they are likely not to be restricted in using a technique to meet their needs. This is not true for the other properties, which mostly lack techniques to address the problem of knowledge hiding. In the case of structural proximity, although there is an additional lack of techniques for reasoning, it appears there is some recent works in exploring this topic, such as in [66], [71], [72]. However, the technique proposed by Pati et al. is limited to creating clusters that are annotated by the GO [71]. Finally, for disjointedness, there is the additional lack of techniques for alignment.

Finally, Table 9 shows the relationship between the motivation for modularization and the ontology formalism. By far the greatest motivation is alignment, which a techniques exists for all listed ontology formalisms. In fact, if considering only the three prevalent ontology formalisms

TABLE 8
Classification of Module Property with respect to Modularization Motivation

Property \ Motivation	Engineering	Reasoning	Knowledge Hiding	Alignment
Conservative Extension	[7], [21], [41], [42], [57], [58], [59], [61], [62], [63]	[21], [30], [70], [74]	[57], [58], [61], [63], [75], [77]	[7], [21], [30], [41], [42], [75], [85], [92]
Structural Proximity	[6], [31], [40], [43], [52], [54], [56]	[66], [71], [72]	[31], [52], [78]	[28], [40], [43], [66], [79], [80], [84], [86], [91], [96], [97]
Semantic Similarity	[11], [44], [45], [46], [49], [50], [53], [54], [60]	[11], [44], [46], [49], [53], [54], [64], [65], [68], [69], [73]	[44], [65], [69], [76]	[8], [11], [44], [45], [46], [49], [50], [64], [68], [81], [82], [83], [87], [88], [89], [90], [93], [94], [95], [97]
Disjointedness	[27], [47], [48], [51], [55]	[18], [27], [47], [48], [67]	–	[51]

TABLE 9
Classification of Ontology Formalism with respect to Modularization Motivation

Formalism \ Motivation	Engineering	Reasoning	Knowledge Hiding	Alignment
OWL	[11], [21], [42], [48], [50], [51], [54], [62], [63]	[11], [21], [30], [48], [54], [64], [66], [69], [72]	[63], [69]	[11], [21], [28], [30], [42], [50], [51], [64], [66], [82], [85], [88], [89], [91], [93]
Description Logic	[7], [27], [41], [47], [55], [57], [58], [59], [61]	[18], [27], [47], [61], [65], [67], [70]	[57], [58], [65], [75], [77]	[7], [41], [75], [92]
Graph	[6], [40], [43], [44], [45], [46], [49], [53], [56], [60]	[44], [46], [49], [53], [68], [71], [73]	[44], [76], [78]	[8], [40], [43], [44], [45], [46], [49], [68], [79], [80], [81], [83], [84], [86], [94], [95], [96], [97]
Other Mathematical Structure	[31], [52]	–	[31], [52]	[87], [90]

– OWL, DL, and a graph – one can pursue any of the tasks of engineering, reasoning, knowledge hiding, or alignment. Knowledge hiding is a less explored motivation for modularization, and has most activity in DL-based ontologies. This aligns with the fact that most interpolation-based techniques, which are motivated by forgetting a signature (i.e., a part of the ontology), are all developed for DL-based ontologies.

5.2.2 Classification of the Role of Data

In this section we assess the literature on whether data is included in the modularization technique, and if so, how it is utilized.

Table 10 captures the relationship between the components of the ontology that are utilized for modularization (data, concepts, or both) and the category of technique (graphical, logical, or hybrid). From this table, it can be seen that data is only used in four techniques [48], [49], [65], [88].

Of the four papers, only Nikitina et al. and Ochieng and Kyanda utilize the data for the modularization purposes [48], [88]. Ochieng and Kyanda proposes that the data guides the entire modularization process by creating alignments between the entities within two ontologies [88]. In the technique proposes by Nikitina et al., the data is the subject of the modularization process, then later the data is reconciled with the T-Box of the ontology to improve performance for consistency checking [48]. The remaining two papers use data either to check for the consistency of the produced module, or for existing inconsistencies before the modularization occurs [49], [65].

The papers that do not utilize the data for the modularization purpose can be classified in one of two categories: the data is not a part of the definition of an ontology, or the data is a part of the ontology yet is not a part of the modularization process. Of the first category for papers that do not consider data a part of the ontology (e.g., [64], [84]), it is immediate that data cannot be used in the modularization process, as there is no data to use. As for the papers from the second category (e.g., [7], [70]), the modularization techniques that are defined work independent from the data. The techniques in both categories are guided entirely by the concept hierarchy.

It is apparent that considering the data while using an ontology may encumber the system. In [103], De Giacomo et al. give examples, such as requiring re-design or intensive maintenance of the ontology when new data is added to the system. Pujara et al. [49] introduce a technique that addresses the common scenario of modules that are 'heavy'

with entities. In their technique, the data is incorporated to the modularization process to specifically avoid the creation of the 'heavy' modules. Similarly, in [65], Wandelt et al. utilize the data to both create partitions that are able to answer queries, and also to update the concept hierarchy to reflect these partitions.

An ideal system seems to be one that can use the data in processes such as modularization, and not be encumbered by the data. De Giacomo et al. [103] proposes a framework of three components: the data, the ontology, and a mapping between the two. Similarly, in [104], [105], Marinache proposes Domain Information System (DIS), a system also constituted of three components: the data layer, the abstract ontology, and an operator that maps the two. With the kinds of systems found in [103], [104], the clear separation of data from the ontology allows for modularization on the concept hierarchy, much like the majority of the techniques in Table 10. The scenario of modules 'heavy' with data is no longer an issue, as the data itself is not considered within a module, only linked to it. Furthermore, the separation of the data layer aims to reduce any data-induced hindrances. For example, any reasoning difficulties associated with an instance that is also a concept (such as with nominals in OWL [106]), or with the update and revision of data does not necessarily require the reconstruction or revision of the entire ontology as the changes may only be local to the data layer.

5.2.3 Classification of Ontology Formalisms

In this section, we investigate how the formalization of the ontology influences which modularization techniques can be used.

Table 11 compares the ontology representation to the approach of modularization (graphical, logical, or both). It is immediate that all logical approaches are designed for ontologies presented using OWL or formalized in DL, while graphical approaches have been designed for various other representations.

We remind the reader that logical approaches are defined using the notion of conservative extension [30] and interpolation [34]. The aim of a logical approach is to produce a module that is both locally correct and locally complete. Such a module can be used with confidence for the intended purpose, as no information pertinent to the module is lost. Thus, an effective usage of a logical approach requires that the information pertinent to a module not necessarily be explicitly stated. This makes DL an ideal candidate for logical approaches, due to its ability to define new, implicit concepts using existing concepts, relations, and operators. In graph theory, the vertices, which are in this context the concepts, are stated explicitly when first defining the graph. In other words, a graph can be represented in DL as an ontology that is composed of only an A-Box. As shown in Table 10, the logical modularization approaches do not utilize an A-Box; data is not used (or considered) in the modularization process. Table 11 shows there is no logical modularization approaches for a graph-based ontology.

Graphical approaches aim to optimize the structural proximity or semantic similarity to produce a set of modules. This metric can be computed using any set of variables relevant to the motivation for modularization, such as the proximity of concepts, the closeness to common ancestors, and the number of neighbors [8]. The computed value is then compared to a user set threshold value to determine concept inclusion to a module. This approach allows for a flexible and interactive modularization process, as the user can set a threshold value to create a module as concise as their needs demand. In addition, the numerous ways to measure semantic similarity allow for multiple ways to modularize the same ontology. Perhaps the greatest advantage of graphical approaches is their intuitive approach, which makes the computation of modules tractable. To rationalize the tractability, we remind the reader the goal of a graphical technique is to reduce the structure, often by extracting a substructure, such that the set of concepts in the substructure is a subset of the ontology concepts. With some techniques (such as view traversal), the reduction also includes the possible reduction of the signature; the structure might not retain all relations of the original. As some graphical approaches rely on statistics and heuristic measures to compute the module, what is extracted might not preserve all the formulas of the module in the ontology (i.e., not be locally correct or complete). Regardless of outcome, graphical modularization techniques extract a subset of concepts from a larger (yet finite) set, with less priority to prove local correctness and completeness (if it exists). Thus, in comparison to logical approaches, which utilize the computationally complex locality-based modules, graphical modularization approaches are perhaps simpler yet tractable to determine.

From an ontology representation perspective, it seems that an ideal representation should be able to utilize both logical and graphical approaches. The work by LeClair et al. in [31], [107] explores this. The formalism used in the paper, referred to as DIS, allows for a combined graphical and logical approach to the modularization. The relationship between the graphical and logical approaches are explored more in [108] to show how a graphical module can be transformed into a logical one. The technique proposed in the paper is based on the technique of Noy and Musen [6], which extracts a view from an ontology. However, the view that is extracted is defined as a Boolean sublattice. Claims regarding the preservation of knowledge are made by utilizing the isomorphism between a Boolean sublattice and an associated Boolean algebra. This type of formalism allows for the description of a module in both structural and logical terms. Beyond this, from Table 11, we observe that ontologies written using OWL offers the ability to utilize both approaches: there exist both logical and graphical approaches for them. The graphical approaches operate by utilizing the graph (through the Resource Description Framework (RDF) triples) to extract subgraphs, while the logical appraoches utilize the underlying DL fragment (if one exists for the sub-language of OWL used). Of the papers evaluated for this work, we found OWL-Lite, OWL DL, OWL Full, and OWL2, where only OWL-Lite and OWL DL have corresponding DL fragments ($\mathcal{EL}$ and $\mathcal{SHOIQ}$, respectively). It is observed from the results that a graphical technique can be applied to any sub-language of OWL. However, as the intent of DL fragments are to be decidable, if it is unknown if there is a corresponding DL fragment for the sub-language of OWL used, it may not be possible to

TABLE 10
Classification of Component with respect to Modularization Approach

Component \ Approach	Logical	Graphical	Hybrid
Data	–	–	–
Concepts	[7], [27], [30], [34], [41], [42], [47], [51], [57], [58], [59], [61], [62], [63], [67], [70], [74], [77], [85], [92]	[8], [11], [28], [40], [43], [44], [45], [46], [50], [52], [53], [54], [56], [60], [64], [66], [68], [69], [71], [72], [73], [76], [78], [80], [81], [83], [84], [86], [87], [89], [90], [91], [94], [95], [96], [97]	[82], [93]
Both	–	[6], [49], [88]	[31], [48], [55], [65]

feasibly apply a logical technique. This inability to ascertain the applicability of a logical technique to the ontology in question is a primary draw-back. If the goal is to have the ability to universally have a graphical or logical technique work in the ontology, there is a need for a more unified ontology representation.

5.2.4 Classification of Classes with Module Properties

In Table 12, the properties are classified based on the formalisms from which they can be produced.

From Table 12, it is immediate that the formalization of the ontology greatly limits the properties of modules that can be produced. For example, an ontology represented as a graph has no techniques that produce a module that is conservatively extended by the ontology. From this table, we make observations with respect to three aspects: the versatility of semantic similarity; the divide between the properties a module exhibits that is modularized from a DL-based ontology versus from a graph; and the ability of OWL-based ontologies to produce a module that exhibits any property.

The first observation is that the semantic similarity property can be exhibited by a module regardless of the ontology formalism used. Its versatility has been discussed in Section 5.1, so instead we focus on investigating the only technique for DL. The technique proposed by Wandelt and Möller is of interest as it seems to provide a new perspective on modularizing DL-based ontologies [65]. Firstly, it utilizes the notion of $\mathcal{O}$-Separability rather than the widely-used conservative extension that is found in the field of modularizing DL-based ontologies. Secondly, the modularization process is guided by assertional axioms, i.e., the data. As stated earlier, the inclusion (and further, the utilization) of data in the modularization process is a contentious topic when considering DL. The research of this work shows opportunity in extending the work of DL with modularization techniques that are not necessarily based on conservative extension, and further, addresses the problems associated with data inclusion.

The second observation is the clear divide between the properties exhibited by a module from a DL-ontology versus those from a graph. This follows from the discussion in Section 5.2.3, which states that logical techniques are designed for ontologies formalized with DL. In contrast, the techniques for ontologies represented as graphs seek to optimize semantic similarity of concepts within modules. Although there seems to be work in closing this divide, as with Wandelt and Möller [65], it remains preliminary and uncommon.

Thirdly, we observe that it appears that for ontologies written using OWL, there exist techniques that can produce modules that exhibit any of the four properties discussed. However, after closely examining the techniques, the same divide between DL and graph representations can be seen. From the techniques that operate on OWL-ontologies to produce modules that exhibit the property of conservative extension, such as in [7], [18], [41], [70], [75], [92], it is observed that the underlying DL fragment is utilized, and the RDF triples (i.e., the graphical component) are not explicitly consulted. This can result in modules that do not contain concepts that are immediately adjacent to the input seed concepts because they are not essential for producing a module that is conservatively extended by the ontology. Similarly, the techniques that produce modules that exhibit semantic similarity or structural proximity properties, such as in [11], [28], [54], [64], [88], [89], [91], utilize the graphical component and do not leverage the DL-fragment (if one exists for the OWL sublanguage). Whereas the DL-centric approaches can potentially miss adjacent concepts because they are not necessary from a logical point of view, the graphical techniques do not consider how the included concepts relate to the expressed DL axioms, and thus, how they affect the implicit knowledge of the module.

It is stated in [4] that while many authors have argued for the benefits of applying principles of modularization to ontologies, there is not yet a common understanding of how modules are defined and what properties they should have. Also, we find in [4], that most of these criteria and metrics for the evaluation of created modules are focused on the techniques of modularization and the analysis of the syntactic structure of the modules. However, we find literature that compares the output of modularization techniques. For instance, in [4] we find measures that compare modules based on local completeness or based the dependency relation of the atomic decomposition. Others consider graphical approaches and use graph-based measures such as connectivity, or degree of vertices to compare graphs. We also find authors that use measures from software engineering such module size, coupling and cohesion, or intra-module distance to compare modules. We remark that several authors (e.g., [109]) pointed out that a module presenting a high score on one measure does not necessarily mean it is

TABLE 11
Classification of Ontology Formalism with respect to Modularization Approach

Formalism \ Approach	Logical	Graphical	Hybrid
OWL	[21], [30], [42], [51], [62], [63], [85]	[11], [28], [50], [54], [64], [66], [69], [72], [89], [91]	[48], [82], [88], [93]
Description Logic	[7], [18], [27], [34], [41], [47], [57], [58], [59], [61], [67], [70], [74], [75], [77], [92]	–	[55], [65]
Graph	–	[6], [8], [40], [43], [44], [45], [46], [49], [53], [56], [60], [68], [71], [73], [76], [78], [79], [80], [81], [83], [84], [86], [94], [95], [96], [97]	–
Other Mathematical Structure	–	[52], [87], [90]	[31]

appropriate for another use case. Hence, a quantitative comparative study is a difficult task that is out of the scope of this review paper. To carry this task, one would apply all the compared modularization techniques to a representative set of ontologies and then examine the quality of the modules obtained.

5.3 Remarks on the Current State of Ontology Modularization Research

In the Section 5.2, we mentioned limitations regarding the current field of modularizing ontologies. In summary, those limitations were: an absence of using (or considering) data for the modularization process, the inability to apply both graphical and logical approaches to the same ontology, and the inability to produce modules that exhibit multiple properties, particularly the properties associated with both graphical and logical techniques. In this section, we further examine these limitations.

The representation that is used for the ontology can be considered as the primary cause for many of these limitations. When examining the inclusion of data, modularization on ontologies represented using some fragment of DL often entirely ignores the A-Box (where the assertions of instances reside; the 'data'). This is because due to its size, the A-Box consumes a large portion of the computation time. As Wandelt et al. [65], [110] discuss, a large source of issues with more expressive fragments of DL comes from the need to split the A-Box in such a way that existing reasoning techniques can still operate. Ultimately, with a DL-based ontology, the ontology engineer must make the decision of whether to ignore the data and focus on the T-Box (i.e., the concept hierarchy), or to include data as a part of the ontology (in the A-Box) and navigate around the issues discussed by the inclusion of data. The same issues regarding the size of the data arise when considering a graph-based representation, where the data and concepts are both represented as vertices.

In addition to the inclusion of data, the representation used was also shown to constrain the modularization techniques that can be used, and thus, the properties of the obtained modules. For example, to apply a graphical approach to a DL-ontology, the RDF triples would need to first be extracted. The constraint in applying graphical approaches to DL-ontologies hinders the ability to produce modules that exhibit properties associated with graphical approaches, such as structural proximity or semantic similarity.

As it was shown in Table 11, OWL provides the greatest versatility in the ability to apply graphical and logical approaches, and thus produces modules that exhibit a wide-array of properties. However, there still exists this divide between techniques that are graphically-based, and those that are logic-based. There does not seem to be a strong effort to produce techniques that utilize both aspects of OWL. Ochieng and Kyanda [88] propose a technique that is perhaps the best example of using the graphical representation of OWL to produce modules that have logical properties (specifically, completeness). Additionally, the large number of sub-languages, and the inability to guarantee the applicability of a technique across any OWL-written ontology adds another layer of challenges to the assessment of modularization techniques.

If the two seemingly disjoint fields of research regarding modularizing (i.e., graph-based ontologies, and DL-based ontologies) are to be united, a representation that captures both is necessary. Efforts for this can be seen with ontologies written in OWL – demonstrated by the numerous hybrid modularization approaches and ability to produce modules which exhibit any of the listed properties – and with ontologies in a Domain Information System (DIS) as explored by LeClair et al. [31]. However, there still exist limitations regarding the OWL sub-languages for which there is no corresponding DL fragment. It also seems there is a clear lack of merging the logic-based techniques with graphical ones within OWL. Additionally, the issues that arise from the inclusion of data are not addressed by OWL. The work by LeClair et al. addresses several of these limitations: the modularization technique is able to be expressed in both structural and logical ways, and the ontology's construction is guided by the data. The ability to express the modularization technique in both a structural and logical way allows for a single holistic view to the system and the modules therein. By characterizing the module in terms of the lattice or algebraic structure, subjectivity is eliminated. The awareness of the data in the modularization process also allows for the ability to communicate how the data is affected.

TABLE 12
Classification of Module Property with respect to Ontology Formalism

Property \ Formalism	OWL	Description Logic	Graph	Other Mathematical Structure
Conservative Extension	[21], [30], [42], [62], [63], [85]	[7], [41], [57], [58], [59], [61], [70], [74], [75], [77], [92]	–	–
Structural Proximity	[28], [54], [66], [72], [91]	–	[6], [40], [43], [56], [71], [78], [79], [80], [84], [86], [96], [97]	[31], [52]
Semantic Similarity	[11], [50], [54], [64], [69], [88], [89], [93]	[65]	[8], [45], [46], [49], [53], [60], [68], [73], [76], [81], [83], [94], [95], [97]	[87], [90]
Disjointedness	[48], [51]	[18], [27], [47], [55], [67]	–	–

5.4 Characteristics of a Desirable Framework for an Ontology Representation

In this section, we articulate the characteristics that an ontology-based system should have in order to address the concerns for modularization discussed in Section 5.3.

The first characteristic is the utilization of the data in the ontology-based system. In [111], it is argued that it is inevitable to use data with the ontology in some parts of the reasoning process, and the authors state that the issue of how the data will be incorporated is seldom discussed within the fields of the Semantic Web or DL. From the results presented in Section 4, we have also observed the absence of employing the data in the modularization process. Data-heavy fields seek to use ontologies to reason on their existing data (e.g., the biomedical sciences [112]). Therefore, the ideal ontology-based system should be able to use the data of the domain in the reasoning process.

In order to do this, the ontology must easily incorporate or link to the data of the domain. In [113], Spanos et al. detail the lack of unification between the relational database and ontological fields, more specifically, the difficulty in creating an ontology from a dataset. Using current Semantic Web research, this process results in OWL Full ontologies, which are undecidable. Wache et al. [5] present frameworks for the interaction between an ontology (or ontologies) and data sets, however they highlight the lack of research in how to construct this link between the dataset and ontology. Thus, the second characteristic that an ontology-based system must provide is a formal mapping between the dataset and ontology that results in a computationally tractable system.

The third desirable characteristic is a rigorously defined scope of the domain conceptualized by the ontology. In [56], it is mentioned that ontologies grow to large sizes due to the interconnectedness of the domains they describe. In [4], it is mentioned that these large monolithic ontologies require the development and use of modularization techniques to increase their usability. An example of such a technique developed for restructuring a monolithic ontology is found in [27], in which interpolation is used to decompose a DL ontology into independent parts. An ideal ontology-based system should not require modularization as a counter-balance to its tendency to grow monolithic. With a well-defined scope of domain, the process of modularization should enhance the ontology rather than be required for the ontology to be usable.

The final desirable characteristic is the ability to formally describe the modules. In [4], it is highlighted that there exists no clear understanding of the ontology modularization field. More specifically, it is unclear what criteria should be used when assessing modules. There exists techniques that are heuristic, and aim to maximize a chosen quality [11], as well as techniques that aim to be more objective and formal [30]. An ideal system should be able to objectively assess and describe the modules produced, as well as easily produce them.

Of the emerging research, it can be seen that these characteristics are beginning to appear in ontology-based systems such as those found in [103], [104], [114]. A primary feature found in these systems is the incorporation of an ontological component, a data-centric component, and a link between the two. This component-based approach results in a system that is able to handle a large amount of data without necessarily growing into a large monolithic system, as traditional methods do. In [103], [104], the data and ontology components are connected through a function, achieving the characteristic of having a well-defined link between data and ontology. Additionally, as the engineering of these systems is guided by the data sets they are linked to, it constrains the scope of the domain to the data that is provided. This in turn ensures that the ontology does not bloat with loosely related (and possibly irrelevant) concepts. The characteristic related to module formalization is best exemplified within [104]. The ontology is a formally defined mathematical construct (containing a Boolean lattice), and thus defines a module to be a sub-ontology (i.e., a Boolean sublattice). The ability to express the properties of the module using formal mathematics ensures an objectivity when comparing modules.

6 Related Work

In this section we introduce other ontology modularization surveys and compare them to this paper.

In [37], techniques are evaluated using the categories of structural or logical. However, the work is limited to ontologies in the biomedical domain. Although both graphical

and DL ontologies are considered, a clear separation between techniques is not provided. Additionally, a distinction between modular ontologies and modularizing ontologies is not made, resulting in motivations that might not be applicable to all techniques the authors propose, such as collaborative design. Lastly, since the time of publishing, several techniques have been developed and a more up-to-date compendium is required. The survey we propose provides a more up-to-date compendium of techniques that goes beyond the biomedical domain with a clear set of definitions and terms.

In the survey paper [101], Oh et al. proposes an evaluation framework for selecting an appropriate ontology modularization tool. The dimensions for assessing modularization tools are: tool performance (e.g., reasoning and language), data performance (e.g., module size, cohesion, and coupling), and tool usability (e.g., user interaction and visualization). While the framework found in [101] proposes a set of criteria that will assist in selecting an ontology modularization tool, our paper expands [101] by going beyond the tools that exist and also classifying techniques that may not yet be implemented with a tool. Additionally, we classify the techniques based on the ontology representations the techniques can be used on, the properties of the module(s) that are produced, and what components of the ontology are utilized for the modularization process, which is not found in [101].

In [109], Khan and Keet introduce the notions of evaluating modularization techniques on three properties: the motivation for modularizing the ontology; the type of technique, or how the technique can be categorized; and the properties of the module(s) produced by the technique. Our study is more recent than this work and includes a more rigorously defined set of motivations, including the predominant motivation of ontology alignment. We also condense the numerous types of modularization techniques into the categories of graphical, logical, or hybrid, while additionally creating a separate metric for what components of the ontology are utilized for modularization. Finally, we provide examples of the metrics utilized by the modularization techniques to ensure the modules they produce exhibit the claimed properties.

In [115], a survey regarding modularization techniques is conducted. The work provides classifications of the modularization techniques that exist, however does not discuss the techniques that populate these classifications. In addition, the paper is from 2006 so does not include many newer works.

Lastly, in [38] we see a thorough collection of methods to modularize DL ontologies. Although it is limited to $\mathcal{ALC}$ and $\mathcal{EL}$ ontologies, it provides techniques that utilize combinations of the T-Box and the A-Box. However, this work does not go beyond DL, whereas our work provides an array of techniques for the various existing ontology representation methods.

7 Conclusion

The body of research on ontology modularization is vast and encompasses a number of highly theoretical aspects as well as numerous techniques and applications. This makes it difficult, especially for the beginning ontology researcher in search of techniques appropriate to the requirements of their particular applications. Our paper is intended to be a gateway into this complex field, and is not intended to address some complexities of the techniques used. We provide a comprehensive, state-of-the-art, albeit high level, review of ontology modularization techniques following the methodology outlined by Kitchenham [35] covering the time period January 1st 2000 - July 31st 2020. We present tables that tabulate the results in response to the research questions posed in the methodology. In the discussion we examine the results and provide a qualitative assessment of the techniques based on the features of applicability, versatility, knowledge preservation, and customizability. We also provide classifications based on five pairs of dimensions: module property with respect to motivation; ontology formalism with respect to motivation; component used with respect to approach; ontology formalism with respect to approach and module property with respect to ontology; formalism. Observations and remarks are made with regards to the current state of the ontology modularization techniques. These classifications can be used as a guideline for selecting a modularization technique that extracts relevant knowledge in accordance with the requirements of the application under consideration. From these classifications, we noted that the limitations of the modularization approach are dependant on the formalism used, and the ability to incorporate data is dependent on the modularization approach used. From these limitations, the characteristics of a desirable framework for an ontology representation, given the context of ontology modularization, are presented.

Acknowledgments

The authors thank the anonymous reviewers for their thorough reviews that greatly improved the quality of the paper.

References

[1] D. Fensel, "Ontologies," in *Ontologies*. Springer, 2001, pp. 11–18.
[2] T. Berners-Lee, J. Hendler, and O. Lassila, "The semantic web," *Scientific american*, vol. 284, no. 5, pp. 34–43, 2001.
[3] S. B. Abbes, A. Scheuermann, T. Meilender, and M. d'Aquin, "Characterizing modular ontologies," in *7th International Conference on Formal Ontologies in Information Systems-FOIS 2012*, 2012, pp. 13–25.
[4] M. d'Aquin, A. Schlicht, H. Stuckenschmidt, and M. Sabou, "Criteria and evaluation for ontology modularization techniques," *Modular ontologies*, pp. 67–89, 2009.
[5] H. Wache, T. Voegele, U. Visser, H. Stuckenschmidt, G. Schuster, H. Neumann, and S. Hübner, "Ontology-based integration of information-a survey of existing approaches," in *IJCAI-01 workshop: ontologies and information sharing*, vol. 2001. Citeseer, 2001, pp. 108–117.
[6] N. Noy and M. Musen, "Traversing ontologies to extract views," *Modular Ontologies*, pp. 245–260, 2009.
[7] A. Armas Romero, M. Kaminski, B. Cuenca Grau, and I. Horrocks, "Module extraction in expressive ontology languages via datalog reasoning," *Journal of Artificial Intelligence Research*, vol. 55, pp. 499–564, 2016.
[8] A. Algergawy, S. Babalou, M. J. Kargar, and S. H. Davarpanah, "Seecont: A new seeding-based clustering approach for ontology matching," in *East European Conference on Advances in Databases and Information Systems*. Springer, 2015, pp. 245–258.
[9] T. Gruber, "What is an ontology," *WWW Site http://www-ksl. stanford. edu/kst/whatis-an-ontology. html (accessed on 07-09-2004)*, 1993.

[10] N. Guarino, D. Oberle, and S. Staab, "What is an ontology?" in *Handbook on ontologies*. Springer, 2009, pp. 1–17.

[11] S. Ghafourian, A. Rezaeian, and M. Naghibzadeh, "Graph-based partitioning of ontology with semantic similarity," in *Computer and Knowledge Engineering (ICCKE), 2013 3th International eConference on*. IEEE, 2013, pp. 80–85.

[12] A. C. Garcia, L. Tiveron, C. Justel, and M. C. Cavalcanti, "Applying graph partitioning techniques to modularize large ontologies." in *ONTOBRAS-MOST*, 2012, pp. 72–83.

[13] P. S. Doran, V. Tamma, T. R. Payne, and I. Palmisano, "An entropy inspired measure for evaluating ontology modularization," in *Proceedings of the fifth international conference on Knowledge capture*. ACM, 2009, pp. 73–80.

[14] X. Xue and Y. Wang, "Optimizing ontology alignments through a memetic algorithm using both matchfmeasure and unanimous improvement ratio," *Artificial Intelligence*, vol. 223, pp. 65–81, 2015.

[15] L. Wang, X. Liu, and J. Cao, "A new algebraic structure for formal concept analysis," *Information Sciences*, vol. 180, no. 24, pp. 4865–4876, 2010.

[16] Y. Wang, "On concept algebra: A denotational mathematical structure for knowledge and software modeling," *International Journal of Cognitive Informatics and Natural Intelligence (IJCINI)*, vol. 2, no. 2, pp. 1–19, 2008.

[17] J. Kohlas and R. F. Stärk, "Information algebras and consequence operators," *Logica Universalis*, vol. 1, no. 1, pp. 139–165, Jan 2007. [Online]. Available: https://doi.org/10.1007/s11787-006-0007-2

[18] C. Del Vescovo, B. Parsia, U. Sattler, and T. Schneider, "The modular structure of an ontology: Atomic decomposition and module count." in *WoMO*, 2011, pp. 25–39.

[19] B. Cuenca Grau, I. Horrocks, Y. Kazakov, and U. Sattler, "Modular reuse of ontologies: Theory and practice," *Journal of Artificial Intelligence Research*, vol. 31, pp. 273–318, 2008.

[20] I. Palmisano, V. Tamma, T. Payne, and P. Doran, "Task oriented evaluation of module extraction techniques," *The Semantic Web-ISWC 2009*, pp. 130–145, 2009.

[21] C. Del Vescovo, D. D. Gessler, P. Klinov, B. Parsia, U. Sattler, T. Schneider, and A. Winget, "Decomposition and modular structure of bioportal ontologies," in *International Semantic Web Conference*. Springer, 2011, pp. 130–145.

[22] B. Konev, C. Lutz, D. Walther, and F. Wolter, "Formal properties of modularisation," *Modular Ontologies*, pp. 25–66, 2009.

[23] M. Kifer, G. Lausen, and J. Wu, "Logical foundations of object-oriented and frame-based languages," *Journal of the ACM (JACM)*, vol. 42, no. 4, pp. 741–843, 1995.

[24] M. Krötzsch, F. Simancik, and I. Horrocks, "A description logic primer," *arXiv preprint arXiv:1201.4089*, 2012.

[25] D. L. McGuinness, F. Van Harmelen *et al.*, "Owl web ontology language overview," *W3C recommendation*, vol. 10, no. 10, p. 2004, 2004.

[26] B. C. Grau, B. Parsia, E. Sirin, and A. Kalyanpur, "Modularity and web ontologies." in *KR*, 2006, pp. 198–209.

[27] B. Konev, C. Lutz, D. K. Ponomaryov, and F. Wolter, "Decomposing description logic ontologies." in *KR*, 2010, pp. 236–246.

[28] M. Kachroudi, S. Zghal, and S. Ben Yahia, "Ontopart: at the crossroads of ontology partitioning and scalable ontology alignment systems," *International Journal of Metadata, Semantics and Ontologies*, vol. 8, no. 3, pp. 215–225, 2013.

[29] P. Doran, "Ontology reuse via ontology modularisation," in *KnowledgeWeb PhD Symposium*, vol. 2006. Citeseer, 2006.

[30] B. C. Grau, I. Horrocks, Y. Kazakov, and U. Sattler, "Just the right amount: extracting modules from ontologies," in *Proceedings of the 16th international conference on World Wide Web*. ACM, 2007, pp. 717–726.

[31] A. LeClair, R. Khedri, and A. Marinache, "Toward measuring knowledge loss due to ontology modularization." in *KEOD*, 2019, pp. 174–184.

[32] R. K. Meyer, "Conservative extension in relevant implication," *Studia Logica*, vol. 31, no. 1, pp. 39–46, 1973.

[33] C. Lutz, D. Walther, and F. Wolter, "Conservative extensions in expressive description logics." in *IJCAI*, vol. 7, 2007, pp. 453–458.

[34] B. Konev, C. Lutz, D. Walther, and F. Wolter, "Logical difference and module extraction with cex and mex." in *Description Logics*, 2008.

[35] B. Kitchenham, "Procedures for performing systematic reviews," *Keele, UK, Keele University*, vol. 33, no. 2004, pp. 1–26, 2004.

[36] Elsevier, "Engineering village," *WWW Site https://www. engineeringvillage. com/home. url (accessed on 01-07-2020)*, 2022.

[37] J. Pathak, T. M. Johnson, and C. G. Chute, "Survey of modular ontology techniques and their applications in the biomedical domain," *Integrated computer-aided engineering*, vol. 16, no. 3, pp. 225–242, 2009.

[38] E. Botoeva, B. Konev, C. Lutz, V. Ryzhikov, F. Wolter, and M. Zakharyaschev, "Inseparability and conservative extensions of description logic ontologies: A survey," in *Reasoning Web International Summer School*. Springer, 2016, pp. 27–89.

[39] C. Wohlin, "Guidelines for snowballing in systematic literature studies and a replication in software engineering," in *Proceedings of the 18th international conference on evaluation and assessment in software engineering*, 2014, pp. 1–10.

[40] M. A. Movaghati and A. A. Barforoush, "Modular-based measuring semantic quality of ontology," in *Computer and Knowledge Engineering (ICCKE), 2016 6th International Conference on*. IEEE, 2016, pp. 13–18.

[41] B. Konev, C. Lutz, D. Walther, and F. Wolter, "Model-theoretic inseparability and modularity of description logic ontologies," *Artificial Intelligence*, vol. 203, pp. 66–103, 2013.

[42] B. C. Grau, I. Horrocks, Y. Kazakov, and U. Sattler, "A logical framework for modularity of ontologies." in *IJCAI*, vol. 2007, 2007, pp. 298–303.

[43] S. Sarkar and A. Dong, "Characterizing modularity, hierarchy and module interfacing in complex design systems," in *ASME 2011 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. American Society of Mechanical Engineers, 2011, pp. 375–384.

[44] K. Etminani, A. R. Delui, and M. Naghibzadeh, "Overlapped ontology partitioning based on semantic similarity measures," in *Telecommunications (IST), 2010 5th International Symposium on*. IEEE, 2010, pp. 1013–1018.

[45] S. Ghafourian, A. Rezaeian, and M. Naghibzadeh, "Modularization of graph-structured ontology with semantic similarity," in *Workshop on Modular Ontologies (WoMO) 2013*, 2013, p. 25.

[46] L. Bi, X.-q. Di, and Y. Zhang, "Ontology modularization method based on the k-pso algorithm," in *Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI), International Congress on*. IEEE, 2016, pp. 2009–2013.

[47] G. Santipantakis and G. A. Vouros, "Modularizing ontologies for the construction of e-shiq distributed knowledge bases," in *Hellenic Conference on Artificial Intelligence*. Springer, 2014, pp. 192–206.

[48] N. Nikitina, B. Glimm, and S. Rudolph, "Wheat and chaff–practically feasible interactive ontology revision," in *International Semantic Web Conference*. Springer, 2011, pp. 487–503.

[49] J. Pujara, H. Miao, L. Getoor, and W. W. Cohen, "Ontology-aware partitioning for knowledge graph identification," in *Proceedings of the 2013 workshop on Automated knowledge base construction*. ACM, 2013, pp. 19–24.

[50] P. Wennerberg, K. Schulz, and P. Buitelaar, "Ontology modularization to improve semantic medical image annotation," *Journal of biomedical informatics*, vol. 44, no. 1, pp. 155–162, 2011.

[51] R. Kontchakov, F. Wolter, and M. Zakharyaschev, "Logic-based ontology comparison and module extraction, with an application to dl-lite," *Artificial Intelligence*, vol. 174, no. 15, pp. 1093–1141, 2010.

[52] S. Babalou, A. Algergawy, and B. König-Ries, "An ontology-based scientific data integration workflow," in *29th GI-Workshop Grundlagen von Datenbanken*, 2017, pp. 30–35.

[53] G. Figueiredo, A. Duchardt, M. M. Hedblom, and G. Guizzardi, "Breaking into pieces: An ontological approach to conceptual model complexity management," in *2018 12th International Conference on Research Challenges in Information Science (RCIS)*. IEEE, 2018.

[54] P. van Damme, M. Quesada-Martínez, R. Cornet, and J. T. Fernández-Breis, "From lexical regularities to axiomatic patterns for the quality assurance of biomedical terminologies and ontologies," *Journal of biomedical informatics*, vol. 84, pp. 59–74, 2018.

[55] B. C. Grau, B. Parsia, E. Sirin, and A. Kalyanpur, "Automatic partitioning of owl ontologies using e-connections." *Description Logics*, vol. 147, 2005.

[56] H. Stuckenschmidt and M. Klein, "Structure-based partitioning of large concept hierarchies," in *International semantic web conference*, vol. 3298. Springer, 2004, pp. 289–303.

[57] B. Konev, D. Walther, and F. Wolter, "Forgetting and uniform interpolation in large-scale description logic terminologies." in *IJCAI*, 2009, pp. 830–835.

[58] P. Koopmann and R. A. Schmidt, "Count and forget: Uniform interpolation of shq-ontologies," in *International Joint Conference on Automated Reasoning*. Springer, 2014, pp. 434–448.

[59] M. Ludwig and B. Konev, "Practical uniform interpolation and forgetting for alc tboxes with applications to logical difference," in *Proc. Int. Workshop Tempor. Represent. Reason.*, 2014, pp. 318–327.

[60] J. Seidenberg and A. Rector, "Web ontology segmentation: analysis, classification and use," in *Proceedings of the 15th international conference on World Wide Web*, 2006, pp. 13–22.

[61] B. Suntisrivaraporn, G. Qi, Q. Ji, and P. Haase, "A modularization-based approach to finding all justifications for owl dl entailments," in *Asian Semantic Web Conference*. Springer, 2008, pp. 1–15.

[62] D. Tsarkov, "Improved algorithms for module extraction and atomic decomposition," in *25th International Workshop on Description Logics*. Citeseer, 2012, p. 345.

[63] Z. Wang, K. Wang, R. Topor, and J. Z. Pan, "Forgetting for knowledge bases in dl-lite," *Annals of Mathematics and Artificial Intelligence*, vol. 58, no. 1-2, pp. 117–151, 2010.

[64] K. Saruladha, G. Aghila, and B. Sathiya, "A partitioning algorithm for large scale ontologies," in *Recent Trends In Information Technology (ICRTIT), 2012 International Conference on*. IEEE, 2012, pp. 526–530.

[65] S. Wandelt and R. Möller, "Islands and query answering for alchi-ontologies," in *International Joint Conference on Knowledge Discovery, Knowledge Engineering, and Knowledge Management*. Springer, 2009, pp. 224–236.

[66] S. S. Ahmed, M. Malki, and S. M. Benslimane, "Ontology partitioning: Clustering based approach," *International Journal of Information Technology and Computer Science*, vol. 7, no. 6, pp. 1–11, 2015.

[67] M. Horridge, J. M. Mortensen, B. Parsia, U. Sattler, and M. A. Musen, "A study on the atomic decomposition of ontologies," in *International Semantic Web Conference*. Springer, 2014, pp. 65–80.

[68] J. Lozano, J. Carbonera, M. Abel, and M. Pimenta, "Ontology view extraction: an approach based on ontological metaproperties," in *Tools with Artificial Intelligence (ICTAI), 2014 IEEE 26th International Conference on*. IEEE, 2014, pp. 122–129.

[69] E. Aroua and A. Mourad, "An ontology-based framework for enhancing personalized content and retrieval information," in *Research Challenges in Information Science (RCIS), 2017 11th International Conference on*. IEEE, 2017, pp. 276–285.

[70] G. M. Santipantakis and G. A. Vouros, "Modularizing owl ontologies using ehq+ddl shiq," in *Tools with Artificial Intelligence (ICTAI), 2012 IEEE 24th International Conference on*, vol. 1. IEEE, 2012, pp. 411–418.

[71] S. K. Pati, S. Mallick, A. Chakraborty, and A. Das, "Informative gene selection using clustering and gene ontology," in *Emerging Technologies in Data Mining and Information Security*. Springer, 2019, pp. 417–427.

[72] A. Tiwari and A. Kumar, "Comparative analysis of optimized algorithms for ontology clustering," in *2018 5th IEEE Uttar Pradesh Section International Conference on Electric al, Electronics and Computer Engineering (UPCON)*. IEEE, 2018, pp. 1–7.

[73] J. Sen, A. R. Mittal, D. Saha, and K. Sankaranarayanan, "Functional partitioning of ontologies for natural language query completion in question answering systems." in *IJCAI*, 2018, pp. 4331–4337.

[74] W. Gatens, B. Konev, and F. Wolter, "Lower and upper approximations for depleting modules of description logic ontologies." in *ECAI*, 2014, pp. 345–350.

[75] C. Del Vescovo and R. Penaloza, "Dealing with ontologies using cods." CEUR, 2014.

[76] D. Cirella and H. Gu, "Generating abstraction networks using semantic similarity measure of ontology concepts," in *2017 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE, 2017, pp. 840–843.

[77] J. Chen, G. Alghamdi, R. A. Schmidt, D. Walther, and Y. Gao, "Ontology extraction for large ontologies via modularity and forgetting," in *Proceedings of the 10th International Conference on Knowledge Capture*, 2019, pp. 45–52.

[78] H. Stuckenschmidt and A. Schlicht, "Structure-based partitioning of large ontologies," in *Modular Ontologies*. Springer, 2009, pp. 187–210.

[79] K. Saruladha, G. Aghila, and B. Sathiya, "Neighbour based structural proximity measures for ontology matching systems," in *Proceedings of the International Conference on Advances in Computing, Communications and Informatics*. ACM, 2012, pp. 1079–1085.

[80] A. Ghazvinian, N. F. Noy, and M. A. Musen, "From mappings to modules: using mappings to identify domain-specific modules in large ontologies," in *Proceedings of the sixth international conference on Knowledge capture*. ACM, 2011, pp. 33–40.

[81] D.-T. Tran, D.-H. Ngo, and P.-T. Do, "An information content based partitioning method for the anatomical ontology matching task," in *Proceedings of the Third Symposium on Information and Communication Technology*. ACM, 2012, pp. 272–281.

[82] Y. Liang, H. Zhu, Q. Tian, and S. Ji, "A method for owl ontology module partition," in *Web Society (SWS), 2010 IEEE 2nd Symposium on*. IEEE, 2010, pp. 372–377.

[83] X. Xu, Y. Wu, J. Chen, and J. Shen, "Sub-ontology mapping based web services discovery framework," in *Advanced Computer Theory and Engineering (ICACTE), 2010 3rd International Conference on*, vol. 3. IEEE, 2010, pp. V3–363.

[84] P. Doran, V. Tamma, and L. Iannone, "Ontology module extraction for ontology reuse: an ontology engineering perspective," in *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*. ACM, 2007, pp. 61–70.

[85] R. Kontchakov, L. Pulina, U. Sattler, T. Schneider, P. Selmer, F. Wolter, and M. Zakharyaschev, "Minimal module extraction from dl-lite ontologies using qbf solvers." in *IJCAI*, vol. 9, 2009, pp. 836–841.

[86] X. Xue and Z. Tang, "An evolutionary algorithm based ontology matching system," *Journal of Information Hiding and Multimedia Signal Processing*, vol. 8, no. 3, pp. 551 – 556, 2017, high heterogeneity;Matcher combination;Matching system;Ontology matching;Optimal model;Recall and precision;Semantic correspondence;State of the art;.

[87] X. Xue and A. Ren, "A large scale multi-objective ontology matching framework," in *International Conference on Intelligent Information Hiding and Multimedia Signal Processing*. Springer, 2017, pp. 250–255.

[88] P. Ochieng and S. Kyanda, "A statistically-based ontology matching tool," *Distributed and Parallel Databases*, vol. 36, no. 1, pp. 195–217, 2018.

[89] T. Mittra and M. M. Ali, "Parallelized and distributed task based ontology matching in clustering environment with semantic verification," *CSI Transactions on ICT*, vol. 5, no. 3, pp. 265–279, 2017.

[90] X. Xue and J.-S. Pan, "A segment-based approach for large-scale ontology matching," *Knowledge and Information Systems*, vol. 52, no. 2, pp. 467–484, 2017.

[91] B. Sathiya, T. Geetha, and K. Saruladha, "Psom 2—partitioning-based scalable ontology matching using mapreduce," *Sādhanā*, vol. 42, no. 12, pp. 2009–2024, 2017.

[92] B. C. Grau, I. Horrocks, Y. Kazakov, and U. Sattler, "Extracting modules from ontologies: A logic-based approach," in *Modular Ontologies*. Springer, 2009, pp. 159–186.

[93] M. Fahad, "Toward analyzing impact of disjoint axioms for merging heterogeneous ontologies," *Journal of Intelligent Information Systems*, vol. 51, no. 1, pp. 49–70, 2018.

[94] Y. Li, Z. Jianhui, J. Liu, and Y. Hou, "Matching large scale ontologies based on filter and verification," *Mathematical Problems in Engineering*, vol. 2020, 2020.

[95] X. Xue, J. Lu, and J. Chen, "Using nsga-iii for optimising biomedical ontology alignment," *CAAI Transactions on Intelligence Technology*, vol. 4, no. 3, pp. 135–141, 2019.

[96] W. Hu, Y. Zhao, and Y. Qu, "Partition-based block matching of large class hierarchies," in *Asian Semantic Web Conference*. Springer, 2006, pp. 72–83.

[97] M. H. Seddiqui and M. Aono, "An efficient and scalable algorithm for segmented alignment of ontologies of arbitrary size," *Journal of web semantics*, vol. 7, no. 4, pp. 344–356, 2009.

[98] C. Wouters, T. Dillon, J. W. Rahayu, E. Chang, and R. Meersman, "A practical approach to the derivation of a materialized ontology view," in *Web Information Systems*. IGI Global, 2004, pp. 191–226.

[99] W. Hu, Y. Qu, and G. Cheng, "Matching large ontologies: A divide-and-conquer approach," *Data & Knowledge Engineering*, vol. 67, no. 1, pp. 140–160, 2008.

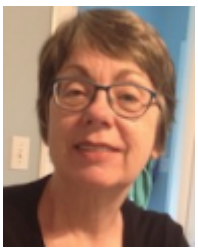
[100] Z. C. Khan, "Evaluation metrics in ontology modules." in *Description Logics*, 2016.

[101] S. Oh and H. Y. Yeom, "A comprehensive framework for the evaluation of ontology modularization," *Expert Systems with Applications*, vol. 39, no. 10, pp. 8547–8556, 2012.
[102] C. Del Vescovo, P. Klinov, B. Parsia, U. Sattler, T. Schneider, and D. Tsarkov, "Empirical study of logic-based modules: Cheap is cheerful," in *International Semantic Web Conference*. Springer, 2013, pp. 84–100.
[103] G. De Giacomo, D. Lembo, M. Lenzerini, A. Poggi, and R. Rosati, "Using ontologies for semantic data integration," in *A Comprehensive Guide Through the Italian Database Research Over the Last 25 Years*. Springer, 2018, pp. 187–202.
[104] A. Marinache, "On the structural link between ontologies and organised data sets," Master's thesis, McMaster University, 2016.
[105] A. Marinache, R. Khedri, A. LeClair, and W. MacCaull, "Dis: A data-centred knowledge representation formalism," in *2021 Reconciling Data Analytics, Automation, Privacy, and Security: A Big Data Challenge (RDAAPS)*. IEEE, 2021, pp. 1–8.
[106] Y. Kazakov, M. Krötzsch, and F. Simancik, "Practical reasoning with nominals in the el family of description logics." in *KR*, 2012, pp. 264–274.
[107] A. LeClair, R. Khedri, and A. Marinache, "Formalizing graphical modularization approaches for ontologies and the knowledge loss," in *Knowledge Discovery, Knowledge Engineering and Knowledge Management*, ser. Communications in Computer and Information Science series, J. Dietz, D. Aveiro, and J. Filipe, Eds. Springer, 2021, vol. 1297, pp. 1–25, invited.
[108] A. LeClair, "A formal approach to ontology modularization and to the assessment of its related knowledge transformation," Ph.D. dissertation, McMaster University, 2022. [Online]. Available: http://hdl.handle.net/11375/27280
[109] Z. C. Khan and C. M. Keet, "An empirically-based framework for ontology modularisation," *Applied Ontology*, vol. 10, no. 3-4, pp. 171–195, 2015.
[110] S. Wandelt and R. Möller, "Towards abox modularization of semi-expressive description logics," *Applied Ontology*, vol. 7, no. 2, pp. 133–167, 2012.
[111] S. Heymans, L. Ma, D. Anicic, Z. Ma, N. Steinmetz, Y. Pan, J. Mei, A. Fokoue, A. Kalyanpur, A. Kershenbaum *et al.*, "Ontology reasoning with large data repositories," in *Ontology Management*. Springer, 2008, pp. 89–128.
[112] I. Merelli, H. Pérez-Sánchez, S. Gesing, and D. D'Agostino, "Managing, analysing, and integrating big data in medical bioinformatics: open problems and future perspectives," *BioMed research international*, vol. 2014, 2014.
[113] D.-E. Spanos, P. Stavrou, and N. Mitrou, "Bringing relational databases into the semantic web: A survey," *Semantic Web*, vol. 3, no. 2, pp. 169–209, 2012.
[114] J. Jaskolka, W. MacCaull, and R. Khedri, "Towards an ontology design architecture," in *Proceedings of the 2015 International Conference on Computational Science and Computational Intelligence*, ser. CSCI 2015, 2015, pp. 132–135.
[115] M. d'Aquin, M. Sabou, and E. Motta, "Modularization: a key for the dynamic selection of relevant knowledge components," in *Proceedings of the 1st International Conference on Modular Ontologies-Volume 232*. CEUR-WS. org, 2006, pp. 15–28.

Andrew LeClair received his M.A.Sc. and Ph.D. in Software Engineering from McMaster University, Hamilton, ON, Canada. He is a AI and Knowledge Graph Researcher at Bosch, and his research interests include ontology development and modularization, software engineering, and the formalization of knowledge systems.

Alicia Marinache is a Ph.D. Candidate in Software Engineering at McMaster University, Hamilton, ON, Canada. A former Software Engineer, her research interests include formalization of information systems and their applications, software engineering, and teaching and learning.

Haya El Ghalayini is a computer science professor in Faculty of Applied Science and technology at Sheridan College, Oakville, Ontario, Canada. She received her PhD in computer science from University of the West of England, Bristol-United Kingdom. Her research interests are in the areas of conceptual modeling and ontologies, e-learning, and utilizing human computer interaction in the usability of the designed blended courses. Dr. El Ghalayini has taught both basic and advanced courses in computer science and software engineering.



Wendy MacCaull received her M.Sc. and Ph.D. degrees in Pure Mathematics from McGill University, Montreal, QC, Canada, and then joined the Department of Mathematics, Statistics and Computer Science at St Francis Xavier University, Antigonish, NS, Canada. She is currently a Senior Research Professor in the Department of Computer Science at StFX. She has served on program committees as member or co-organizer for numerous conferences and workshops, and as referee for a large number of national and international grants. Her research interests include model-based software engineering, ontologies, nonclassical logics, automated theorem proving, and clinical workflow processes. The past dozen years her research has focused on methods to build correct software for safety critical processes in the healthcare domain.

Ridha Khedri received M.Sc. and Ph.D. degrees in computer science from Laval University, Quebec, QC, Canada, in 1993 and 1998, respectively. In March 1998, he joined the Communications Research Laboratories, McMaster University, Hamilton, ON, Canada, as a Postdoctoral Researcher under the supervision of Prof. D.L. Parnas. Currently, he is a Professor of software engineering at McMaster University. From July 2016 to June 2019 he served as the Chair of the Department of Computing and Software. He is a Licensed Professional Engineer in the province of Ontario, and member of the Association for Computing Machinery and the IEEE Computer Society. He has been the co-organizer, program committee member, and referee of more than 30 international workshops and conferences. His research interests include engineering of medical device software, information security, and ontology-based reasoning. His research record includes more than 90 peer-reviewed articles.