Computer Science Review 35 (2020) 100223

Contents lists available at ScienceDirect

Computer Science Review

journal homepage: www.elsevier.com/locate/cosrev

Survey

Combining UML and ontology: An exploratory survey

Meriem Mejhed Mkhinini *, Ouassila Labbani-Narsis, Christophe Nicolle

CIAD, EA 7533, univ. Bourgogne Franche-Comté, UB, 21000 Dijon, France

ARTICLE INFO

Article history:
Received 29 January 2019
Received in revised form 13 October 2019
Accepted 20 December 2019
Available online 8 January 2020

Keywords:
Unified modeling language (UML)
Ontology
Systematic literature review (SLR)
Software engineering
Ontology engineering

ABSTRACT

UML models and ontologies are two knowledge representations with different strengths and weaknesses. Until recently, they were considered unrelated research domains. However, studies investigating their underlying paradigms and the approaches combining these two are increasingly emerging. Nevertheless, the state of the art research covering the relationship between the two is still under exploration. In this paper, we aim to provide a comprehensive overview of both domains by conducting a literature review of the relevant research work. In this survey, the relationship between UML and ontology is investigated from both the theoretical and practical perspectives. We present a detailed classification of the existing work based on the considered issues and their practical use cases. Finally, we provide an evaluation of the existing work according to the criteria we identified.

Contents

1. Introduction

Among the attempts to improve knowledge exchange across multi-disciplinary fields, the reconciliation of object-oriented modeling and semantic modeling approaches raised interest in recent years. On one hand, the unified modeling language (UML)

* Corresponding author.
E-mail address: meriem.mejhed-mkhinini@u-bourgogne.fr
(M. Mejhed Mkhinini).

https://doi.org/10.1016/j.cosrev.2019.100223
1574-0137/

is the de-facto standard formalism for software design and analysis boosted by the model-driven engineering community [1]. For a long time, the UML language served as support to engineers around the world to model the applications domain and communicate efficiently. On the other hand, the Artificial Intelligence (AI) community endorses ontologies as a means of knowledge representation and reasoning [2]. The more recent rise of the Semantic Web furthered the development of ontology web languages to provide a solution for formally describing the domain concepts and their semantics.

Although UML and ontologies originate from separate fields, they are both used for conceptual modeling. Indeed, the similarities and differences between them motivated researchers to investigate their combination in integrated approaches. The earliest research work combining them, according to our findings, was proposed in 1999 [3]. Yet, only few studies have tackled the issue of reviewing the state of the art research. Based on our several studies, we find that beginners in the domain, find it challenging to build an overview of the leading scientific issues in the literature. To the best of our knowledge, no study has yielded an exploratory survey about the research work combining UML and ontologies.

In this paper, we conduct a Systematic Literature Review (SLR) of the research work combing UML and ontologies. This exploratory survey aims to identify the critical research questions, analyze the literature, and outline the key challenges and open issues. Using the results of the SLR, we propose a classification of the existing works based on the motivation of each approach. We also summarize the characteristics of the proposed UML to ontology transformations in the reviewed articles and present the complexity of the existing transformation rules.

The remainder of this paper is organized as follows: Section 2 gives a brief presentation of UML, ontologies, their similarities, and their underlying assumptions. Section 3 describes our research process based on the systematic literature review guidelines and the first results, which focused on identifying the state-of-the-art approaches combining UML and ontologies. Section 4 presents our classification of the existing work and the scientific issues we extracted from the studies. Section 5 details our synthesis of the transformations between UML and ontologies reported in the literature. Section 6 discusses the open issues, concludes this paper, and identifies future research perspectives.

2. Preliminaries and definitions

In this section we briefly recall some definitions UML modeling language, ontologies, and the similarities motivating the literature work.

2.1. Unified modeling language

UML emerged from the unification of three object-oriented modeling methods: the Booch Method [4], the Object Modeling Technique (OMT) [5], and the Objectory Method [6]. The Object Management Group (OMG[1]) first established UML as a standard in 1997 and continues to manage it today. UML offers many kinds of diagrams to represent a system from different viewpoints. If we consider the structural diagrams such as the class diagram, the object diagram, or the profile diagram, we can see that they showcase the static concepts of a system and how they relate to each other. However, behavior diagrams describe the dynamic behavior of objects in a system, and the series of changes to the system over time.

[1] https://www.omg.org/.

Table 1
Comparison of UML and ontologies underlying assumptions.

	UML	Ontology
Object-centered	✓	✓
Global scope properties	✓	✓
First knowledge layer	Constant	Evolving
Interpretation	Close world assumption (CWA)	Open world assumption (OWA)
Naming assumption	Unique	Synonym
Abstract syntax	Semi-formal	Formal
Concrete syntax	Conforms to metamodel	Description logic based

Many of the UML structures are based on graphical notations defined by OMG MOF[2] metamodel. This architecture allows extensions to the standard UML metamodel using stereotypes, tagged values, and constraints by grouping them into a UML Profile, which allows adapting UML metamodel to different platforms and domains.

2.2. Ontology

First appeared in metaphysics as the study of the structure of reality, the term ontology was adopted in Artificial Intelligence (AI) research in 1980 to define a conceptualization of a knowledge domain, based on concepts, relationships, and restrictions between them [7]. An ontological model formally represented in a computer-readable format allows known facts or assumptions to be used to derive a conclusion or to make inferences (i.e., reasoning). This formal representation of knowledge relies on logic-based languages, namely description logics (DL) [8].

Later in 2001, Berners-Lee envisioned a Semantic Web which provides a common framework that allows data to be shared and reused across application, companies, and community boundaries. The rise of the new envisaged Web furthered the development of DL based languages such as OWL (Ontology Web Language). Indeed, OWL is a new markup language for publishing and sharing data using Web ontologies [9]. Currently, technologies for the Semantic Web are developed and managed by the World Wide Web Consortium (W3C) [10].

2.3. Similarities and differences

The work of [11,12] showed that UML and ontologies share similarities, which make it possible to investigate integrated approaches combining them. They also illustrated the differences that raise some challenges. We summarize the similarities and differences in Table 1.

A fundamental resemblance between UML and ontology is using object-centered abstractions to represent a domain of interest. This property is translated via the basic language constructs used in OWL and UML (i.e., classes, properties, and relations). Indeed, both UML and ontologies describe knowledge in a two-layered structure. The first layer defines the abstract structure of the domain of application. The second layer is the instantiation of the first one.

However, ontological knowledge is interpreted under the open-world assumption, which entailed that the model is a representation of partial knowledge about a domain. If a statement cannot be inferred as true or false about an object, it is assumed to be unknown. In contrast, UML models are interpreted under the closed world assumption. This means that if a statement is not announced to be true, it is assumed to be false. So, a UML model

[2] https://www.omg.org/mof/.

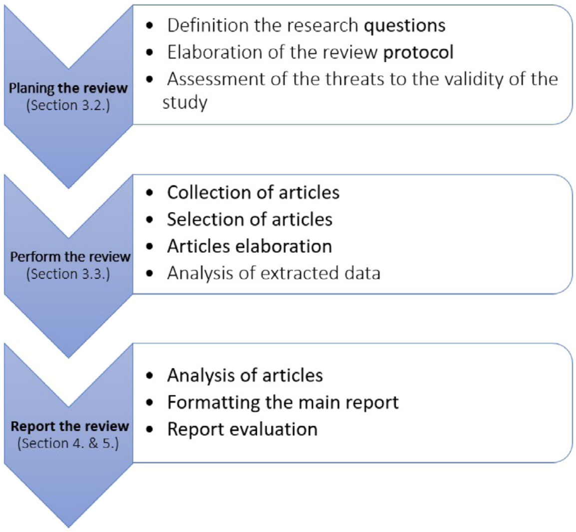

Fig. 1. Systematic Literature Review Process.

is considered a complete representation of all the knowledge in the domain.

The naming assumption in UML and ontologies also differs. Logic-based ontology languages allow the definition of equivalent classes. This implies that two classes with different names can be describing the same individuals (objects). In UML, two different class names imperatively describe distinct classes. Moreover, the scope of relations is also different in UML and ontologies. Properties that describe relations in an ontology have a global scope. They can be defined independently to describe a relationship between objects. In UML, a relation exists only between two or more classes. Nevertheless, in both cases, relations have the semantics of their own, participate in the classification process, and can be built into taxonomies.

Finally, in ontologies axioms are used to restrict the definitions of classes and/or properties whereas, in UML, Object Constraint Language (OCL) [13] is used to define constraints that guarantee a valid state of objects at all times. In this survey, we focus primarily on UML.

3. Systematic literature review (SLR)

The SLR is a type of literature review conducted according to a sequence of activities previously defined in a research protocol [14]. Commonly used in the fields of medicine and biology, the SLR protocol aims to identify, evaluate, and interpret all available research relevant to a particular topic area. Such a comprehensive approach is expensive in terms of time and participants, but it helps to avoid bias when selecting the relevant studies to the research questions. The selected papers for analysis are the primary studies, and the systematic review is called the secondary study.

This section explains how we apply the systematic literature review guidelines to conduct our exploratory survey. First, Section 3.1 highlights the research questions, the review protocol, and how threats to the validity of the survey are overcome. Then, Section 3.2 demonstrates the execution of the defined protocol and the first results. Finally, Section 3.3 reviews the previous surveys.

The considered process is depicted in Fig. 1. The first step is planning the literature review by defining its stages and identifying potential constraints that might encounter us. We start by formulating the research questions which we aim to answer. Then, we elaborate the adopted protocol to conduct the review as well as the actions relating to its evaluation. Second, we perform the activities planned in the protocol, including the collection and selection of articles, the elaboration of the final selection of articles (primary studies), and the analysis of the extracted information. Finally, we document and explain the review, summarizing the results of each research question.

3.1. Planning the review

The survey aims to review the current work combining the modeling language UML and ontologies. In the early stages of this review, we envisioned two scenarios joining the use of UML and ontologies together. First, taking advantage of the specificities of UML or ontologies to complement one or the other. Second the use of UML or ontologies to address a limitation in one or the other. Based on these two scenarios, we first fix the research questions. Then, we define the protocol of the review. Finally, we detail the actions of evaluations carried out.

3.1.1. Definition of the research questions

In this literature review, we aim to answer the following research questions:

RQ1: To understand the goal of the existing work, we ask the following question: «What is the motivation behind combining UML models and ontologies in each work?»

RQ2: We want to highlight the issues motivating each research work: «What are the scientific issues addressed in existing approaches?»

RQ3: The UML language allows representing a domain conceptually using different diagrams. To better identify the relation between ontologies and the different UML diagrams or the UML language in general, we ask the following question: «What elements of the UML standard are used in each work?»

RQ4: A conceptualized ontology can be specified in different ontology languages such as DAML, OIL + DAML, OWL, RDF... To better understand the level of expressiveness of the ontology or ontologies used in a paper, we ask the following question: «Which ontology language does the authors employ in their work?»

3.1.2. Elaboration of the review protocol

Once the research questions are defined, we outline the review protocol (Fig. 2). First, we identify the electronic libraries which will constitute our information source. Then, we define the keywords search which we will use to query the electronic libraries. Then, we fix the inclusion and exclusion criteria to select the primary studies among the queried items. Finally, we present the quality criteria to evaluate the selected articles.

Selection of data sources

We select commonly used electronic libraries, including IEEEXplore,[3] SpringerLink,[4] ACM Digital Library,[5] and ScienceDirect.[6] We also use google scholar's search engine.

Keywords definition

To query the information sources, we selected the main two keywords, *ontology* and *UML*. However, in the execution phase, the results surpassed 15,000 items which suggest that these two keywords are too general, and the results are too broad. To obtain more relevant results, we redefined more specific keywords. We combined the two main keywords and the following

3 https://ieeexplore.ieee.org/Xplore/home.jsp.
4 https://link.springer.com/.
5 https://dl.acm.org/.
6 https://www.sciencedirect.com/.

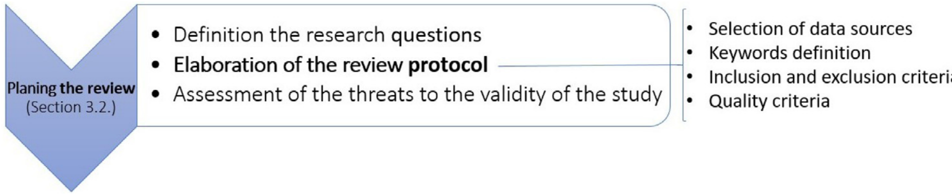

Fig. 2. Elaboration of the review protocol.

terms: *object-oriented modeling*, *model-driven engineering*, *software engineering*, *Semantic Web*, *reasoning*, *ontology engineering*, *transformation*, *mapping*. We stop collecting articles when we find ten consecutive titles in the search results which, in our opinion, are inconsistent with the query [15].

Inclusion and exclusion criteria

To filter the returned articles from the keyword search and keep relevant papers to our research questions, we define two inclusion criteria and eight exclusion criteria.

Inclusion criteria

The following are the inclusion criteria.

InC1: The article is written in English and accessible via the online libraries.
InC2: The article describes an approach combining UML and ontologies to solve a problem in a specific use case.

Exclusion criteria:

The following are the exclusion criteria.

ExC1: Posters or demonstrations that do not provide enough details about their contribution.
ExC2: Existing surveys which do not describe a contribution. They will be discussed separately in this paper.
ExC3: Non-accessible papers which we cannot recover.
ExC4: Duplicated papers returned from different electronic libraries.
ExC5: The contribution does not concern a combination of UML and ontologies. Some articles only use UML as a simple visualization support for the contribution of the article.
ExC6: Article not linked to UML or using another modeling language.
ExC7: The contribution of the article is more about the combination of OCL [13] and ontology than about UML and ontology.
ExC8: Books detailing papers already collected.

The exclusion criteria will be applied in three steps. The first four exclusion criteria will be applied directly to the collected papers to eliminate inaccessible items, demonstrations, posters, duplicates, and surveys. The remaining articles validate the first inclusion criterion. Then, the fifth and sixth exclusion criteria will be applied after a first reading of the summaries and introductions. Finally, the remaining articles will be thoroughly read, and the last two exclusion criteria will be applied to keep the articles according to their content. The selected articles validate the second inclusion criterion and constitute our primary studies.

Quality criteria

The definition of quality criteria in the SLR protocol makes it possible to evaluate the relevance of the primary studies collected to the research questions. We define our quality criteria as a list of questions admitting a yes, neutral or no, answer. These questions concern the details provided in the article on motivation, context, results, and limits of their contribution [14,15]. The criteria considered in our studies are:

C1: Do the authors provide a clear motivation for their work?
C2: Do the authors provide an detailed description of their proposed approach and their results?
C3: Are the limits of the study discussed?

3.1.3. Assessment of threats to the validity of the study

To ensure the quality of the review, we anticipate the potential threats and plan the corrective actions to reduce their impact [16]. In our case, the collection and selection of the primary studies are carried out by a single researcher (the Ph.D. student). To limit the occurrence of potential errors, we adopted the recommended verification techniques [14].

Collection strategy

Limiting our search to a single digital library and a fixed keyword list may lead us to miss some existing work. To reduce the effect of this threat:

1- We conducted the search using four different electronic databases. For each primary study, we searched its list of references to identify other possibly relevant articles.
2- We used synonyms and words related to the two main keywords in our research. These keywords have been updated several times when reading selected articles (whole or abstract). A new search was done each time a new keyword was added.

Selection strategy

Risk of bias in the selection process can occur because one person conducts it. To mitigate potential bias effects, we have taken the following actions:

1- The research process and the selection process were repeated twice in March 2018 and in August 2018. The renewal of the processes did not alter the identified studies. During the reviewing of the article, we re-conducted the search for the third time in July 2019. The obtained results do not show changes in previously validated articles. However, more recent articles were included.
2- We use a test-retest method. We randomly select 10% of primary studies and re-applied the inclusion and exclusion criteria. The items which were identified as not sure to exclude were left in the primary studies. In summary, we believe that the inclusion and exclusion criteria were sufficiently precise to minimize any selection bias.
3- When a positive or negative answer cannot be given to a quality assessment question, we attribute the response neutral to indicate it.

Although repeating the review protocol three times did not change the results, the non-determinism of some search engines in electronic libraries and the execution of this protocol by a single person, represent a risk for the reliability of this systematic review. The collected primary studies may not be reproducible by other researchers with the same results.

3.2. Performing the review

This section describes how we performed the SLR according to the previously defined protocol. Fig. 3 depicts the different

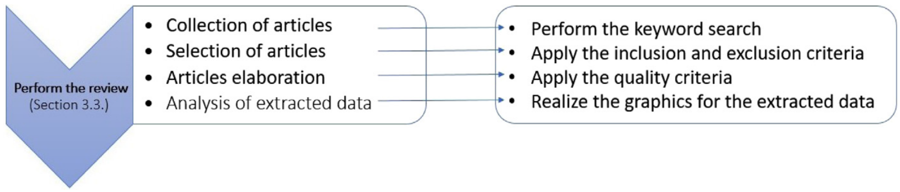

Fig. 3. Actions for performing the review.

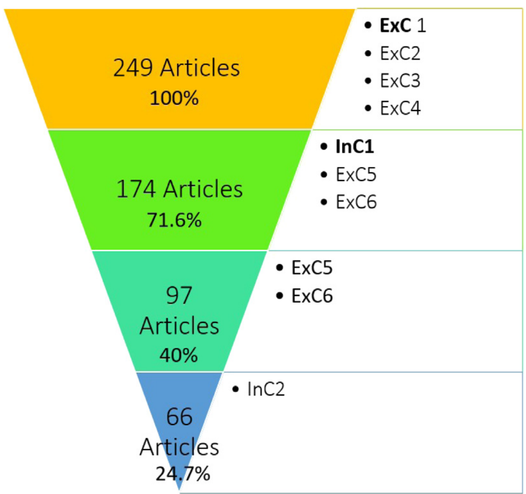

Fig. 4. Articles filtering process.

Table 2
Extracted data from the primary studies.

Attribute	Description
Bibtex entry	The bibtex entry of the study
Type	How was the study published: (Journal = J, Conference proceedings = C, Workshop = A, Technical report = R or Thesis = T.)
Year	The year of publication
Country	Countries where the authors are located
UML	The UML diagram(s) treated in the paper: (The diagram of classes, objects, use cases, activities, a UML profile, etc.)
Ontology language	The ontology language(s) addressed in the paper (OWL light, OWL DL, OWL full, RDF, DAML, OWL-S, etc.)
Transformation rules	Are the transformation rules explicitly defined and detailed? Some articles only mention the ability to have a transformation for some UML elements, but do not provide the transformations rules.
Transformation type	The study can describe the transformation rules between different types of UML diagrams into an OWL representation (UML to ontology) or describe the transformation rules between ontological languages into UML (Ontology to UML). The study can also describe a two-way transformation.
Tool	The name of the developed tool for the transformation process
Scientific locks	the scientific problems that are addressed in the article
Contribution & Innovation	The scientific contribution or application of the article
Limitations	The limits of the proposed approach discussed in the article or identified by the PhD student (first author of this paper).
Category	The category of the article according to the motivation of the study.

steps we follow. First, we query the selected electronic libraries with the set of defined keywords. A total of 249 studies were collected from different sources. Then, we apply our inclusion and exclusion criteria. Fig. 4 illustrates the three phases of the articles filtering.

First, the first four exclusion criteria are applied to eliminate inaccessible papers, demonstrations, posters, duplicate articles, and surveys. The remaining 174 articles are in English. Among them, there are four literature surveys which will be reviewed separately. We read the summary and the introduction of these studies then we apply the fifth and sixth exclusion criteria which exclude 77 articles. We thoroughly read the remaining 97 articles. After the application of the last exclusion criteria, 60 articles are included in this review. The selected articles validate the second inclusion criterion.

To address our research questions, we extract from the primary studies the different information described in Table 2. Based on the extracted data, we perform some statistical analysis to describe our primary studies. Fig. 5 illustrates the temporal and geographical distributions of papers. The investment rate in R&D in each country could partly explain the geographical distribution of the papers. We also notice an increase in the research activity as of 2010 with 66% (44 papers) of our primary studies published after 2010.

Proceeding from the quality criteria defined above, we evaluate the explanations in each primary study as to its motivation, the description of the proposed approach, and its limits. Overall this step require several re-readings to interpret incomplete or unclear information and to extract the information in Table 2. Conforming with the quality criteria defined previously, we give a score to each article. We attribute the score Y (yes) when it is fully met. We assign the score P (partially/neutral) if the quality criterion is not entirely satisfied. If the quality criterion is not satisfied, we assign the score N (no) to the study. Fig. 6 shows the average evaluation of our primary studies. The motivation and the context of the work are, on average, clearly explained (C1, C2). Indeed, many of the selected papers are well-detailed journal papers (32 journals) or publications in conferences, as shown in Fig. 7. Among our primary studies, we also count a thesis. However, it was particularly challenging to find a clear description of the proposed approaches' limitations, as the third evaluation criteria (C3) demonstrates in Fig. 6.

3.3. Existing surveys

We select four recent surveys that present an overview of the existing works dealing with the relationship between UML and ontologies. The authors in [17] review the existing solutions to automate ontologies' generation. Among these solutions, they discussed the transformations of UML representations (mainly the class diagram) into an ontological representation as a way to enable automatic ontology development. The survey of [18] described a classification of the approaches integrating UML

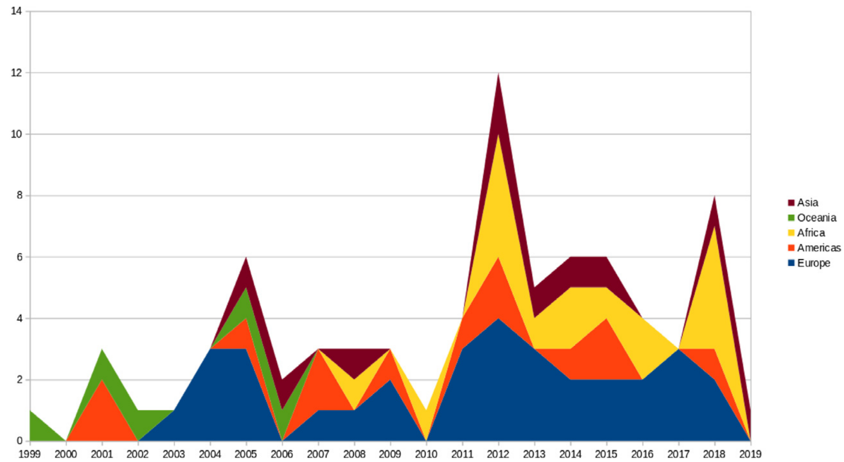

Fig. 5. Temporal and geographical distributions of the primary studies.

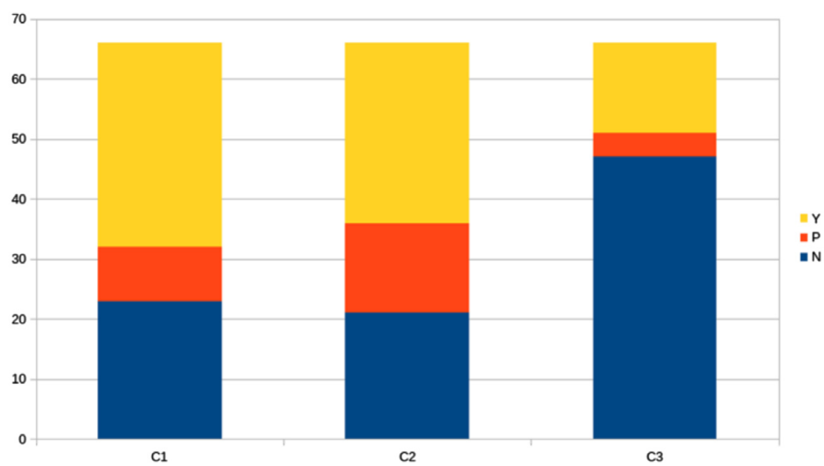

Fig. 6. The qualitative evaluation of the primary studies.

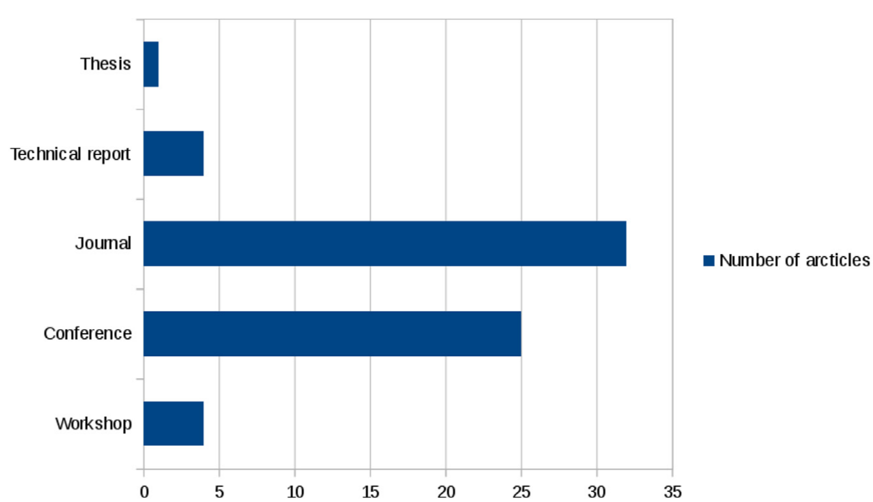

Fig. 7. Primary studies publication's type.

technologies and ontology technologies into three categories: (i) model validation using automated ontology reasoning techniques, (ii) model enrichment using ontologies to infer implicit knowledge from UML models and reconvert these inferences as facts in new models, (iii) ontological modeling using UML graphical notation for modeling ontologies. Although this classification provides a preview of the context of the existing work, it is insufficient to include other work that we have selected, exploring mainly the homogenization of different models using ontologies or formal modeling of knowledge within software engineering. We draw on the work of [18] to extend the categories classifying the different approaches combining UML and ontologies.

Table 3
A summary of the primary studies grouped by category.

Category	Articles
Knowledge re-acquisition	[20–25]
Unification of knowledge representation models	[26–32]
Ontology-based information systems	[33–37]
UML-based ontology modeling	[3,38–56]
Ontology-based software development	[18,57–69]
Model validation	[70–80]
Unassigned	[81–83]

Other literature surveys focus on the evaluation of the relation between the elements of the UML class diagram and ontology languages, in particular, OWL. The study of [12] gives a detailed comparison between the underlying paradigms of UML and ontology, as well as the similarities and the differences of the languages on the semantic and the syntactic basis. In a more recent study, [19] conducted a systematic review of the UML class diagram transformation rules to the OWL 2 ontology language. However, their review is limited to eighteen studies and focus only on the transformation rules from UML to OWL 2.

This section presented the activities accomplished to conduct our literature review. We collected articles from different electronic libraries. Then, based on our inclusion and exclusion criteria, we selected our primary studies which we thoroughly examined to retrieve relevant information. We also evaluated the included articles. The following sections details and discusses the results of the review regarding the defined research questions.

4. Classification of the existing researches

The first research question deals with the context and the motivations of the work coupling UML and ontologies. In order to answer to this question, we have identified the primary motivations of the existing approaches, and we propose six categories to classify them. Table 3 shows the primary studies distribution over these categories.

4.1. Knowledge re-acquiring

This category includes approaches using UML diagrams as a ready-to-use representation of a knowledge domain to generate or enrich ontologies. Researches which are classified in this category confirmed that the frequent use of UML in recent years in laboratories and industries, has created a valuable resource of domain models that can be reused and translated into ontologies. Their main aim was to enrich the Semantic Web and save time previously spent on specifying and conceptualizing a given area of interest.

4.2. Unification of knowledge representation models

This category regroups approaches using an ontological representation in order to unify various types of conceptual modeling languages. In practice, UML diagrams are not the only models for representing knowledge in the life-cycle of a project. Indeed, there are several representations of such a task. In this category, conducted researches aimed to build a homogeneous global model repository. They are thus facilitating the development of model processing tools, such as model mining, model checking, or model transformations tools. These approaches are of two types: The first type relies on unifying the heterogeneous models using model transformations then extracting the final ontology. However, the second type deals with extracting an ontology from the different models, then integrating the fragments into a final ontology.

4.3. Ontology-based information systems

This category includes approaches aiming to incorporate ontologies in the information systems of organizations. Their goal is to enhance the semantic interoperability between companies components as well as achieve web compliance, by using semantically formal description of the data, in the form of an ontology [84]. Several approaches focus on the direct mapping of relational databases to ontologies. However, we will focus in the remainder on methods combining ontology and UML.

4.4. Ontology-based software development

This category includes methodologies aiming to integrate ontology development into the software engineering community practices and vice versa. On one side, using UML as an intermediate minimizes the learning curve for developers, which promotes the use of ontology as a formal knowledge representation structure [65]. On other side, the software engineering discipline has gained matureness through the growing interest of industries. The latter led to the work of organizations such as OMG, which standardized methods and tools to support object-oriented software development. Hence, combining software engineering and ontology practices would be advantageous for both [85]. Indeed, the efforts to advocate UML as a modeling language for ontologies have fostered the integration of ontology practices into the software engineering community. The articles focusing on UML as a modeling language will be treated in the next category.

4.5. UML-based ontology modeling

This category includes efforts to use UML (or UML-similar) graphical notation to design ontologies. Related Works in this category advocated the use of UML as a modeling language able to represent ontologies. The main objective of these studies is to help novices with complex and unfamiliar ontologies languages.

4.6. Model validation

This category includes approaches using automated reasoning techniques for the verification and the validation of formal language models. The purpose of these approaches is to detect different errors in UML models. These errors deal with the conformance of the model to the OMG specification, the semantic errors that contradict common sense, or user requirements. UML diagrams are transformed into a formal ontological language to enable model validation.

In order to answer the first research question, we propose the six categories above classifying the existing work about UML and ontologies. The lack of a classification of the literature work makes it difficult to have an overview of the background research already conducted to place future contributions. We believe that our classification can help structure the existing work and facilitate the positioning of later research.

4.7. Issues addressed in the primary studies (RQ2)

The work in each category deals with a specific set of scientific issues. We have synthesized all the scientific locks identified in Table 4.

While the issues treated in the primary studies appears specific to the motivation of each work, we notice that some issues are recurrent in different use-cases. In particular, the problem of the semantic heterogeneity in distributed and delocalized companies (I7). In such networked companies, information management is essential to ensuring the effective operation of its internal

Table 4
Addressed issues in primary studies per category.

Category	Issues
Ontology-based software development	**I1.** Lack of a solution enabling automatic web services discovery, execution, composition, and inter-operation [57,64,65,69] **I2.** How to enable exchanging domain-specific objects within inter-agent messages [59] **I3.** Extracting domain knowledge present in different languages and tools into software models (UML) [63] **I4.** Existing representations of relationships in a social network (E-learning) lack semantics and cannot analyze the interaction between its elements [66] **I5.** Comprehension of software source code is difficult for the developer when the documentation becomes outdated or unavailable [61,67] **I6.** Knowledge acquired during the implementation and commissioning of the software is only reflected in the code [67] **I7.** Semantic heterogeneity in cross-sectorial, multiresources Enterprises [18,68] **I8.** Software developers are unfamiliar with Semantic Web ontologies [58,60,62]
Unification of knowledge representation models	**I9.** Harmonization of spatial data from heterogeneous sources is a demanding task because of interoperability problem at the syntactic and semantic level [26] **I10.** How to integration farming production data, including their geospatial dimension, and publish them as Linked data [28] **I11.** Lack of a unifying framework that respects most language features of the static structural components and constraints of ER, and UML [31]. **I7.** Semantic heterogeneity in cross-sectorial, multiresources Enterprises [29,30,32]
UML-based ontology modeling	**I12.** RDF graphs visualizations are low-level an insufficient to visualize the conceptual level of ontology building blocks such as classes and properties [40] **I13.** UML does not satisfy needs for representation of Ontology concepts borrowed from Description logic [42,46] **I14.** Ontology languages are not commonly known and the reader is quickly overloaded with textual definitions [43] **I15.** Existing modeling approaches do not enable ontology instance modeling using UML language [44] **I16.** Lack of support of automatic generation of operational ontology definitions [38,44,56] **I17.** Clients using the caGrid modeling tool have to ensure the semantic interoperability between their models manually [51] **I18.** Existing transformation approaches of UML-to-OWL neglect the problem of mapping the OWL data-type system to the UML type system and reverse [48] **I19.** RDF and DAML, have no standard graphical representation [53] **I1.** Lack of a solution enabling automatic web services discovery, execution, composition, and inter-operation [52] **I7.** Semantic heterogeneity in cross-sectorial, multiresources Enterprises [41,47] **I8.** Software developers are unfamiliar with Semantic Web ontologies [3,39,42,45,46,55]
Knowledge re-acquisition	**I20.** Creating new ontology from scratch is a difficult task that requires significant understanding of the domain knowledge and the ontology language [20,21] **I21.** How to fully capture the semantics of UML class diagrams [21] **I22.** knowledge acquisition bottleneck in ontology engineering **I23.** Maintaining the consistency between extracted data (in ontologies) and the UML model [24,25] **I24.** How to link the ontology extracted from the UML diagram to the open Web of Data [23] **I25.** UML language do not allow automatic reasoning unlike ontology languages [22]
Ontology-based information systems	**I26.** Organizations have a significant investment their current data model (usually relation database) want to continue using RDBMS technology at the implementation whilst introducing the ontology as a specification until it has the resources to update all its systems. [36] **I27.** Lack of a procedure for generating a specialized RDF schema and the set of Java classes from the UML class diagram [34] **I20.** Creating new ontology from scratch is a difficult task that requires significant understanding of the domain knowledge and the ontology language [33,35] **I24.** How to link the ontology extracted from the UML diagram to the open Web of Data [37]
Model validation	**I28.** UML model lacks a well-defined formal semantics [70,73] **I29.** UML models created in initial stages of software development are likely to contain design errors (unsatisfiable concepts, inconsistent behavioral diagram) that influence the software quality [71,72,75,76] , [79] **I30.** How to ensure the compliance of the UML diagrams to the domain of application [76–78,80] **I21.** How to fully capture the semantics of UML class diagrams [74] **I25.** UML language do not allow automatic reasoning unlike ontology languages [74]

services. However, heterogeneous enterprise applications, either at the business or at manufacturing levels, can cause a problem of misunderstanding and exchanging information, due to different viewpoints, for which these applications are developed. It also presents a risk of loss of information semantics when exchanging between those heterogeneous systems. The work we review in our survey addresses this issue by proposing the use of ontologies as models to trace relevant and shared information related to the knowledge domain in question. They argue that the ontology provides a fixed set of concepts whose meanings and relationships are shared and approved between users and systems. To build these ontologies, UML models are provided or extracted from different existing information sources. Then transformation rules are applied between the elements of the UML model (often a class diagram) and the constructors of an ontology language (often OWL DL).

Another recurrent issue in the literature is how to facilitate ontology development, especially for developers unfamiliar with formal ontology languages (I8, I14, I16, I20). Approaches addressing these issues propose to integrate technology components in model-driven engineering, that are currently available and familiar to developers in ontology engineering. We especially note approaches of [42,44,56] advocating Model-driven Architecture(MDA) guidelines for ontology development. MDA[7] is an OMG's software design approach that focuses on three different levels representing the abstraction layers of the application. The Computation Independent Model (CIM) is the first layer that describes the application and its environment. Then, the Platform Independent Model (PIM) presents a specification, usually in UML, of the requirements, structure, and functionality of the application with no technological details related to the platform. The Platform Specific Model (PSM) is the third layer, which is the closest to the final application code. The passage between these models rely on a mapping process, that is, a set of transformation rules to apply to source and target models. This architecture can be transferred to ontology engineering where the ontology is considered the PSM, and the UML-based model of the ontology is the PIM as recommended by the OMG.

7 https://www.omg.org/mda/.

Proceeding from the rise of the Semantic Web and the role of ontologies, the OMG called for a proposal of an Ontology Definition Metamodel (ODM) in 2003. Currently, ODM 1.1's specification defines metamodels for RDF and OWL and their corresponding UML profile. However, ODM reported several common problems in metamodel transformations from the UML profile to OWL metamodel [86]. The issue of structure conflation occurs when two constructs in the source metamodel map to a single construct in the target metamodel. It is the case for UML binary associations and class attributes, which both map to OWL properties. There is also the issue of structure loss when a complex construct is transformed into a collection of simpler constructs. For example, the UML N-ary associations are usually mapped to a class and a collection of OWL properties. Another issue stems from the lack of constructs in the target metamodel that is available in the source metamodel. Last but not least, the fact that UML is organized around classes and OWL is organized around classes, and individuals cause a problem of incompatible structural principles in the transformation. The ODM specification also does not include support for Semantic Web Services, although it was addressed in [52] who suggested a UML profile for modeling semantic Web services based on [42]'s metamodel and a novel metamodel for UML activity elements and constraints.

A third common issue in Table 4 is related to the semi-formal syntax of UML language, which makes it less machine exploitable than ontologies. Although the issue of automatically detecting errors in a UML diagram was addressed in model-driven engineering [87], existing modeling tools still provide little support to detect errors related to the well-formedness of a diagram to the OMG specification. These tools do not also detect semantic errors, such as contradicting constraints or cyclic generalizations. Such design errors propagate into the development phase, where the cost of repair is considerably high. The ontology-based approaches to treat this issue propose to formalize the UML diagrams, usually a class diagram, into a logic-based language, usually OWL [72,73]. Such formalization allows the use of inference engines[8] often associated with ontologies to check the consistency of the model. The most cited work setting the foundation of logic-based formalization of UML class diagram to enable automatic reasoning is [74]'s approach. The authors proposed a formalization based on ALCQI description logic and proved the reasoning complexity of EXPTIME-complete. Recent approaches [75–77] suggest using existing domain ontologies to facilitate requirements analysis process and extract UML diagrams from textual requirements. They use domain ontology in natural language processing (NLP) techniques to identify concepts in the requirements' textual description. These concepts are examined to extract classes, attributes, relations following the identified concepts.

5. Transformations between UML and ontologies

In this section, we address the third and fourth research questions. Most primary studies propose approaches based on a syntactic transformation between UML and an ontology language to address their identified issues. This section presents the transformations we extracted from these studies. Table 5 provides the general features of the transformation frameworks under study. For each proposal, we detail which UML diagram(s) and ontology language are concerned with the transformation rules. We also indicate what kind of transformation the study proposes (From UML to an ontology language, vice versa, or both ways). Some approaches provide details about the transformation rules and propose a tool to conduct the process.

[8] http://www.cs.man.ac.uk/~sattler/reasoners.html.

The evaluation of the information presented in Table 5 shows that UML class diagram (Fig. 8) and OWL languages (Fig. 9) are the main elements concerned in the transformation proposals. Nevertheless, OMG's UML profile for OWL and other UML profile proposals have important results is the reviewed articles. Indeed, many argue that the UML class diagram is insufficient to capture the expressiveness of ontology languages. Therefore, to overcome this limitation, the class diagram needs to be extended with stereotypes to include other OWL constructs. However, depending on the motivation of the work, some approaches do not need to capture ontology language syntax perfectly. For model validation, researchers aim to capture most of the UML class diagram in ontology languages.

To investigate the transformation rules, we selected the articles providing enough details about the UML to an ontology language transformation. Table 6 groups the proposed mappings between each treated class diagram constructs and DL constructs. Our results show that some disagreements exist for some constructs like the aggregation and composition relations. We verified the transformed class diagram constructs with the OMG specification. Only 68.7% of the UML constructs have a corresponding constructor in an ontology language (Fig. 10). Together, the present findings confirm [12]'s statement about the lack of a complete mapping between all the constructs of the two languages.

6. Conclusion and discussion

In this article, we presented an exploratory work of the literature combining UML and ontology. Indeed, in this study, we have reviewed 66 scientific research papers coupling these two knowledge representations since 1999 in order to summarize the literature of this issue and help sketch possible future directions in this area. One of our review objectives was to identify the essential research questions. Moreover, we analyzed the primary studies and outlined the landscape of the current research on the subject. The proceeding discussion highlights the limitations and challenges we identified.

There is an incomplete syntactic correspondence between UML class diagram constructs and ontology languages. Based on our findings, we know that a one to one transformation between the UML class diagram and an ontology language is nearly impossible. Although [12] stated the same conclusion before, the primary studies rarely address the repercussions of an incomplete transformation. The approaches aiming to extract existing knowledge in UML diagrams or evaluate the correctness of a diagram, often conduct a syntactic UML-to-ontology transformation. However, ignoring some constructs in the source model threatens the validity of the target model. Consequently, we raise the question about the possible ambiguities or errors caused by the transformation itself and how does it alter the obtained ontology.

UML diagrams, other than the class diagram, are often excluded from the knowledge extraction process. To model a domain of application, UML modelers often resort to different diagrams to capture both the static structure of the system and the dynamic behavior of its objects. However, the majority of the reviewed studies limit their scope to the class diagram, which links to the previous threat of neglecting essential domain knowledge present in other diagrams.

There is rarely an evaluation of the resulting ontology. Part of the primary studies claims they propose a ready to use ontology as the outcome of their approach. However, we hardly find an evaluation of the completeness of the resulting ontology. We believe information about the computational efficiency of the generated ontology and its ability to fulfill the required tasks,

Table 5
Transformations between UML and ontology languages(RQ3 & RQ4).

Articles	UML	Ontology languages	Transformation rules	Type of transformation	Tools
[22,35,45,50,51]	Subset of UML class diagram	OWL 1 (DL)	Detailed: [22,35,50] Not detailed: [41,45]	UML to ontology	Ontology development for the semantic web [35], caGrid [51], GenerateOWL (script) [22], XSLT processor [45]
[21,24,27,33,41, 62,66,79–81]	Subset of UML class diagram	OWL 2 (DL)	Detailed [21,24,33,62,66], Not detailed: [27,41,79–81]	UML to ontology	UML2OWL2 [24], UML2OWL [21], ATOM3 [81], OLED OntoUML Lightweight Editor [41]
[48]	Subset of UML class diagram	OWL 2 (DL)	Detailed	Tow-ways	
[40]	Subset of UML class diagram	OWL 2 (DL)	Not detailed	Ontology to UML	OWLGrEd
[23,28,34,37,68]	Subset of UML class diagram	RDF	Detailed [37] Not detailed [23,28,34,68],	UML to ontology	ShapeChange [28] XSLT processor [34]
[43]	Subset of UML class diagram	Suggested Upper Merged Ontology (SUMO)	Detailed	Ontology to UML	SUMO Translater
[74]	Subset of UML class diagram	ALCQI, DLRifd	Detailed	UML to ALCQI and DLRifd	–
[73]	Subset of UML class and object diagrams	OWL 2 (DL)	Detailed	UML to ontology	
[72]	Subset of UML class, object and state diagrams	OWL 2 (DL)	Detailed	UML to ontology	
[3]	Subset of UML class and object diagrams	RDF	Not detailed	UML to ontology	
[70]	subset of activity diagram	OWL 2 (DL)	Detailed	UML to ontology	
[18,36,49,56]	OMG's UML profile for OWL	OMG's OWL metamodel	Detailed [18,56] Not detailed [36,49]	UML to ontology	TwoUse Toolkit [18], OWL2EA [36], Acceleo Template [56], XSLT processor [49]
[64,65,69]	UML profile	OWL-S	Detailed [65] Not detailed [64,69]	UML to ontology	iSemServ [65,69]
[52]	UML profile	OWL-S	Not detailed	Two-ways	
[83]	UML profile	OWL-S	Detailed	UML to OWL-S	XSLT processor
[82]	UML profile	OWL 2 (DL)	Not detailed	Tow-ways	
[44,46]	UML profile	OWL 1 (DL)	Detailed [44], Not detailed [46]	UML to ontology	
[53,59]	UML profile	DAML	Detailed [53] Not detailed [59]	UML to ontology	
[55]	UML profile	OBO ontologies	Detailed	Ontology to UML	OBO-RO Editor/OBO2UML

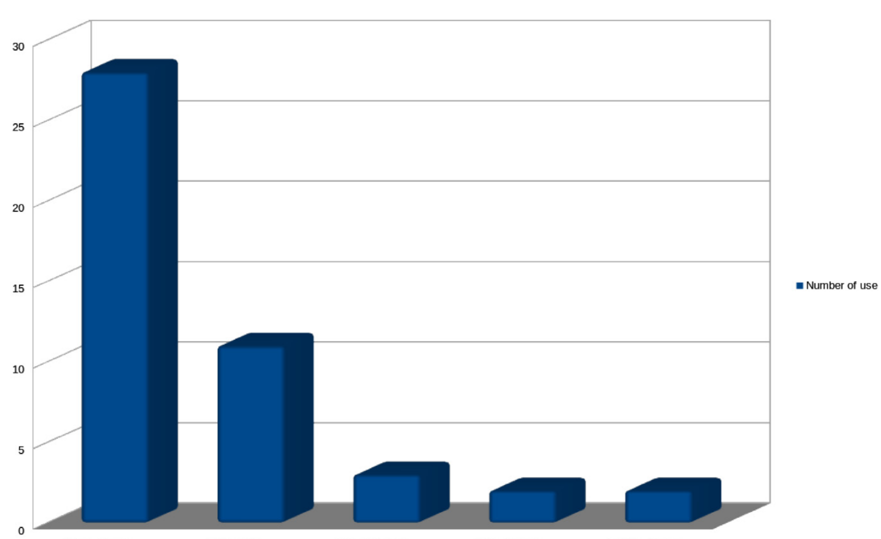

Fig. 8. The UML diagrams involved in the transformation.

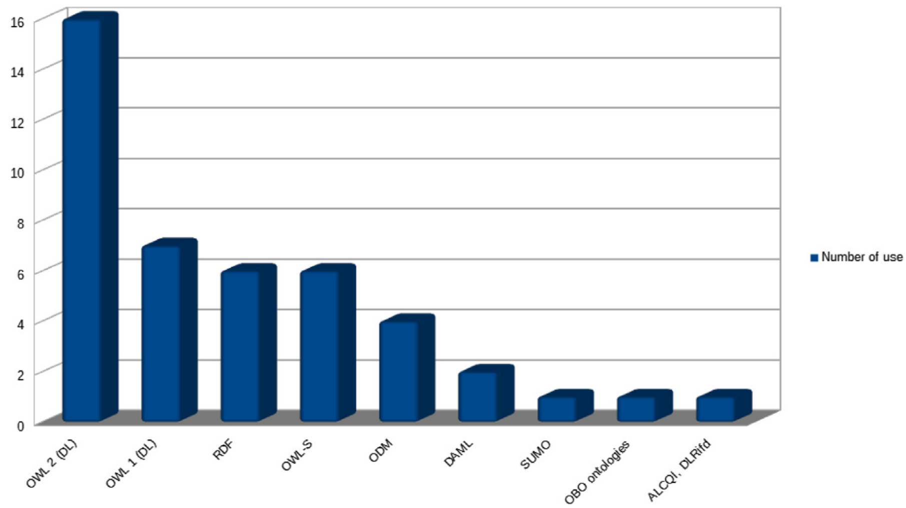

Fig. 9. Ontology languages involved in the transformation.

Table 6
DL formalization of a subset of UML class diagram.

Class diagram constructs	DL formalization	Articles
Class	Concept	[24,33,37,50,62,72–74]
Class attribute name	Role Data property Role Data property	[74] [37,62,72] [24,33,50]
Attribute type	Datatype Concept Datatype or concept	[24,37,72] [74] [33,50,88]
Attribute multiplicity	Qualified number restriction Functional data property	[72,74,88] [37,62]
Operation	Name: role or concept, parameters: roles	[74]
Enumeration	Datatype and DataOneOf (OWL 2)	[24,33]
Generalization	subsumption	[24,33,37,50,62,72–74]
Complete generalization-specialization	Equivalent concepts and union of concepts axioms	[24,74]
Disjoint generalization-specialization	Disjoint concepts axiom	[24,33,72–74]
Complete and disjoint generalization-specialization	Disjoint and union of concepts axioms Value Partitions Design Pattern	[24,33,72–74] [62]
Multiple inheritance	Subsumption of the intersection of the most general concepts	[24]
Association	Two roles A role and its inverse role	[37,73] [24,33,50,62,72,74]
Association multiplicity	Qualified number restriction and functional property Qualified number restriction	[33] [24,37,72–74]
Reflexive association	Reflexive role	[24]
Composition	Irreflexive and functional role SWRL constraints Asymmetric, irreflexive and functional role Role	[33] [72] [24] [74]
Aggregation	Role Functional and irreflexive role	[74] [33]
N-ary association	A concept et n roles	[24,74]
Association class	A concept and two roles	[24,50]
Inheritance between associations	role subsumption	[24,33,72]

such as query answering, classification, or consistency checking, are essential to defend the contribution [89].

Acceptance of UML profiles for ontology languages among developers. As we demonstrated, several approaches suggest extending UML language with stereotypes to define a profile englobing the ontology language constructs. Although this solution may overcome the challenges stated above, we raise concerns about the applicability of the proposed UML profile in real use cases. We found that articles recommending a UML profile for an ontology, start by identifying which construct in the ontology language lacks a similar construct in the class diagram, then define the appropriate stereotype. OMG's profile for ontology is also based on the ODM metamodel for OWL. In both cases, we believe the resulting profile notations resemble the ontology

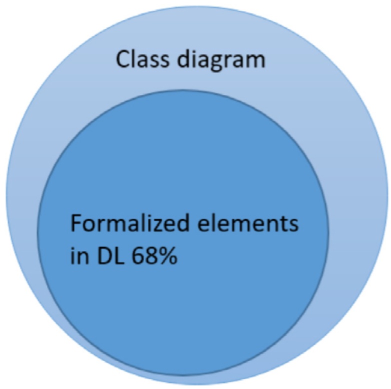

Fig. 10. Percentage of formalized class diagram constructs in DL.

language rather than the class diagram notations, which raises concerns about its usefulness to minimizes the learning curve for software developers.

Reconciling different conceptualizations in heterogeneous models. Some primary studies suggest unifying the knowledge dispersed in different heterogeneous models into one ontology. The different models are mapped into an ontology each, which are later merged into a final ontology. However, we do not find in such approaches details about managing conflicts in different models, such as reconciling older models of a system with newer ones, handling redundant information, or solving differences in the conceptualization itself.

Neglecting existing vocabularies when generating ontologies form UML diagrams. Ontology reuse has been a subject of interest to facilitate ontology engineering when constructing new ones [90]. Still, the primary studies do not consider reusing existing ontological knowledge systematically in the UML-to-ontology transformations.

To conclude, there is a significant overlap between UML and ontologies, which lead to considerable research work combining the two and for several transformation proposals. However, there is also a significant loss of information due to lack of features, differences in semantics, and modeling approaches. The paper presents our observations and explanations of the limitations and peculiarities in UML-to-ontology transformations. We introduce a classification of the existing research work which can assist both the UML users and ontology uses to create a better overview of the state of the art.

For future work, we aim to address the semantic equivalence between UML and ontologies rather than the syntactic comparison. Future work could define a shared mathematical definition of semantic equivalence for both UML and ontologies and how to conduct semantic-based transformation. Additionally, we believe that the potential of reusing domain ontologies to enrich the transformation semantically is promising and may reduce knowledge loss.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] OMG, Unified Modeling Language version 2.5.1, OMG Document Number: formal/17-12-05, 2017, https://www.omg.org/spec/UML/2.5.1/.

[2] J. McCarthy, Circumscription—A form of non-monotonic reasoning, in: Non-Monotonic Logic, Artificial Intelligence 13 (1) (1980) 27–39 (Special Issue).

[3] S. Cranefield, M. Purvis, UML as an ontology modeling language, in: Proceedings of the Workshop on Intelligent Information Integration, Stockholm, Sweden, 1999.

[4] G. Booch, Object-oriented Analysis and Design with Applications, second ed., Benjamin-Cummings Publishing Co., Inc., Redwood City, CA, USA, 1994.

[5] J. Rumbaugh, M. Blaha, W. Premerlani, F. Eddy, W. Lorensen, Object-oriented Modeling and Design, Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1991.

[6] I. Jacobson, Object-Oriented Software Engineering: A Use Case Driven Approach, Addison Wesley Longman Publishing Co., Inc., Redwood City, CA, USA, 2004.

[7] T. Gruber, A translation approach to portable ontology specifications, Knowl. Acquis. 5 (2) (1993) 199–220.

[8] F. Baader, I. Horrocks, C. Lutz, U. Sattler, An Introduction to Description Logic, Cambridge University Press, 2017.

[9] Semantic Web W3C, The Semantic Web, 2001. https://www.w3.org/2001/sw/. (Accessed 01 August 2018).

[10] Semantic Web Activity W3C, What is a Vocabulary?, 2015. https://www.w3.org/standards/semanticweb/ontology. (Accessed 01 August 2018).

[11] A. Pătraşcu, Comparative analysis between OWL modelling and UML modelling, Econ. Insights — Trends Challenges 4 (2) (2015) 87–94.

[12] K. Kiko, C. Atkinson, A Detailed Comparison of UML and OWL, Technical Report TR-2008-004, Department for Mathematics and Computer Science, University of Mannheim, 2005, URL https://madoc.bib.uni-mannheim.de/1898/.

[13] OMG, Object Constraint Language (OCL) version 2.4, OMG Document Number: formal/14-02-03, 2014, https://www.omg.org/spec/OCL/2.4/.

[14] B. Kitchenham, S. Charters, Guidelines for performing Systematic Literature Reviews in Software Engineering, Technical Report EBSE 2007-001, Keele University and Durham University Joint Report, 2007, URL http://www.dur.ac.uk/ebse/resources/Systematic-reviews-5-8.pdf.

[15] D. Calvaresi, D. Cesarini, P. Sernani, M. Marinoni, A.F. Dragoni, A. Sturm, Exploring the ambient assisted living domain: A systematic review, Journal of Ambient Intelligence and Humanized Computing 8 (2) (2016) 239–257.

[16] X. Zhou, Y. Jin, H. Zhang, S. Li, X. Huang, A map of threats to validity of systematic literature reviews in software engineering, in: Proceedings of the 23rd Asia-Pacific Software Engineering Conference (APSEC), 2016, pp. 153–160.

[17] I. Bedini, B. Nguyen, Automatic Ontology Generation : State of the Art, PRiSM Laboratory Technical Report, University of Versailles, 2007, pp. 1–15.

[18] F. Parreiras, Marrying model-driven engineering and ontology technologies: The twouse approach (Ph.D. thesis), University of Koblenz-Landau, 2011.

[19] M. Sadowska, Z. Huzar, Representation of UML class diagrams in OWL 2 on the background of domain ontologies, E-Inf. Softw. Eng. J. 13 (1) (2019) 63–103.

[20] A. Mehmood, A. Mehmood, B. Akhtar, Enrich existing ontologies with new knowledge from existing artifacts, in: Proceedings of the International Conference on Intelligent Technologies and Applications, 2018, pp. 630–641.

[21] Z. Xu, Y. Ni, W. He, L. Lin, Q. Yan, Automatic extraction of OWL ontologies from UML class diagrams: A semantics-preserving approach, World Wide Web 15 (5–6) (2012) 517–545.

[22] N. Gherabi, M. Bahaj, A new method for mapping UML class into OWL ontology, in: Software Engineering, Databases and Expert Systems, SEDEX, Int. J. Comput. Appl. (1) (2012) 5–9, (Special Issue).

[23] D. De Paepe, G. Thijs, R. Buyle, R. Verborgh, E. Mannens, Automated UML-Based Ontology Generation in OSLO2, in: Proceedings of the European Semantic Web Conference, 2017, pp. 93–97.

[24] O.E. Hajjamy, K. Alaoui, L. Alaoui, M. Bahaj, Mapping UML to OWL2 ontology, J. Theor. Appl. Inf. Technol. 90 (1) (2016) 126–143.

[25] M. Nekvasil, V. Svátek, M. Labský, Transforming existing knowledge models to information extraction ontologies, Lect. Notes Bus. Inf. Process. 7 (2008) 106–117.

[26] M. Strzelecki, A. Iwaniak, J. Łukowicz, I. Kaczmarek, Supporting spatial data harmonization process with the use of ontologies and Semantic Web technologies, Int. Arch. Photogrammetry Remote Sens. Spatial Inf. Sci. Arch. 7 (2013) 231–236.

[27] H. Elasri, E.M. El Abbassi, A. Sekkaki, M. Fahad, Semantic integration of UML class diagram with semantic validation on segments of mappings, 2018, CoRR, abs/1801.04482.

[28] R. Palma, T. Reznik, M. Esbrí, K. Charvat, C. Mazurek, An INSPIRE-based vocabulary for the publication of agricultural linked data, in: Proceedings of the International Experiences and Directions Workshop on OWL, 2016, pp. 124–133.

[29] H. Elasri, A. Sekkaki, Semantic integration process of business components to support information system designers, Int. J. Web Sem. Technol. 4 (2013) 51–65.

[30] K. Robles, A. Fraga, J. Morato, J. Llorens, Towards an ontology-based retrieval of UML class diagrams, Inf. Softw. Technol. 54 (1) (2012) 72–86.

[31] C. Keet, P. Fillottrani, An ontology-driven unifying metamodel of UML class diagrams, EER, and ORM2, Data Knowl. Eng. 98 (2015) 30–53.
[32] C. Quer, X. Franch, C. Palomares, A. Falkner, A. Felfernig, D. Fucci, W. Maalej, J. Nerlich, M. Raatikainen, G. Schenner, M. Stettinger, J. Tiihonen, Reconciling practice and rigour in ontology-based heterogeneous information systems construction, in: R.A. Buchmann, D. Karagiannis, M. Kirikova (Eds.), Proceedings of the Practice of Enterprise Modeling, Springer International Publishing, Cham, 2018, pp. 205–220.
[33] J. Zedlitz, J. Jörke, N. Luttenberger, From UML to OWL 2, Commun. Comput. Inf. Sci. 295 (2012) 154–163.
[34] S. Cranefield, Networked knowledge representation and exchange using UML and RDF, J. Digital Inf. 1 (8) (2001).
[35] S. Benslimane, M. Malki, M. Rahmouni, A. Rahmoun, Towards ontology extraction from data-intensive web sites: An html forms-based reverse engineering approach, Int. Arab J. Inf. Technol. 5 (1) (2008) 34–44.
[36] F. Loaiza, S. Wartik, J. Thompson, D. Visser, E. Kenschaft, The Best of All Possible Worlds: Applying the Model Driven Architecture Approach to a JC3IEDM OWL Ontology Modeled in UML, Institue For Defense Analyses Alexandria VA, 2014.
[37] M. Bahaj, J. Bakkas, Automatic conversion method of class diagrams to ontologies maintaining their semantic features, Int. J. Soft Comput. Eng. 2 (2014).
[38] K. Falkovych, M. Sabou, H. Stuckenschmidt, UML for the semantic web: Transformation-based approaches, Knowl. Transform. Sem. Web 95 (2003) 92–106.
[39] P. Fillottrani, E. Franconi, S. Tessaris, The ICOM 3.0 intelligent conceptual modelling tool and methodology, Semantic Web 3 (3) (2012) 293–306.
[40] R. Liepins, M. Grasmanis, U. Bojars, OWLGrEd ontology visualizer, in: Proceedings of the International Conference on Developers, vol. 1268, 2014, pp. 37–42.
[41] J. Guerson, T. Sales, G. Guizzardi, J. Almeida, OntoUML lightweight editor: A model-based environment to build, evaluate and implement reference ontologies, in: Proceedings of the 19th International Enterprise Distributed Object Computing Conference Workshops and Demonstrations, 2015, pp. 144–147.
[42] D. Đurić, MDA-based ontology infrastructure, Comput. Sci. Inf. Syst. 1 (1) (2004) 91–116.
[43] B. Hnatkowska, Automatic SUMO to UML translation, e-Inf. Softw. Eng. J. 10 (1) (2016) 51–67.
[44] D. Gasevic, D. Djuric, V. Devedzic, MDA-based automatic OWL ontology development, Int. J. Softw. Tools Technol. Transf. 9 (2) (2007) 103–117.
[45] D. Gašević, D. Djurić, V. Devedžić, Converting UML to OWL ontologies, in: Proceedings of the 13th International World Wide Web Conference on Alternate Track Papers & Posters, 2004, pp. 488–489.
[46] D. Djurić, D. Gašević, V. Devedžić, A UML profile for OWL ontologies, in: U. Aßmann, M. Aksit, A. Rensink (Eds.), Proceedings of the Model Driven Architecture, Springer Berlin Heidelberg, Berlin, Heidelberg, 2005, pp. 204–219.
[47] S. Chung, W. Tai, D. Osullivan, A. Boran, A semantic mapping representation and generation tool using UML for system engineers, in: Proceedings of the International Conference on Semantic Computing, ICSC, 2014, pp. 235–241.
[48] J. Zedlitz, N. Luttenberger, Data types in UML and OWL-2, in: Proceedings of the Seventh International Conference on Advances in Semantic Processing, 2013, pp. 32–35.
[49] D. Almeida Ferreira, A. Silva, UML to OWL Mapping Overview An analysis of the translation process and supporting tools, in: 7th Conference of Portuguese Association of Information Systems, 2013.
[50] A. Belghiat, M. Bourahla, Transformation of UML models towards OWL ontologies, in: Proceedings of the 6th International Conference on Sciences of Electronics, Technologies of Information and Telecommunications, 2012, pp. 840–846.
[51] J. McCusker, J. Phillips, A. Beltran, A. Finkelstein, M. Krauthammer, Semantic web data warehousing for caGrid, BMC Bioinformatics 10 (2009) S2.
[52] R. Grønmo, M. Jaeger, H. Hoff, Transformations between UML and OWL-S, in: Model Driven Architecture–Foundations and Applications, vol. 3748, 2005, pp. 269–283, doi: 10.1007/11581741_20.
[53] K. Baclawski, M. Kokar, P. Kogut, L. Hart, J. Smith, J. Letkowski, M. Aronson, Extending UML to support ontology engineering for the semantic web, in: Proceedings of the International Conference on the Unified Modeling Language, vol. 2185, Springer Berlin Heidelberg, 2001, pp. 342–360.
[54] S. Brockmans, R. Volz, A. Eberhart, P. Löffler, Visual modeling of OWL DL ontologies using UML, in: Proccedings of the International Semantic Web Conference, Springer Berlin Heidelberg, 2004, pp. 198–213.
[55] R.C. Waldemarin, C.R. de Farias, OBO to UML: Support for the development of conceptual models in the biomedical domain, J. Biomed. Inf. 80 (2018) 14–25.
[56] N. Laaz, S. Mbarki, An MDA approach based on UML and ODM standards to support big data analytics regarding ontology development, in: Proceedings of the Third International Conference on Smart City Applications, 2018, pp. 211–225.
[57] J. Yang, I. Chung, Automatic generation of service ontology from UML diagrams for semantic web services, in: Proceedings of the First Asian Conference on the Semantic Web, 2006, pp. 523–529.
[58] S. Baset, K. Stoffel, OntoJIT: Parsing native OWL DL into executable ontologies in an object oriented paradigm, in: Proceedings of the 13th International Workshop, OWL: EXperiences and Directions–REasoner Evaluation, in: LNCS, vol. 10161, Springer, 2017, pp. 1–14, http://dx.doi.org/10.1007/978-3-319-54627-8_1.
[59] S. Cranefield, M. Purvis, A UML profile and mapping for the generation of ontology-specific content languages, Knowl. Eng. Rev. 17 (1) (2002) 21–39.
[60] J. Evermann, Y. Wand, Ontology based object-oriented domain modelling: fundamental concepts, Requir. Eng. 10 (2) (2005) 146–160.
[61] R. Kadar, S.M. Syed-Mohamad, Semantic-based extraction approach for generating source code summary towards program comprehension, in: Proceedings of the 9th Malaysian Software Engineering Conference (MySEC), 2015, pp. 129–134.
[62] B. Bouihi, M. Bahaj, Building an e-learning system's OWL ontology by exploring the UML model, J. Theor. Appl. Inf. Technol. 87 (3) (2016) 380–387.
[63] X. Wang, C.W. Chan, Ontology modeling using UML, in: X. Wang, R. Johnston, S. Patel (Eds.), Proceedings of the 7th International Conference on Object-Oriented Information Systems, Springer London, London, 2001, pp. 59–68.
[64] R. Thiagarajan, M. Stumptner, A native ontology approach for semantic service descriptions, in: Proceedings of the Second Australasian Ontology Workshop on Advances in Ontologies, vol. 72, 2006, pp. 85–90.
[65] J. Mtsweni, E. Biermann, L. Pretorius, iSemServ: A model-driven approach to developing semantic web services, S. Afr. Comput. J. 52 (1) (2014) 55–70.
[66] B. Bouihi, M. Bahaj, Moodle's ontology development from UML for social learning network analysis, in: Proceedings of the International Conference on Learning and Optimization Algorithms: Theory and Applications, 2018, pp. 41:1–41:6.
[67] G. Chiozzi, R. Karban, L. Andolfato, A. Tejeda, A UML profile for code generation of component based distributed systems, in: Proceedings of the 13th International Conference on Accelerator and Large Experimental Physics Control Systems, 2011, pp. 867–870.
[68] G. Landolfi, A. Bami, G. Izzo, E. Montini, A. Bettoni, M. Vujasinovic, A. Gugliotta, A.L. Soares, H.D. Silva, An ontology based semantic data model supporting a maas digital platform, in: Proceeding of the International Conference on Intelligent Systems (IS), 2018, pp. 896–904.
[69] J. Mtsweni, E. Biermann, L. Pretorius, iSemServ: Towards the engineering of intelligent semantic-based services, in: International Conference on Web Engineering, Springer, Berlin, Heidelberg, 2010, pp. 550–559.
[70] H. He, Z. Wang, Q. Dong, W. Zhang, W. Zhu, Ontology-based semantic verification for UML behavioral models, Int. J. Softw. Eng. Knowl. Eng. 23 (02) (2013) 117–145.
[71] A.H. Khan, S.H.A. Musavi, A.-U. Rehman, A. Shaikh, Ontology-based finite satisfiability of UML class model, IEEE Access 6 (2018) 3040–3050.
[72] A. Khan, I. Porres, Consistency of UML class, object and statechart diagrams using ontology reasoners, J. Vis. Lang. Comput. 26 (2015) 42–65.
[73] A. Khan, E. Suenson, I. Porres, Representation and conformance of UML models containing ordered properties using OWL 2, in: Proceedings of the First International Workshop on Ordering and Reasoning, vol. 2011, 2011, pp. 1–12.
[74] D. Berardi, D. Calvanese, G. De Giacomo, Reasoning on UML class diagrams, Artificial Intelligence 168 (1–2) (2005) 70–118.
[75] H. Herchi, W. Ben Abdessalem, From user requirements to UML class diagram, 2012, CoRR abs/1211.0713.
[76] N. Jali, D. Greer, P. Hanna, Behavioral model generation from use cases based on ontology mapping and GRASP patterns, in: Proceedings of the 26th International Conference on Software Engineering and Knowledge Engineering, 2014, pp. 324–329.
[77] P. More, R. Phalnikar, Generating UML diagrams from natural language specifications, Int. J. Appl. Inf. Syst. 1 (8) (2012) 19–23.
[78] P. Kroha, R. Janetzko, J. Labra, Ontologies in checking for inconsistency of requirements specification, in: Proceedings of the 3rd International Conference on Advances in Semantic Processing, 2009, pp. 32–37.
[79] E.K. Elsayed, E.E. El-Sharawy, Detecting design level anti-patterns; structure and semantics in UML class diagrams, J. Comput. Phys. 13 (6) (2018) 638–654.
[80] M. Sadowska, Z. Huzar, Semantic validation of UML class diagrams with the use of domain ontologies expressed in OWL 2, in: Proceedings of the Software Engineering Conference: Challenges and Solutions, 2017, pp. 47–59.
[81] A. Belghiat, M. Bourahla, UML class diagrams to OWL ontologies: A graph transformation based approach, Int. J. Comput. Appl. 41 (3) (2012) 41–46.
[82] J. Zedlitz, N. Luttenberger, Transforming between UML conceptual models and OWL 2 ontologies, in: Proceedings of the Terra Cognita Workshop on Foundations, Technologies and Applications of the Geospatial Web, in Conjunction with the 11th International Semantic Web Conference, 2012, pp. 15–26.

[83] J.T. Timm, G.C. Gannod, Specifying semantic web service compositions using UML and OCL, in: Proceedings of the IEEE International Conference on Web Services (ICWS), 2007, pp. 521–528.
[84] M. Godinez, E. Hechler, K. Koenig, S. Lockwood, M. Oberhofer, M. Schroeck, The Art of Enterprise Information Architecture: A Systems-Based Approach for Unlocking Business Insight, Pearson Education, 2010.
[85] H. Knublauch, Ontology-driven software development in the context of the semantic web: An example scenario with Protege/OWL, in: Proceedings of the 1st International Workshop on the Model-Driven Semantic Web, 2004, pp. 381–401.
[86] R. Colomb, K. Raymond, L. Hart, P. Emery, C. Welty, G. Xie, E. Kendall, The object management group ontology definition metamodel, Ontol. Softw. Eng. Softw. Technol. (2006) 217–247.
[87] M. Richters, A precise approach to validating UML models and OCL constraints (Ph.D. thesis), University of Bremen, Germany, 2002.
[88] J. Zedlitz, N. Luttenberger, Conceptual modelling in UML and OWL-2, Int. J. Adv. Softw. 7 (1) (2014) 182–196.
[89] D. Vrandečić, Ontology evaluation, in: Handbook on Ontologies, Springer Berlin Heidelberg, Berlin, Heidelberg, 2009, pp. 293–313, http://dx.doi.org/10.1007/978-3-540-92673-3_13.
[90] A.S. Butt, A. Haller, L. Xie, RecOn: Ontology recommendation for structureless queries, Appl. Ontol. 11 (4) (2016) 301–324.