



# A systematic review on time-constrained ontology evolution in predictive maintenance

Alda Canito<sup>1</sup> · Juan Corchado<sup>2</sup> · Goreti Marreiros<sup>1</sup>

Published online: 6 October 2021

© The Author(s), under exclusive licence to Springer Nature B.V. 2021

## Abstract

With the modernization of industry and introduction of IoT, maintenance practices have been moving from reactive to proactive and predictive approaches. The identification of faults often relies on the analysis of real-time data provided by streams and unstructured sources. Ontologies have been applied to the maintenance field in order to add a semantic layer to the data and facilitate interoperability, and combined with other approaches for explainability and fault diagnosis, among others. In such a time-sensitive domain, it is important that ontologies go beyond static representations of the domain and allow not only for the incorporation of time related knowledge, but must also be able to adapt to new knowledge and evolve. This systematic review presents four research questions to provide a general understanding of the state of the art of the representation of time and ontology evolution in the predictive maintenance field. The results have shown that there are several ways of representing the evolution of knowledge that are fairly established and several specific evolutionary actions are discriminated and analyzed. Similarly, there is a diverse group of metrics that can be exploited to measure change and to establish evolutionary trends and even predict future stages of the ontology. Studies on the representation of time show us that it can be done either quantitative or qualitatively, with some approaches combining the two. Applications of these to the problem of ontology evolution are still in the open. Finally, results show that while applications of ontologies to the field of predictive maintenance are plenty, there are not many studies focusing on their evolution or in the effective application of their ability to reason with time constraints. The results obtained in this systematic review are particularly relevant for devising solutions that make use of the ontology's potential for time representation and evolution in the predictive maintenance field.

**Keywords** Ontology evolution · Temporal reasoning · Time-constrained relationships · Predictive maintenance · Systematic literature review

---

✉ Alda Canito  
alrfc@isep.ipp.pt

Extended author information available on the last page of the article

## 1 Introduction

The modernization and introduction of Internet of Things (IoT) technologies in industry is a big promotor of the adoption of predictive and proactive approaches to equipment maintenance in place of mainly reactive or scheduled interventions (Ansari et al. 2019; Saeed et al. 2019). The goal of predictive maintenance is to intervene in equipment before faults effectively take place, which relies on the continuous monitorization of equipment condition and identification of anomaly statuses or future faulty states (Ansari et al. 2019). For this to happen, it is important that data regarding equipment condition is obtained and processed in useful time (Ansari et al. 2019), as well as be properly understood by all intervening parties in all parts of the process—from its acquisition to its processing, analysis and delivery to the end users. Ontologies have found applications in industry and in the predictive maintenance field as a tool to semantically describe data and to provide reasoning and inference processes over it. Some architectures propose using ontologies to describe data captured through sensors (Klusch et al. 2015; Steinegger et al. 2017), processes (Saeed et al. 2019; Dibowski et al. 2016) or for the delivery of results; Smoker et al. 2017). Reasoning with ontologies has found applications for providing diagnostics, recognition of qualitative fault states and distinguishing between different types of failure (Delgoshaei et al. 2017; Klusch et al. 2015; Steinegger et al. 2017; Ferrari et al. 2017), and can be combined with other mechanisms for predictive maintenance purposes.

Predictive maintenance activities require time-sensitive and time-aware processes. A lot of information arrives in real-time and through streams, which may then be analyzed through machine learning and data mining algorithms and is bound to change its meaning over time in a form of concept drift, as the characteristics that define normal or abnormal behavior slowly transform over time. While there is a considerable amount of work regarding the applications of ontology to the predictive maintenance field, they often fail to understand and model these requirements. Most of the available ontologies focus on the description of a static domain and do not make much use of temporal representation or temporal constraints. For ontologies to change over time and adapt to the dynamic domains they aim to describe, they must be able to evolve in some way. Ontology evolution is described in Stojanovic (2004) as the process through which an ontology adapts to the changes in a domain in a timely fashion, while consistently propagating the changes to other artifacts that depend on it. This process often relies on the identification of which changes occurred in the domain that made the ontology outdated, the materialization of those changes into specific evolutionary actions and the execution of those—followed by consistency checks to ensure that the new ontology version is logically sound. Predictive maintenance approaches often have to model the gradual change in equipment behavior as its components wear down with use. As such, there is a variation on what are the characteristics of normal/abnormal behavior over time, and similarly of what constitutes a failure. Ontology evolution, therefore, becomes a necessity and can be employed to identify and model these changes, ensuring that the domain description is up-to-date and remains accurate. Many existing approaches to ontology evolution, however, fail to provide formalisms for temporal abstract notions (Shaban-Nejad and Haarslev 2015).

The purpose of this work is to understand the different ways ontologies can be used to represent time-sensitive domains and to explore how these domains evolve over time in the face of new knowledge. To do so, four research questions are presented, and

existing studies that may answer to them are analyzed and discussed in order to reach an overview of the current state of the art in the topics of time representation, identification of change, evolution of the domain and the applications of these in the predictive maintenance field.

The remainder of this document is structured as follows: Sect. 2, in which the procedure taken in this systematic review is presented in detail and all steps of the process are accounted for. In this section, we go from the definition of the research questions to the selection of the works to be included in this review, while explicating the inclusion and exclusion criteria and the data sources considered. In Sect. 3, we show which answers to the different research questions are proposed by the analyzed studies to the best of our knowledge. In Sect. 4, we take a deeper look at these studies and identify potential challenges and future directions for research. Finally, Sect. 5 outlines the Conclusions and Future Work.

## 2 Method

The literature review will focus on the analysis of existing Ontology Evolution methodologies, in order to assess how they answer these very questions and how they handle the field's inherent challenges. Analysing the challenges involved will allow us to establish how the new methodology can improve on the existing ones and a better understanding of the existing problems and challenges in this field of study. Furthermore, the applicational context of this work lies in the field of ontologies for Predictive Maintenance and industrial processes, and therefore existing works in this area will also be explored.

The goal of a systematic review of the literature is to extract, analyse and interpret a corpus of work pertaining to a specific field of study. Systematic reviews are meant to be thorough and to follow a sequential, detailed, specific and repeatable workflow that allows other researchers to achieve similar results. The methodology described by Kitchenham (2004) proposes that systematic reviews should follow a number of steps, from the development of the review protocol; entailing the rationale and research questions, the processing, analysis, and demonstration of the research's results. As such, this review starts by presenting the Research Questions and the motivating problem. Then, the data sources selected for this review will be presented, along with the process of selecting and using the most adequate combination of search terms that generated a query that could be equally used in the different data sources. Inclusion and Exclusion criteria, along with quality assessment criteria are described. Finally, a detailed account of the data extraction process is given, showing how the articles found in those data sources were continuously filtered until the final number of articles to review is reached.

### 2.1 Research questions

For the purposes of this systematic review, we define the main research question as: "What are the current applications of time-based reasoning in ontology evolution in the field of predictive maintenance?". In order to properly answer this question, we can divide the main question into four subdomain questions, which can be found in Table 1. The first question is mainly concerned with the representation of changes made to an ontology: what methods are used to describe and implement specific evolutionary actions and the conditions in which they take place, and how previous versions of the ontology are maintained.

**Table 1** Research questions

	Research question
RQ1	What methods have been used to represent the evolution of knowledge over time?
RQ2	What mechanisms can be employed to identify and measure change in knowledge?
RQ3	How can temporal validity be applied to knowledge evolution?
RQ4	What methods of knowledge acquisition and time representation are commonly applied in the predictive maintenance field?

**Table 2** Electronic databases

Identifier	Database	URL
DS1	Web of Science	<a href="https://www.webofknowledge.com/">https://www.webofknowledge.com/</a>
DS2	ACM Digital Library	<a href="https://dl.acm.org/">https://dl.acm.org/</a>
DS3	IEEE Explore	<a href="https://ieeexplore.ieee.org/">https://ieeexplore.ieee.org/</a>

The second question, on the other hand, focuses in a related but distinct task, that is: the identification and measurement of change, i.e., the application of metrics for ontology evolution. Here, it is relevant to assess which metrics exist, which are the most popular approaches and what characteristics of the ontology or other sources of knowledge they should be applied to. The third question enters in the domain of time representation, allowing us to explore the different ways through which time is frequently represented in ontologies and how the temporal validity of concepts and relationships can be represented and utilized. This question also opens the possibility of these representations being used for ontology evolution. Finally, question four will attempt to find specific applications of all these topics in the field of predictive maintenance.

## 2.2 Data sources

Identification of the data sources to use in the systematic review is the first step of the process. According to Paré et al. (2015), in order to maintain clarity and rigor, it is advisable to maintain a small number of comprehensible data sources. Furthermore, the data sources selected make use of the same operators in the same fields, and thus it is possible to use the same query string in all of them with little to no modification. Table 2 identifies the electronic databases chosen for this study. There is some overlap within these data sources, which is accounted for in a following step.

## 2.3 Search terms

Considering the field of application of this work, along with the research questions to keep in mind, a number of domains were selected to assess not only how many studies have been made for each of the considered subdomains, but also how the combination of these affects the number of results. As such, consider the following domains and subdomains selection and the respective keywords used to define each one, as seen in Table Table 3:

Search terms were applied to Title, Abstract and Keywords. Different combinations of these domains were attempted and evaluated over the number of results they provided

**Table 3** Domains and keywords used to select search terms

Domain	Keywords
Ontology	“Ontology”
Knowledge evolution	(“Knowledge evolution” OR “knowledge acquisition” OR “schema induction” OR “ontology learning” OR “domain expansion” OR “ontology evolution”)
Temporal information	(“Time representation” OR “representation of time” OR “temporal information” OR “time-based reasoning” OR “temporal reasoning” OR “reasoning with time” OR “temporal representation”)
Predictive maintenance	(“Fault prediction” OR “predictive maintenance” OR “fault detection”)
Stream learning	(“Machine learning” AND (“event” or “stream”))

before the final search query was reached. The results of these attempts can be seen in Table 4. Because “Ontology” is a fundamental domain that must be present in all works, it always appears as an intersection in the queries.

Unfortunately, the intersection of all the original domains does not yield any results. Similarly, the intersection between Knowledge Evolution and Temporal Information returns a very low number of works. As such, the union between these two fields (intersection with ontology for the reasons stated above) was attempted, which returned a bigger number of hits that include the few ones returned by the intersection. While this makes for less focused results, studies that address at least one of the issues may provide nonetheless interesting insights. Additionally, we can see there are very few works currently dealing with ontologies in the field of predictive maintenance, suggesting that another union was likely to be necessary. Because the intersection between Ontology, Stream Learning and Knowledge Evolution is particularly interesting to this research, it will be included as the union with the previously mentioned one. The final query string can be found in Table 5.

## 2.4 Quality assessment: inclusion and exclusion criteria

In order to assess whether a specific article should be included in this review or not, a set of rules were devised. These will evaluate a number of parameters that are not related to the contents of the article, such as if it is peer-reviewed, or if it is in book format; while others will be pertaining to the quality of the article’s content and the relevance of its contributions. Proposals are to be considered in terms of their novelty and the theoretical fundaments they aim to support; otherwise, preference to fully developed and tested systems will be given. Table 6 reflects the criteria to considered for inclusion of a given work, and Table 7 those for exclusion:

## 2.5 Data extraction

Figure 1 illustrates the decision-making process used in this systematic review. This process begins with the data extraction from the different digital libraries by applying the aforementioned search terms, and by selecting the last 5 years of studies (the time interval was set to 2015–2019). The last five years were selected as the time period for this review, considering it is a trade-off between gathering data on the most recent research while also

**Table 4** Combination of the domains and respective results

Domains	Results				Total
	WoS	ACM	IEEE		
Ontology	23,820	3302	3074		30,196
Knowledge evolution	2632	483	336		3451
Temporal information	3738	775	1243		5756
Predictive maintenance	12,202	913	5145		18,260
Knowledge evolution OR temporal information	6369	1258	1529		9156
Knowledge evolution AND temporal information	4	0	2		6
Ontology AND (knowledge evolution OR temporal information)	397	154	85		636
Ontology AND predictive maintenance	18	2	11		31
Ontology AND (knowledge evolution OR temporal information) OR (ontology AND predictive maintenance)	415	156	105		676
Ontology AND (knowledge evolution OR temporal information) AND (ontology AND predictive maintenance)	0	0	0		0
Ontology AND stream learning AND (temporal information OR knowledge evolution)	36	9	0		45

**Table 5** Query string

(((("Knowledge evolution" OR "knowledge acquisition" OR "schema induction" OR "ontology learning" OR "domain expansion" OR "ontology evolution") OR ("time representation" OR "representation of time" OR "temporal information" OR "time-based reasoning" OR "temporal reasoning" OR "reasoning with time" OR "temporal representation")) AND "ontology")) OR (("ontology") AND ("fault prediction" OR "predictive maintenance" OR "fault detection"))))

OR

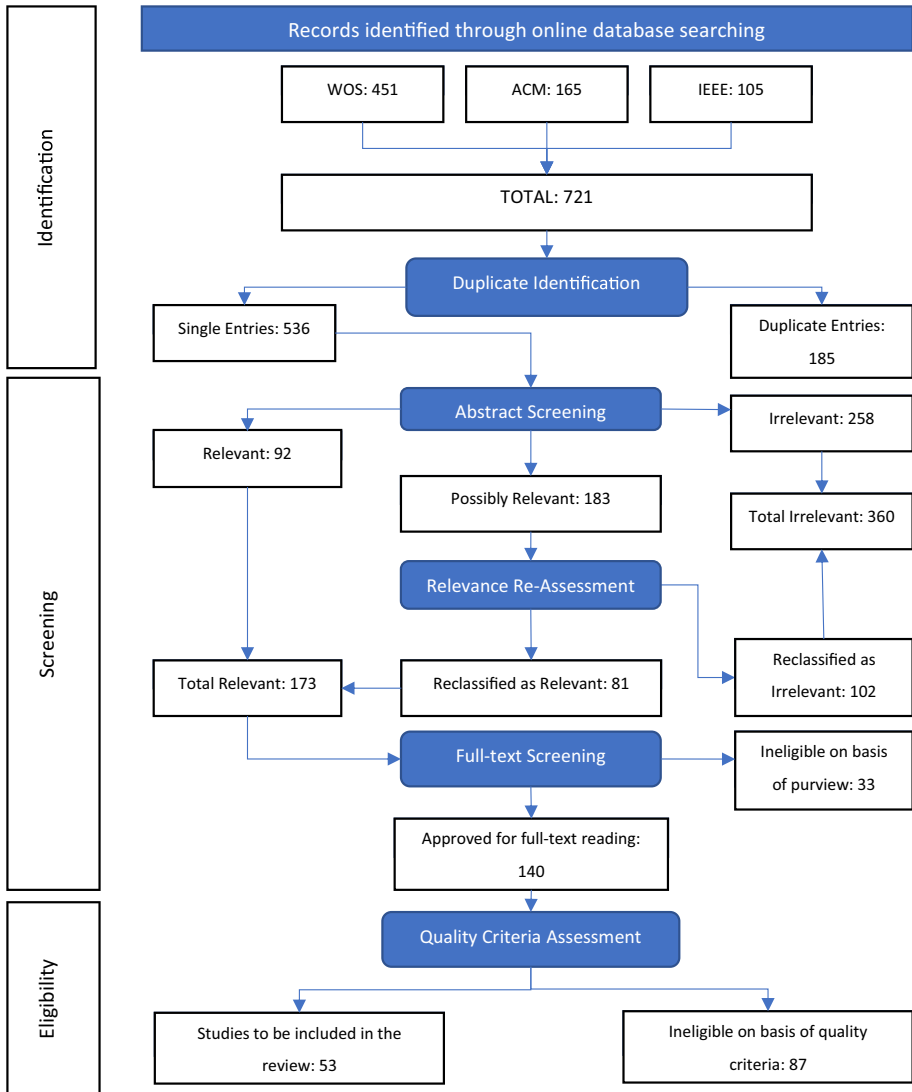
("Ontology" AND "machine learning" AND (("event" OR "stream" OR "time representation" OR "representation of time" OR "temporal information" OR "time-based reasoning" OR "temporal reasoning" OR "reasoning with time" OR "temporal representation") OR ("knowledge evolution" OR "knowledge acquisition" OR "domain expansion" OR "ontology evolution")))

**Table 6** Inclusion criteria

	Inclusion criterion
IC1	The source belongs to the field of knowledge engineering/representation in computer science
IC2	Source is peer-reviewed
IC3	The source describes a significant contribution to the fields of study
IC4	The source describes a methodology or a tool for ontology evolution or evaluation, including new evaluation tools and systems
IC5	The source describes how the concept of Time is (or can be) handled, i.e. how it is represented and how reasoning is affected
IC6	The source must describe a tool, an application scenario, or otherwise have a strong theoretical proposal
IC7	Scenarios of application of ontologies that represent and reason with time

**Table 7** Exclusion criteria

	Exclusion criterion
EC1	The source is over 5 years old
EC2	The source is either a book chapter, dissertation, systematic review or thesis
EC3	The source is not written in English
EC4	The source focuses on information extraction for a very specific domain, lacking adaptability to different domains
EC5	Contribution to the field of ontologies is unclear or not the focus of the source
EC6	The source does not clarify the methods through which ontological concepts or relations are extracted
EC7	The source overly focuses on the applications of the ontology and not on its semantics, learning or evolution
EC8	The source is focused on describing and applying NLP techniques for information extraction purposes, and not on the application of their results to an ontology
EC9	Sources that describe the semantic technologies employed but not how these are effectively used in their use-cases are not to be considered



**Fig. 1** Flow diagram showing the eligibility process

trying to pinpoint older, more foundational works. This search resulted in a total of 721 records to review.

The first phase in the process was the removal of duplicates, which was done in two steps: a first one, aided by JabRef's<sup>1</sup> identity function, identifying 34 duplicate entries, and a second, manual one which identified an additional 151 duplicates. Whenever two articles were published by the same author in the same field and pertaining to continuation of the

<sup>1</sup> <https://www.jabref.org/>.



same study, only the most recent one was considered (Kitchenham 2004). Similarly, in this phase it was possibly to identify references that were not to articles and were therefore excluded, as well as any inaccessible articles and those not written in English. While most of these cases were detected in this phase, a few were still discovered in later stages. Ultimately, the identification phase resulted in the removal of 185 entries.

The resulting 536 records moved on the next phase, Screening. This process was separated in three different subtasks: (1) Abstract Screening, (2) Relevance Re-Assessment and (3) Full-text Screening.

Abstract Screening consists of reading the abstracts of all the articles and classify them according to three different categories: Relevant, Possibly Relevant and Irrelevant. These are meant to assess whether the topics of the articles are relevant to the research questions or not. Whenever it is not possible to assess this immediately, the articles are classified as Possibly Relevant, to be screened later. This process allows for a quick elimination of articles focused on different topics, those do not describe new methodologies or tools, or any others which are not written in English or are in the appropriate formats but still ended up in the results. After this task was undertaken, 92 papers were classified as Relevant, 183 as Possibly Relevant, and 258 were considered Irrelevant for the purposes of this review and therefore discarded.

The Relevance Reassessment task pertains to evaluate those articles which were classified as Possibly Relevant. Unlike the Full-text screening that is done in a later stage, this is merely a diagonal read of the work to quickly obtain information that was not clear through the abstract. After this process, 102 articles were reclassified as Irrelevant and discarded, while 81 were added to the Relevant pile and moved to the next stage, resulting in a total of 140 articles approved for Full-Text Reading and completing the Screening phase.

The Eligibility stage pertains to how each individual article describes a work that is directly related to the potential answers of the Research Questions. To do so, a full, deep reading and understanding of each article is in order. Each study is analyzed according to the established quality criteria and only those describing clear contributions to the field or relevant application scenarios are approved. This Eligibility stage also includes sorting the articles according to their contents for easier description and categorization of their contents. As a result, 87 articles were deemed ineligible on basis of the quality criteria assessment, while 53 are to be included in this review.

### 3 Results

This section describes the results obtained through this systematic review, focusing on how the works retrieved through this process help answering the research questions.

#### 3.1 RQ1: what methods have been used to represent the evolution of knowledge over time?

Of the 53 retrieved search results, 15 focus on the subjects related to representing and managing changes in domain knowledge. While ontology evolution is considered the process through which an ontology is modified over time, ontology versioning is a stronger version of this process by annotating which modifications occurred and when are properly described and stored, with the possibility of accessing previous states. According to Bayoudhi et al. (2019), there are two main approaches to ontology

versioning: naïve strategies, which store all ontology versions independently—which may lead to overheads in storage; and change-based approaches, which store a reference accepted version of an ontology as the base point, storing the incremental modifications that must be applied to the reference ontology on runtime—which may lead to overheads in processing time. In an alternate version of this, Timestamp-based approaches opt by tagging each axiom with a time interval stating their validity. In the case of change-based approaches, change must be defined and described in such a way that allows for the dynamic reconstruction of an ontology version.

Of the retrieved results, four describe change-based approaches (Kondylakis and Papadakis 2018; Tsalapati et al. 2017; Shaban-Nejad and Haarslev 2015; Peixoto et al. 2016), while one (Grandi 2016) describes a naïve approach. Bayoudhi et al. (2017) tries to bridge the gap between the approaches by materializing both the reference and latest versions. The other works do not specify in which approach could be used, focusing instead on other subjects.

When it comes to describing and materializing change, two studies systematize all possible evolutionary actions through means of an ontology (Kondylakis and Papadakis 2018; Peixoto et al. 2016), while one study stores this information in a relational database (Tsalapati et al. 2017), and another (Grandi 2016) in XML. Four approaches describe the modifications through means of axioms and rules (Bayoudhi et al. 2017; Zheleznyakov et al. 2019; Wang et al. 2015; Mahfoudh et al. 2015). Two studies, Wang et al. (2015) and Zheleznyakov et al. (2019), focus specifically on issues and limitations of ontology evolution when using DL-Lite.

Six works describe and apply graph transformations to enact the evolutionary process (Tsalapati et al. 2017; Shaban-Nejad and Haarslev 2015; Grandi 2016; Zheleznyakov et al. 2019; Mahfoudh et al. 2015) and six others provide insights in how to deal with consistency issues that may arise when evolutionary action is taken (Peixoto et al. 2016; Bayoudhi et al. 2017; Wang et al. 2015; Touhami et al. 2015; Gaye et al. 2015; Sad-Houari et al. 2019). Of these, one (Mahfoudh et al. 2015) focuses on checking consistency before applying the changes.

Three works focus on the identification of candidate evolutionary actions (Kondylakis and Papadakis 2018; Grandi 2016; Osborne and Motta 2018; Benomrane et al. 2016). Of these, one (Osborne and Motta 2018) aims to ease this step by allowing the user to specify requirements for new candidates and another (Benomrane et al. 2016) allows an ontologist to select actions they want to see executed.

Three works (Kondylakis and Papadakis 2018; Grandi 2016; Mahfoudh et al. 2015) focus on the traceability of changes and provenance queries, with (Tsalapati et al. 2017) being solely concerned with query rewriting after the ontology has changed.

Finally, twelve works offer insights about in which ways an ontology can be changed (Kondylakis and Papadakis 2018; Shaban-Nejad and Haarslev 2015; Peixoto et al. 2016; Grandi 2016; Bayoudhi et al. 2017; Wang et al. 2015; Mahfoudh et al. 2015; Touhami et al. 2015; Gaye et al. 2015; Sad-Houari et al. 2019; Benomrane et al. 2016). These studies assess and categorize the possible evolutionary actions that can be performed, materializing them into simpler, executable operations. As such, the difference between two ontology versions can be described as the set of operations that must be executed to transformed into the other. A summary of these operations is represented in Table 8:

For ease of reading, the entries presented in Table 8 have been grouped into categories of similar actions, even if in some studies they may be named differently. It should be noted that in some cases, such as Bayoudhi et al. (2019), specify only that axioms can be either added or removed, but do not specify types of axioms. As such, it is assumed that the

**Table 8** Ontology evolution operations

Study	Operation
Kondylakis and Papadakis (2018), Shaban-Nejad and Haarslev (2015), Peixoto et al. (2016), Grandi (2016), Bayoudhi et al. (2017), Mahfoudh et al. (2015), Touhami et al. (2015), Gaye et al. (2015), Sad-Houari et al. (2019), Benomrane et al. (2016)	Add Class
Kondylakis and Papadakis (2018), Shaban-Nejad and Haarslev (2015), Peixoto et al. (2016), Grandi (2016), Bayoudhi et al. (2017), Mahfoudh et al. (2015), Touhami et al. (2015), Sad-Houari et al. (2019)	Add SuperClass
Kondylakis and Papadakis (2018), Shaban-Nejad and Haarslev (2015), Peixoto et al. (2016), Bayoudhi et al. (2017), Mahfoudh et al. (2015), Touhami et al. (2015), Gaye et al. (2015), Sad-Houari et al. (2019)	Add Property
Kondylakis and Papadakis (2018), Shaban-Nejad and Haarslev (2015), Peixoto et al. (2016), Bayoudhi et al. (2017), Mahfoudh et al. (2015), Touhami et al. (2015), Gaye et al. (2015), Sad-Houari et al. (2019)	Delete Property
Kondylakis and Papadakis (2018); Shaban-Nejad and Haarslev (2015), Peixoto et al. (2016), Grandi (2016), Bayoudhi et al. (2017), Touhami et al. (2015), Sad-Houari et al. (2019)	Delete SuperClass
Shaban-Nejad and Haarslev (2015), Grandi (2016), Bayoudhi et al. (2017), Touhami et al. (2015), Gaye et al. (2015), Sad-Houari et al. (2019), Benomrane et al. (2016)	Delete Class
Kondylakis and Papadakis (2018), Bayoudhi et al. (2017), Mahfoudh et al. (2015), Touhami et al. (2015)	Change Domain
Kondylakis and Papadakis (2018), Bayoudhi et al. (2017), Mahfoudh et al. (2015), Touhami et al. (2015)	Generalize Domain
Kondylakis and Papadakis (2018), Bayoudhi et al. (2017), Peixoto et al. (2016), Mahfoudh et al. (2015)	Add Super Property
Kondylakis and Papadakis (2018), Peixoto et al. (2016), Bayoudhi et al. (2017), Mahfoudh et al. (2015)	Rename Property
Kondylakis and Papadakis (2018), Bayoudhi et al. (2017), Peixoto et al. (2016)	Delete Super Property
Kondylakis and Papadakis (2018), Bayoudhi et al. (2017), Mahfoudh et al. (2015)	Generalize Range
Kondylakis and Papadakis (2018), Bayoudhi et al. (2017), Mahfoudh et al. (2015)	Change Range
Kondylakis and Papadakis (2018), Bayoudhi et al. (2017), Mahfoudh et al. (2015)	Specialize Domain
Kondylakis and Papadakis (2018), Bayoudhi et al. (2017), Mahfoudh et al. (2015)	Specialize Range
Bayoudhi et al. (2017), Mahfoudh et al. (2015), Touhami et al. (2015)	Add Cardinality Restriction
Bayoudhi et al. (2017), Mahfoudh et al. (2015), Touhami et al. (2015)	Remove Cardinality Restriction
Kondylakis and Papadakis (2018), Bayoudhi et al. (2017)	Change Datatype
Kondylakis and Papadakis (2018), Bayoudhi et al. (2017)	Add Comment
Kondylakis and Papadakis (2018), Bayoudhi et al. (2017)	Delete Comment
Kondylakis and Papadakis (2018), Bayoudhi et al. (2017)	Group Properties
Kondylakis and Papadakis (2018), Bayoudhi et al. (2017)	Ungroup Properties
Kondylakis and Papadakis (2018), Bayoudhi et al. (2017)	Pull Down Class
Kondylakis and Papadakis (2018), Bayoudhi et al. (2017)	Pull Up Class
Peixoto et al. (2016), Mahfoudh et al. (2015)	Rename Class

**Table 8** (continued)

Study	Operation
Kondylakis and Papadakis (2018), Bayouhdi et al. (2017)	Change to DataType Property
Kondylakis and Papadakis (2018), Bayouhdi et al. (2017)	Change to Object Property
Bayouhdi et al. (2017), Mahfoudh et al. (2015)	Add Equivalent Class
Bayouhdi et al. (2017), Mahfoudh et al. (2015)	Add Disjoint Class
Bayouhdi et al. (2017), Mahfoudh et al. (2015)	Remove Equivalent Class
Bayouhdi et al. (2017), Mahfoudh et al. (2015)	Remove Disjoint Class
Bayouhdi et al. (2017), Mahfoudh et al. (2015)	Add Equivalent Object Property
Bayouhdi et al. (2017), Mahfoudh et al. (2015)	Remove Equivalent Object Property
Bayouhdi et al. (2017), Mahfoudh et al. (2015)	Add Disjoint Object Property
Bayouhdi et al. (2017), Mahfoudh et al. (2015)	Remove Disjoint Object Property
Peixoto et al. (2016), Touhami et al. (2015)	Change (preferred) Label
Mahfoudh et al. (2015)	Add Individual
Mahfoudh et al. (2015)	Remove Individual
Mahfoudh et al. (2015)	Rename Individual

approach is agnostic enough to support any type of axiom. Furthermore, not all works supply extensive lists of the operations they consider.

### 3.2 RQ2: what mechanisms can be employed to measure change in knowledge?

Of the 53 retrieved search results, 9 deal with subjects related to identifying and measuring changes in knowledge over time. The works presented in Benomrane et al. (2016), Zhang et al. (2016), Cardoso et al. (2018), Duque-Ramos et al. (2016), Cano-Basave et al. (2016), Stavropoulos et al. (2019), Li et al. (2015), Algosaiibi and Melton (2016), Ziembinski (2016) identify metrics that can be employed to measure changes between ontology versions.

The retrieved results show there are two main comparison methods employed: comparing the ontology's elements to (new or existing) corpora in order to identify potential new terms, or terms that may be becoming irrelevant (Benomrane et al. 2016; Cardoso et al. 2018; Cano-Basave et al. 2016; Ziembinski 2016); and comparing different versions of the same ontology by analyzing the changes in its internal structure (Zhang et al. 2016; Duque-Ramos et al. 2016; Stavropoulos et al. 2019; Li et al. 2015; Algosaiibi and Melton 2016). Table 9 displays a set of intrinsic metrics that can be obtained by analysing the ontology's internal structure, while Table 10 shows metrics related to extrinsic properties (i.e., obtained through the analysis of data sources to which the ontology can be applied, and not to the ontology itself).

The identified metrics can be combined in different ways in order to calculate the stability of concepts between different versions of the ontology (Cardoso et al. 2018; Algosaiibi and Melton 2016).

When it comes to the methods using to identify new candidates and to analyze evolutionary trends, Natural Language Processing and Machine Learning algorithms have been popularly used to prospect text and identify potential changes for the ontology. Of the retrieved studies, they are explicitly employed in Benomrane et al. (2016), Cardoso et al.

**Table 9** Metrics applied in ontology evolution (ontology structure)

	Metric
Duque-Ramos et al. (2016), Stavropoulos et al. (2019), Li et al. (2015)	Number of direct Superclasses
Duque-Ramos et al. (2016), Li et al. (2015)	All Superclasses (distance to thing)
Cardoso et al. (2018), Duque-Ramos et al. (2016)	Number of direct Subclasses
Cardoso et al. (2018), Duque-Ramos et al. (2016), Li et al. (2015)	All Subclasses (0 if isLeaf)
Cardoso et al. (2018)	Siblings of Class
Duque-Ramos et al. (2016), Stavropoulos et al. (2019), Li et al. (2015), Algozaibi and Melton (2016)	ObjectProperties per Class
Duque-Ramos et al. (2016), Stavropoulos et al. (2019), Li et al. (2015), Algozaibi and Melton (2016)	DataTypeProperties per Class
Cardoso et al. (2018)	Region stability
Cardoso et al. (2018)	Last evolution
Cardoso et al. (2018)	Max similarity of Superclass
Cardoso et al. (2018)	Max similarity of Sibling
Cardoso et al. (2018)	Max similarity of Subclass
Duque-Ramos et al. (2016)	Number of restrictions
Cardoso et al. (2018), Duque-Ramos et al. (2016), Stavropoulos et al. (2019)	Individuals per Class
Cardoso et al. (2018), Duque-Ramos et al. (2016), Li et al. (2015), Algozaibi and Melton (2016)	Number of Classes
Li et al. (2015)	Tree impurity

**Table 10** Metrics applied in ontology evolution (vocabulary)

	Metric
Benomrane et al. (2016)	Ontologist's feedback
Cardoso et al. (2018)	Citations in corpora (concept history)
Cardoso et al. (2018)	Attribute length
Benomrane et al. (2016)	Relevance of lexical relations
Cardoso et al. (2018)	Similarity (cosine)
Cardoso et al. (2018), Ziembinski (2016)	Similarity (avg)
Cardoso et al. (2018), Stavropoulos et al. (2019), Ziembinski (2016)	Label similarity
Cano-Basave et al. (2016)	Lexical innovation
Cano-Basave et al. (2016)	Lexical adoption
Cano-Basave et al. (2016)	Linguistic progressiveness

(2018), Cano-Basave et al. (2016), Stavropoulos et al. (2019), Ziembinski (2016). A summary of the applied algorithms can be found in Table 11.

Furthermore, of the retrieved works, two (Benomrane et al. 2016; Zhang et al. 2016) employ Multi-Agent Systems to describe the ontology and potential changes.

Two works condense their metrics into matrixes in order to measure the distance between two versions of the same ontology (Duque-Ramos et al. 2016; Algozaibi and Melton 2016). Two other works do this by relying on the ontology's graph structure (Zhang et al. 2016; Li et al. 2015).

**Table 11** Algorithms applied in ontology evolution

	Algorithm
Cardoso et al. (2018)	Boosted trees
Benomrane et al. (2016)	Lexico-syntactic patterns
Cardoso et al. (2018)	Random forest
Cardoso et al. (2018)	Decision trees
Cardoso et al. (2018)	Logistic
Cardoso et al. (2018)	Support vector machine
Cano-Basave et al. (2016)	Linear discriminant analysis
Ziembinski (2016)	Clustering

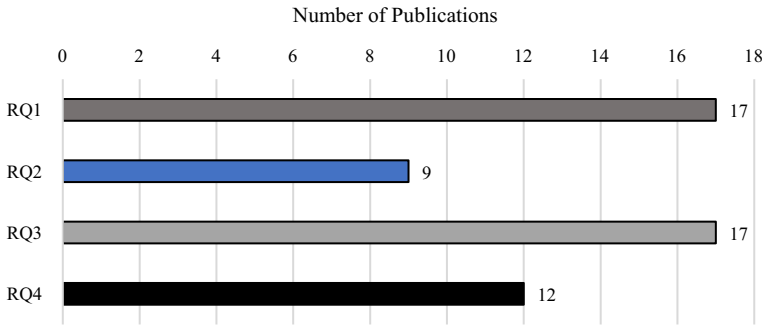
### 3.3 RQ3: how can temporal validity be applied to knowledge evolution?

The results show that there are two main approaches when it comes to representing time-constrained relationships in ontologies, namely: (1) reification, which usually assumes the form of 4D-fluents (Harbelot et al. 2015; Krieger et al. 2016; Ghorbel et al. 2019; Burek et al. 2019; Chen et al. 2018; Kessler et al. 2015; Gimenez-Garcia et al. 2017; Meditskos et al. 2016) and (2) n-quads (Calbimonte et al. 2016; Tommasini et al. 2017). In simple terms, reification describes a temporal relationship through means of a class; 4D-fluents improve on this approach by removing the cardinality constraint and allowing for any object to have any number of attributes, at any given time; n-quads, on a different approach, allows for the introduction of any number of dimensions to a triple, including time.

As for relationships that represent time but are not in themselves time-constrained, there are two main approaches: by describing qualitative (e.g. through Allen's temporal algebra) (Chen et al. 2018; Harbelot et al. 2015; Piovesan et al. 2015; Zhou et al. 2016; Zhang and Xu 2018; Batsakis et al. 2015) or quantitative (representing time points, intervals and durations) (Meditskos et al. 2016; Burek et al. 2019; Ghorbel et al. 2019; Gimenez-Garcia et al. 2017; Krieger et al. 2016; Kessler et al. 2015; Piovesan et al. 2015; Calbimonte et al. 2016; Tommasini et al. 2017; Chen et al. 2018; Batsakis et al. 2015; Li et al. 2019; Baader et al. 2018, 2015), or through partial classification by means of fuzzy ontologies (Gimenez-Garcia et al. 2017).

### 3.4 RQ4: what methods of knowledge acquisition and time representation are commonly applied in the predictive maintenance field?

In regards to knowledge acquisition, data about failures and equipment status is often provided through documental sources (e.g. maintenance reports or manuals) (Sad-Houari et al. 2019; Wang et al. 2019; Cho et al. 2019; Smoker et al. 2017; Delgoshai et al. 2017), existing systems (e.g. Enterprise Resource Planning and Manufacturing Execution Systems) (Dibowski et al. 2016; Bayar et al. 2016; Cho et al. 2019; Delgoshai et al. 2017), sensors (Dibowski et al. 2016; Saeed et al. 2019; Cho et al. 2019; Klusch et al. 2015; Steinegger et al. 2017; Ferrari et al. 2017), or combinations of these. In these cases, the representation of time is done exclusively through time points (Cho et al. 2019; Delgoshai et al. 2017; Saeed et al. 2019; Klusch et al. 2015; Ferrari et al. 2017), and no approaches explicitly using time-constrained relationships have been found.



**Fig. 2** Number of publications for each research question

Finally, half of all retrieved studies propose or show potential combination of ontologies with other methods in order to achieve better results in the identification of events and potential failures, such as machine learning algorithms (Cho et al. 2019; Smoker et al. 2017; Steinegger et al. 2017), multi-agent systems (Steinegger et al. 2017), rules (Sad-Houari et al. 2019; Dibowski et al. 2016; Bayar et al. 2016), case-based reasoning (Ansari et al. 2019) and probabilistic models (Klusich et al. 2015; Ferrari et al. 2017).

## 4 Discussion

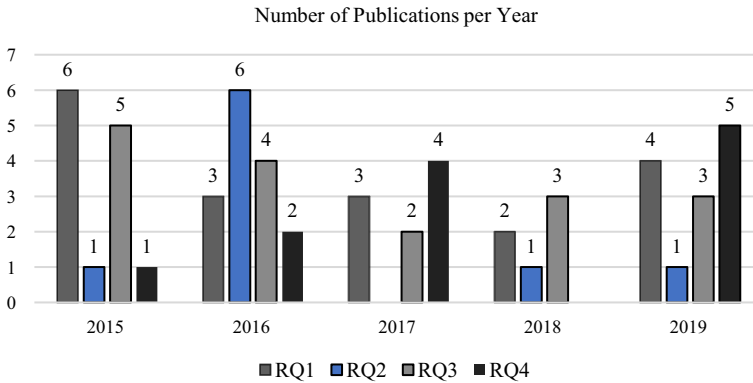
This section will take a deeper look at the answers to the research questions, provide a general discussion of the different works retrieved and attempt to identify possible new directions for the fields of study of ontology evolution and time representation in predictive maintenance.

Below, Fig. 2 shows the number of publications that answer each of the research questions:

When it comes to the evolution of knowledge, for the past 5 years, we can say that methods of representation (RQ1) and the representation of time using ontologies (RQ3) are the most popular research points (both with 32% of the retrieved studies focusing in some way on those topics). Applications of ontology that employ temporal reasoning (RQ4) are slightly less popular (23%). Finally, when it comes to proposing and establishing metrics to identify changes in knowledge over time (RQ2), we can say that this topic has gotten the least attention (17%). To get a better look at how the popularity of the different topics evolved over time, we can split the number of publications that answer each RQ per year.

For RQ1, we can see in Fig. 3 that there has been a decrease in relative popularity over the years, with the number of publications reaching the lowest point in 2018. However, this trend appears to be changing in 2019, in which the number of published studies is greater than in the 3 previous years, indicating a potential renewed interest in the topic.

As was seen in Fig. 2, RQ2 got the least attention in the last five years, which makes the sample harder to evaluate. Figure 2 shows us that interest in the topic peaked in 2016, with no studies being published in the following year. However, for the past two years, the number has risen back to its 2015 values, showing a resurfacing in interest. As the second most popular topic, we can see that the topics in RQ3 have maintained their popularity relatively steady for the last five years, but with a slightly downwards tendency. The worst year was,



**Fig. 3** Number of studies per year per RQ

however, 2017, with one more publication per year in 2018 and remaining steady since. Finally, as for the trends regarding RQ4, for the first three years considered in this study, there was a significant increase in interest in ontologies for predictive maintenance with a temporal component. This trend took a great shift in 2018, in which no new studies were found. However, this is largely compensated by the drastic increase in studies in 2019, the year in which most studies on the topic have been published.

#### 4.1 RQ1: what methods have been used to represent the evolution of knowledge over time?

A third of the works retrieved explore topics related to the representation of the evolution of knowledge over time. In regards to approaches to ontology evolution, approximately 22% of the publications advise for change-based approaches, and 5% for the naïve approach. The remainder of the studies do not specify which of the two they could be applied to, although works that propose representing specific evolutionary actions (55%) suggest applications in change-based scenarios.

On the topic of ontology versioning, the authors in Bayoudhi et al. (2019) propose a strategy for efficient retrieval of knowledge from past versions of an ontology, while keeping consistency in the evolution events. While many previous works have focused on the idea of consistency checking a posteriori, here it is assessed a priori with the aim of reducing both processing time and storage space.

In Grandi (2016), the author proposes representing and storing the different versions of an ontology in a relational database in order to make the versioning easily manipulatable through SQL queries. How the evolution itself occurs or how it is identified is not in the scope of this study: the focus is exclusively on the description of change and versioning. The process starts with the identification of the different types of changes, i.e., primitive operations that can be applied when evolving an ontology. Each ontology version is identified with a unique timestamp and any version of it can be reconstructed at any given time using SQL queries. Retroactive application of a given operation is also possible through manipulation of its timestamp. The work presented in Shaban-Nejad and Haarslev (2015) follows a similar premise, with a particular focus in retaining decidability and keeping consistency in interactions with other existing ontologies. To do so, they generate sub-graphs



for the ontologies and assess the impact of a set of changes upon them; as such, it possible to inform the other ontologies of which changes they must undergo in order to remain consistent with changes in their neighbourhood. On the other hand, Bayoudhi et al. (2017) attempts to bridge the gap between change-based and timestamp-based approaches by proposing a hybrid approach that also takes benefit from relational database technologies, in which only the reference version and the most recent version of the ontology are effectively materialized.

EvoRDF (Kondylakis and Papadakis 2018) is a tool devised to explore the evolution of ontologies over time, showing what changed between any two versions. To do so, they present an ontology based on the Open Provenance Model to describe the types of changes that can take place that allow for description of how and when the change occurred. Answering *why* certain changes have happened is also explored in this work, by providing the user with the evolutionary path of a given concept or property, i.e., which change operations have been made to the concept. In the same vein, Kozierkiewicz and Pietranik (2019) propose a formal framework for ontology evolution which is able to answer queries such as which changes occurred and when. Ontology versions are considered a timeseries, and for any pair of them, the framework can identify which changes were applied in terms of classes, relationships, properties and instances.

In Osborne and Motta (2018), the authors discuss the possibility of introduction of parameterizable requirements for the new ontology version to meet. The solution is presented not as a means to identify what changes should occur, but as a guiding hand for the user in selecting which changes from a possible change set would have the least impact in the performance of the applications that use the ontology. The parameterization of this algorithm is particularly rich, allowing for requirements as vast as the number of concepts of the ontology (or on a specific branch of the ontology) or favouring newer over older concepts (and vice-versa). The representation of time becomes an important factor to consider in this last part, as it influences how relevant a particular class may be. In Benomrane et al. (2016), the authors describe a multi-agent system that executes evolutionary actions in response to an ontologist's needs, and use the ontologist's feedback to improve the system's performance. While this study briefly touches the subject of identification of candidates, its main focus is on implementing change and incorporating the ontologist's feedback.

In a different approach to ontology evolution, the authors in Peixoto et al. (2016) consider using an ontology-described classification model which is applied to a stream of unstructured textual data in a Big Data context. Considering that concept drift is bound to happen in such streams, with older concepts being slowly replaced with new ones, the classification model must be able to identify and adapt to such change. An ontology is proposed to describe the adaptative process by describing the possibilities of feature evolution, concept evolution and concept drift and how these can identify change and propose possible modifications to another ontology, which is based upon existing literature on ontology evolution. Upon change identification, a request for change is made, the consistency of the ontology upon the change is checked, and the modification is applied. When inconsistencies arise, the algorithm attempts to resolve them through additional changes, although these must ultimately be addressed by a human specialist.

The collaborative aspect of ontology evolution is explored in Mihindikulasooriya et al. (2017). The authors focus mainly on structural change within the ontology, i.e., the addition, deletion, splitting or merging of concepts, properties, restrictions and axioms. The study focuses mainly on analysing existing and popular collaborative ontologies and their evolution over time in order to assess trends or possible guidelines for such types of ontologies. Furthermore, it suggests that the scope and complexity of such ontologies effectively

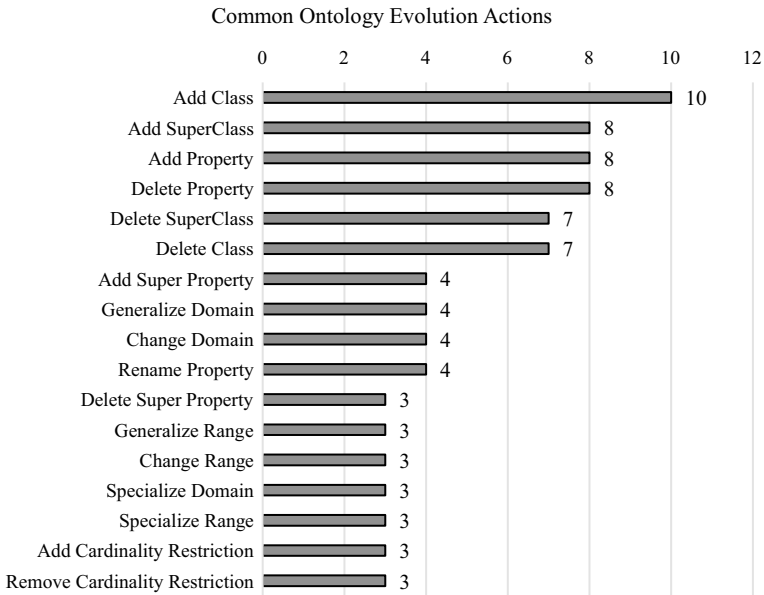
makes them harder to use, and more prone to ambiguity over time; to counteract such tendency, the authors propose the usage of annotations for editors, but do not identify how to employ them or how they could be used.

There are several approaches to maintaining consistency between ontology versions, i.e., to make sure that a specific change to an ontology will not result in contradictory axioms. In this vein, the work presented in Mahfoudh et al. (2015) takes advantage of an algebraic approach called Simple PushOut, in which possible modifications to the ontology are described through rules. The application of those rules should then result in either the addition or rejection of the new axioms, meaning that the consistency check is effectively done before any changes to the ontology are performed. The work presented in Wang et al. (2015) presents the consistency problem from a different point of view: when new assertions are added to the ABox that contradict the TBox, the TBox must drop axioms in order to maintain consistency. This solution is particularly suited for automatic evolution processes and relies on a new TBox contraction operator that allows modification of existing axioms whenever possible, instead of their outright removal.

Another work dedicated to inconsistency analysis under ontology evolution is presented in Gaye et al. (2015). To do this, the authors propose decomposing an ontology in sub-graphs (here described as “communities”) and exploring the ripple effects of change within a community and between adjacent communities. To measure inconsistency, the authors propose calculating the probability of dependence between entities and assertions and describing simple basic operations that can be executed on the ontology. For each operation, pre-conditions, invariants and post-conditions are settled with their respective inconsistency measuring formula. Similarly, coherence constrains, operations and pre and post-conditions are also explored in Touhami et al. (2015). A set of evolutionary strategies for inconsistency scenarios are also presented.

The impact of ontology evolution in existing rules is approached in Sad-Houari et al. (2019). A change in an ontology may render deployed rules in engines or Business Rules Management Systems unusable, or even make these produce unexpected outcomes when triggered. To guarantee that any step of the evolution process does not lead to unintended consequences, each change is executed, and affected rules are tested for consistency, namely contradiction, rules that become inapplicable or invalid, domain violations and redundancy. If any of these inconsistencies arise, they must be solved before committing the change through a negotiation process that takes place between all experts and using Contract Net Protocol, which may result in either the rejection of the change or in updating all affected rules.

The evolution of knowledge is also a factor when it comes to the querying mechanisms. In Tsalapati et al. (2017), the authors present a method to effectively rewrite queries when the ontology evolves; in this scenario, the queries are deemed static, while the ontologies are dynamic. Unlike previously existing algorithms, the method presented does not compute new queries from scratch upon the introduction of new axioms; instead, it will attempt to expand existing queries by discovering how these are affected by the change, whether through new inferences or renaming of variables, while also removing redundant clauses. When the ontology is contracted, the algorithm identifies the minimal subset of clauses that are necessary to answer a given query under change, removing those considered unnecessary while attempting to maintain as much of the previous computation as possible. The authors warn, however, that different reasoning engines tend to produce different results and, while the algorithm they present is adaptable to such variability, it can also end up with distinct results or computational times. Different optimization methods are thus presented for expansion and contraction



**Fig. 4** Commonly described ontology evolution actions

scenarios. On the same topic, the work presented in Zheleznyakov et al. (2019) discusses the expansion and contraction of DL-Lite knowledge bases and proposes a novel formula-based approach that maintains the DL-lite constraints while also not being computationally overintensive.

When it comes to the specific evolutionary actions proposed by the 15 studies chosen to answer RQ1, some patterns are noticeable. Figure 4, below, presents us the most commonly described ontology evolution actions in these studies. *N.b.*: for clarity reasons, only actions that are described in more than two studies have been considered.

Adding classes, hierarchical relations and properties seem to be the main concern when it comes to describing evolutionary actions. This makes sense from the perspective that, in many cases, light-weight ontologies can effectively function as taxonomies, which can explain the focus on hierarchical relationships. This is not the case of heavy-weight ontologies, which are more complex and make a more extensive use of the Description Logics capabilities, and may find these tools lacking in the description of their evolution processes; particularly in regards to dealing with possible inconsistencies. Graph-based approaches seem, for the most part, to ignore this potential hindrance and may find little applicability in such scenarios. It is worth noting that disjoint relationships (either for classes or properties) are only considered in two of the fifteen studies. This is also true for moving classes up or down the ontology tree or the many possibilities that exist when editing properties. The consistency check implications of different types of relationships portrayed in heavy-weight ontologies may, however, be much more computationally intensive. Nonetheless, since they can be represented in OWL, they too may be subject to evolutionary processes and therefore need addressing. With some works already representing change through means of complex ontologies (Kondylakis and Papadakis 2018; Peixoto et al. 2016), it seems that these possibilities may be more commonly addressed in the future. Furthermore, using ontologies for the semantic description of change could come

with the capability of using time-constrained relationships, i.e., to have changes be applicable through particular periods of time.

#### 4.2 RQ2: what mechanisms can be employed to measure change in knowledge?

Approximately a sixth of the works focus on measuring changes in knowledge. Of these, 44% are focused in working from a single ontology version and an incoming source of new terms (such as streams or new documents), and 66% focus instead of identifying and analyzing changes between two existing ontology versions.

In Benomrane et al. (2016), the authors propose a multi-agent system to represent both the process of ontology evolution and the ontologist's feedback. Incoming documents introduce new candidate terms that may or may not be included in a new ontology version, with each candidate term being associated with an agent with its own relevance value, and only the most relevant candidates are presented to the ontology. This process pertains to make the evolution process easy for the ontologist by allowing agents to cooperate in finding the appropriate locations for the candidate terms they represent in the ontology. The ontologist's feedback on the candidates—whether they are approved or not, if they need to be revised, among others—allows the agents to adapt their behaviour for future proposals. In a similar approach also based in multi-agent systems, the authors in Zhang et al. (2016) treat the ontology as an autonomous entity responsible for its own evolution, supplying the tools to describe states between operational changes. Each change is then described as a number of that must take place and be agreed upon for the change to become effective and their corresponding logical operations.

In Cardoso et al. (2018), the authors propose a stochastic model that implements machine learning techniques in order to identify which sections of the ontology may require updating. While the work is mainly focused on the machine learning models applied and the selected features, it proposes interesting metrics to evaluate the likelihood of a given concept to evolve, and how the change can be represented and accounted for. Each concept is described according to three dimensions: (1) its structural characteristics, i.e. the number of attributes, siblings, super and sub concepts; (2) temporal characteristics, i.e. the history of the concept's evolution and (3) the relational characteristics, i.e., additional relations identified for the concept via external sources such as the Web. Furthermore, four types of evolution actions are identified: Extension, Removal, Description Change and Move. The study of the stability of concepts and their likelihood to change in the future is done by analysing the evolution of these variables over time with machine learning models and the revision most likely to be required.

Metrics, both for the identification of change and to measure the degree of change between ontology versions, are particularly important for ontology evolution analysis. The OQuaRe Framework (Duque-Ramos et al. 2016) presents a set of qualitative metrics for ontology evolution processes. A quality model assigns value to the different ontology's characteristics—a wide set, which includes, among others, quality requirements such as reliability and maintainability. The variation of these metrics between ontology versions allows the authors to make further judgements about the quality of the change, the quality of the resulting ontology, and to describe evolutionary trends.

In Cano-Basave et al. (2016), the authors forward a new concept related the identification of incoming changes to an ontology, which they call Ontology Forecasting. The authors propose using algorithms established in the Natural Language Processing field to identify additional concepts in new corpora that may be used for evolving existing

ontologies as the domain complexity goes over time. They do this by balancing two metrics: lexical innovation and lexical adoption, which identify the emergence of new topics in a new dataset, and the continuity in terms between datasets, respectively. A third metric is proposed, that of linguistic progressiveness of a domain, a ratio between the two previous metrics. This metric is employed in a new model, the Semantic Innovation Forecast Model, which aims to identify which concepts are more likely to be introduced to the ontology by analysing changes in the domain.

In Stavropoulos et al. (2019), the issue of semantic drift is tackled and how to identify and implement that change in an existing ontology. Semantic drift is here presented as the evolution of the meaning of a given concept to the community over time and which must be crystallized into the ontology in some form. The authors identify two approaches for measuring change between two versions of the same ontology based on the ability to assess the identity of a given concept: identity-based approach, in which there is a clear, one-to-one match between all concepts in both ontologies, and a morphing-based approach, in which each concept evolves into new, sometimes similar concepts, but whose identity cannot be ascertained between different versions. The then authors propose a third, hybrid approach which first assumes all changes to be morphing and then looking for the most similar concept in the most recent version of the ontology, assuming that one as their identity. For any concept, three aspects are considered when calculating the change: the labels of the concept, i.e. their textual description, the intensional aspect, i.e. its usage in the domain and relationships to other concepts, and the extensional aspect, i.e. the instances.

Metrics proposed for identifying change in Stavropoulos et al. (2019) are the string similarity between labels, and Jaccard similarity between relationships for both the intensional and extensional aspects, with the final metric being an average of the three. By using their proposed hybrid-approach, the authors then suggest using this metric over several change iterations of the ontology in order to analyse the stability of concepts over time, identifying and measuring change. Further metrics designed for properties specifically are described in Li et al. (2015). These rely on the graph-like structure of ontologies and can be used to measure different versions of the same ontology in regard to the stability of its properties and concepts over time.

Ontology complexity tends to increase as the ontology evolves, making the identification and application of changes more difficult over time. In Algosaibi and Melton (2016), the authors propose that each concept can be considered a point in space, with the coordinates referring to the three identified metrics: the number of properties, the number of subclasses and the number of superclasses. By calculating the distance between two points, they authors estimate the complexity of the change and how costly its application would be.

When it comes to analysing data from streams, there's a strong assumption that this data is either structured (e.g. RDF) or, if unstructured, at least textual in nature; so much that NLP algorithms can be applied to it. A lot of applications of streams and machine learning algorithms use sensor data (e.g. Dibowski et al. 2016; Klusch et al. 2015; Steinegger et al. 2017; Ferrari et al. 2017; Cho et al. 2019), which is a largely unexplored field when it comes to potential triggering of ontology evolution, to the best of our knowledge.

### 4.3 RQ3: how can temporal validity be applied to knowledge evolution?

33% of the studies explores ways to represent time using ontologies. Of these, 47% propose reified-based approaches for relationships constrained in time, and 11%

propose adding a new dimension to the data through n-quads. Furthermore, 35% detail usage of qualitative temporal relationships between concepts, and 82% detail usage of quantitative relationships. 17% of the studies propose a combination of the two approaches, particularly for usage in query answering. Additionally, 5% of the works consider using fuzzy ontologies in combination with the previous approaches.

In Meditskos et al. (2016), temporal events are discrete and described as Situations, with specific start and end time points. The work described in Harbelot et al. (2015) represents the evolution of entities over time through the implementation of Time Slices. Time Slices are comprised by four dimensions, namely Identity, Semantic, Spatial and Temporal. Any concepts in the ontology should therefore be described through time slices, meaning that definitions and relationships hold true for specific intervals (and for specific locations). The temporal dimension itself is represented through either Temporal Points (discrete moments in time) or Intervals (sets of Temporal Points). In a similar approach, Burek et al. (2019) discusses different methods of modelling time, such as: indexing time points, implementation of 4D-fluents both vertically and horizontally. Ultimately the authors decide for a combination of both horizontal 4D-fluents and discrete time points representation, which is similarly used in Batsakis et al. (2015), which also proposes combining Allen's temporal algebra and fixed time points using interval duration calculus relations. In Gimenez-Garcia et al. (2017), in order to reason with precise and imprecise temporal information, crisp and fuzzy time representations are employed to extend both the representative capabilities of the 4D-fluents and the inferences possible through the application of Allen's temporal algebra. Zhang and Xu (2018) uses Allen's temporal algebra to define SPARQL operators what can be used to discover hidden temporal relationships in datasets that are described through time points. Without applying Allen's algebra, the work in Meditskos et al. (2016) presents properties for the representation of specific time points contained within actions or intervals and (Li et al. 2019; Baader et al. 2015, 2018) approach the representation of time and timed-relations for query answering purposes.

Other approaches based on reification can be found in Krieger et al. (2016), Kessler et al. (2015), Piovesan et al. (2015), which use Time Ontology to describe time points and intervals, and in Chen et al. (2018). In this last study, the authors apply the Time Event Ontology, which describes time points, intervals, durations and timed relationships, among others. The timed relationships, in particular, are described using Allen's temporal algebra. Furthermore, durations can be described not just through the difference between two time points, but also by specifying its length in terms of hours, minutes and seconds. For temporal uncertainty, temporal relations can use an approximation parameter.

Calbimonte et al. (2016) uses a n-quad based approach in which each RDF triple is assigned a timestamp that can be used to establish sliding windows in RDF streams. Also in the field of stream processing, the work presented in Tommasini et al. (2017) introduces time-aware operators for point-based time semantics that can be employed to establish temporal dependencies, filter and merge events.

While the reification approach to time and timed-constrained relationships is by far the most popular, to the best of our knowledge they find no intersection with the approaches described in RQ1. When it comes to representing time in this scenario, the approaches that are explicit about their approach fall under quantitative representations based on time points.

#### 4.4 RQ4: what methods of knowledge acquisition and time representation are commonly applied in the predictive maintenance field?

Of the 12 retrieved studies in the field of predictive maintenance, only one (8%) addresses the topic of ontology evolution. In the other 92% of the studies, ontologies are used for fault detection and diagnosis (50%), fault propagation (16%) and otherwise for data representation and categorization. 50% of the studies apply ontologies to the description of data acquired from sensors via streams.

In Sad-Houari et al. (2019) the authors present an ontology and a set of maintenance rules that are triggered according to the risk of failure or equipment damage. While this study is focused in keeping the ontology and existing rules consistent, it is an interesting insight into the direction future studies in the evolution of ontologies for predictive maintenance may take. Predictive maintenance, as the name suggests, has a strong temporal component to it. The use of ontologies in maintenance has been increasing over the years, with applications for representing the domain, semantically annotating data (Wang et al. 2019; Smoker et al. 2017) and possible applications for explainability of different scenarios. When the temporal dimension is considered in these ontologies, it is only in a quantitative fashion, through means of time points (Cho et al. 2019; Delgoshaei et al. 2017; Saeed et al. 2019; Klusch et al. 2015; Ferrari et al. 2017), with no time-constrained relationships—which makes for a very static way of representing a domain that is so inherently dynamic and time-dependant. Qualitative time relations such as those provided by Allen's temporal algebra have also not been applied to the predictive maintenance field to the best of our knowledge. While there is potential for the application of temporal reasoning to the predictive maintenance field, the time representation capabilities in the ontologies presented are underdeveloped and underutilized for this particular purpose. In all but one study (Saeed et al. 2019), this is compensated by incorporating the ontology in a framework that makes use of other approaches and tools (Ansari et al. 2019; Klusch et al. 2015; Steinegger et al. 2017; Dibowski et al. 2016; Smoker et al. 2017; Ferrari et al. 2017; Sad-Houari et al. 2019; Cho et al. 2019; Bayar et al. 2016). Of these, rule-based systems and machine learning approaches are most popularly employed, accounting for 66% of all solutions.

Of the retrieved works, 50% are particularly related to fault detection and diagnosis (FDD) based on ontologies (Delgoshaei et al. 2017; Klusch et al. 2015; Steinegger et al. 2017; Ferrari et al. 2017). Sensor streams are semantically annotated with ontologies for fault diagnosis in Klusch et al. (2015) and Steinegger et al. (2017). In Klusch et al. (2015), a domain ontology describes sensors, components, symptoms and faults and is applied both offline, to historical data, and online, to real-time data, using C-SPARQL and Bayesian Networks. The semantics of the ontology further help with diagnosis tasks by establishing connections between faulty components. Steinegger et al. (2017) employs autonomous agents to the same end. In Ferrari et al. (2017), the authors propose a generic ontology for FDD, coupled with other contextual ontologies such as the Time Ontology and the Space Ontology, and possibly domain-dependent ontologies. This approach makes for a domain-independent framework for FDD. Detection is performed through the comparison of real versus expected values and diagnosis is inferred by analysing evidence data. Another FDD framework is proposed in Steinegger et al. (2017), which integrates a number of modular ontologies with a higher-level reference ontology. In this scenario, ontologies are applied for root cause analysis and alarm management. In Ferrari et al. (2017), the ontology is complemented with probabilistic models and uncertainty propagation algorithms. An ontology for fault diagnosis from sensor data is presented in Saeed et al. (2019). It makes

use of time points and intervals to describe observations and failure symptoms to assess the effective time of failure. In Ansari et al. (2019), both ontologies and case-based reasoning approaches are applied to identify when and how faults in equipment will occur.

The work presented in Dibowski et al. (2016) uses ontologies to describe interconnections between locations, equipment and others for fault propagation simulation purposes. The fault propagation mechanisms themselves are described using SWRL rules, which are triggered when certain variables enter fault states. Fault propagation effects are further separated into inevitable and possible consequences. In Bayar et al. (2016), the authors propose an ontology design based on immune systems to be applied to fault prediction in manufacturing environments. Failure statuses, or disruptions, can be identified and propagation paths inferred, with their expected delays in production through SWLR rules.

In Cho et al. (2019), the ontology provides a semantic layer to real-time data captured through various sources, which can be used to enrich machine learning algorithms' inputs and generate reports about equipment status. Results of these algorithms can be used to enrich the ontology with new instances and improve its performance over time. Similarly, the authors in Smoker et al. (2017) use ontologies to support the predictive algorithms by classifying maintenance data.

A lot of the studies already propose using ontologies to describe information captured through sensors (Saeed et al. 2019; Klusch et al. 2015; Steinegger et al. 2017; Dibowski et al. 2016; Ferrari et al. 2017; Cho et al. 2019), which is, in most cases, is distributed in real-time through streams. There is a potential here to go beyond annotation of streams with timestamps and into more expressive and complex temporal relationships, particularly for time constraints, as suggested by Calbimonte et al. (2016), Tommasini et al. (2017) and Klusch et al. (2015).

## 5 Conclusions and future work

This work presents a systematic review in the topic of time representation and ontology evolution in the predictive maintenance field. Of the 702 studies retrieved from the different data sources, 140 were approved for full-text reading, and 53 have been included and reviewed in this work. These studies were distributed through four research questions regarding methods of representation of the evolution of knowledge, mechanisms employed in the identification and measure of change, representation of time and time-constrained relationships and, finally, how all these topics are applied to the field of predictive maintenance. Analysing the distribution of the results through the research questions it was possible to assess that both evolution of knowledge and time representation are very well represented, with each making up 32% of all retrieved studies, while the representation of time-constrained relationships was the least popular topic in the last five years, effectively having no publications in one of the years considered in this work. It is important to note however, that while there was an initial decrease in the number of studies over time in all the observed fields, there was a resurfacing in interest in ontologies for predictive maintenance in the last year, while the other fields have recovered some of the interest and maintained it for the last two years.

On the topic of representation of the evolution of knowledge, the different approaches to ontology evolution and versioning have been presented. How to represent specific evolutionary actions and how to account for their impact in the consistency of the ontology and further necessary actions was also explored, with some works even going as far as to



represent these using ontologies. A summary of the most commonly considered evolutionary actions was presented and briefly discussed.

As for identifying and measuring change, we have analysed the two different ways this can be done, either by comparison of new data with an existing ontology, or by comparing two existing ontology versions and discussing the changes. Several metrics were proposed by the studies retrieved, and these have been discriminated according to whether they pertain to the internal structure of the ontology, or to new vocabulary in the generation of candidate evolutionary actions. A quick survey of the algorithms used to employ these metrics is also presented.

Regarding studies in the representation of relationships that evolve over time, these divide twofold: in time-restrained relationships, we find reification (i.e., declaring a new class that will actually represent a relationship between two other classes with a time dimension) to be the most popular approach that does not include designing new reasoning mechanisms. Furthermore, we take a look at relationships that represent time in quantitative and qualitative ways, and possible ways to combine the two.

Unfortunately, the intersection of all these areas finds only one application in the field of predictive maintenance. While ontologies find application in the predictive maintenance field, they make little use of their time representation capabilities and are more often than not applied for interoperability purposes. Furthermore, the ontologies themselves serve mostly as a complement to other tools in order to generate predictions, allowing for an easier understanding of the domain and uncovering implied connections in the data that can be harvested by different predictive algorithms. This, however, may represent a research opportunity in the future, particularly given how dynamic this domain can be, with the introduction or removal of new elements, and how time-dependent it is. Not only must the conclusions be reached in useful time, but they must also consider the temporal aspect of the data they come from and the potential of the domain to evolve over time.

As for future work, we will use the results obtained in this systematic review to devise solutions that make use of ontology's potential for time representation and evolution in the field of predictive maintenance.

**Acknowledgements** This work was supported by national funds through FCT—Fundação para a Ciência e Tecnologia through project UIDB/00760/2020 and Ph.D. scholarship with reference SFRH/BD/147386/2019.

**Author contributions** AC: Conceptualization, Investigation, Visualization, Writing—Original draft preparation, Writing—Review & Editing. JC: Supervision. GM: Supervision, Validation, Writing—Reviewing and Editing.

**Funding** Alda Canito is supported by national funds through FCT—Fundação para a Ciência e Tecnologia through project UIDB/00760/2020 and Ph.D scholarship with reference SFRH/BD/147386/2019.

## Declarations

**Conflict of interest** The authors declare that they have no conflict of interest.

## References

Algozaibi AA, Melton Jr. AC (2016) Three dimensions ontology modification matrix. In: Proceedings of 2016 2nd international conference on information management (ICIM2016).

- Ansari F, Glawar R, Nemeth T (2019) PriMa: a prescriptive maintenance model for cyber-physical production systems. *Int J Comput Integr Manuf* 32(4–5, SI):482–503. <https://doi.org/10.1080/0951192X.2019.1571236>
- Baader F, Borgwardt S, Lippmann M (2015) Temporal conjunctive queries in expressive description logics with transitive roles. In: Pfahringer B, Renz J (eds) *Ai 2015: advances in artificial intelligence*, vol 9457, pp 21–33. [https://doi.org/10.1007/978-3-319-26350-2\\_3](https://doi.org/10.1007/978-3-319-26350-2_3)
- Baader F, Borgwardt S, Forkel W (2018) Patient selection for clinical trials using temporalized ontology-mediated query answering. In: *Companion proceedings of the the web conference 2018*, pp 1069–1074. <https://doi.org/10.1145/3184558.3191538>
- Batsakis S, Antoniou G, Tachmazidis I (2015) Integrated representation of temporal intervals and durations for the semantic web. In: Bassiliades N, Ivanovic M, KonPopovska M, Manolopoulos Y, Palpanas T, Trajcevski G, Vakali A (eds) *New trends in database and information systems II*, vol 312, pp 147–158. [https://doi.org/10.1007/978-3-319-10518-5\\_12](https://doi.org/10.1007/978-3-319-10518-5_12)
- Bayar N, Darmoul S, Hajri-Gabouj S, Pierrelval H (2016) Using immune designed ontologies to monitor disruptions in manufacturing systems. *Comput Ind* 81(SI):67–81. <https://doi.org/10.1016/j.compind.2015.09.004>
- Bayouhdi L, Sassi N, Jaziri W (2017) A hybrid storage strategy to manage the evolution of an OWL 2 DL domain ontology. In: ZanniMerk C, Frydman C, Toro C, Hicks Y, Howlett RJ, Jain LC (eds) *Knowledge-based and intelligent information & engineering systems*, vol 112, pp 574–583. <https://doi.org/10.1016/j.procs.2017.08.170>
- Bayouhdi L, Sassi N, Jaziri W (2019) Efficient management and storage of a multiversion OWL 2 DL domain ontology. *Expert Syst*. <https://doi.org/10.1111/exsy.12355>
- Benomrane S, Sellami Z, Ben Ayed M (2016) Evolving ontologies using an adaptive multi-agent system based on ontologist-feedback. In: *2016 IEEE tenth international conference on research challenges in information science (RCIS)*, pp 1–10. <https://doi.org/10.1109/RCIS.2016.7549292>
- Burek P, Scherf N, Herre H (2019) Ontology patterns for the representation of quality changes of cells in time. *J Biomed Semant*. <https://doi.org/10.1186/s13326-019-0206-4>
- Calbimonte J-P, Mora J, Corcho O (2016) Query rewriting in RDF stream processing. In: *Proceedings of the 13th international conference on the semantic web. Latest advances and new domains*, vol 9678, pp 486–502. [https://doi.org/10.1007/978-3-319-34129-3\\_30](https://doi.org/10.1007/978-3-319-34129-3_30)
- Cano-Basave AE, Osborne F, Salatino AA (2016) Ontology forecasting in scientific literature: semantic concepts prediction based on innovation-adoption priors. In: Blomqvist E, Ciancarini P, Poggi F, Vitali F (eds) *Knowledge engineering and knowledge management, EKAW 2016*, vol 10024, pp 51–67. [https://doi.org/10.1007/978-3-319-49004-5\\_4](https://doi.org/10.1007/978-3-319-49004-5_4)
- Cardoso SD, Pruski C, Da Silveira M (2018) Supporting biomedical ontology evolution by identifying outdated concepts and the required type of change. *J Biomed Inform* 87:1–11. <https://doi.org/10.1016/j.jbi.2018.08.013>
- Chen HW, Du J, Song H-Y, Liu X, Jiang G, Tao C (2018) Representation of time-relevant common data elements in the cancer data standards repository: statistical evaluation of an ontological approach. *JMIR Med Inform*. <https://doi.org/10.2196/medinform.8175>
- Cho S, May G, Kiritsis D (2019) A semantic-driven approach for industry 4.0. In: *2019 15th international conference on distributed computing in sensor systems (DCOSS)*, pp 347–354. <https://doi.org/10.1109/DCOSS.2019.00076>
- Delgoshaei P, Austin MA, Veronica D (2017) Semantic models and rule-based reasoning for fault detection and diagnostics: applications in heating, ventilating and air conditioning systems. In: Austin M, Snow A, Mourlin F (eds) *Twelfth international conference on systems (ICONS 2017)*, pp 48–53
- Dibowski H, Holub O, Rojíček J (2016) Knowledge-based fault propagation in building automation systems. In: *2016 international conference on systems informatics, modelling and simulation (SIMS)*, pp 124–132. <https://doi.org/10.1109/SIMS.2016.22>
- Duque-Ramos A, Quesada-Martinez M, Iniesta-Moreno M, Fernandez-Breis JT, Stevens R (2016) Supporting the analysis of ontology evolution processes through the combination of static and dynamic scaling functions in OQuaRE. *J Biomed Semant*. <https://doi.org/10.1186/s13326-016-0091-z>
- Ferrari R, Dibowski H, Baldi S (2017) A message passing algorithm for automatic synthesis of probabilistic fault detectors from building automation ontologies. *IFAC PapersOnline* 50(1):4184–4190. <https://doi.org/10.1016/j.ifacol.2017.08.809>
- Gaye M, Sall O, Bousso M, Lo M (2015) Measuring inconsistencies propagation from change operation based on ontology partitioning. In: *2015 11th international conference on signal-image technology internet-based systems (SITIS)*, pp 178–184. <https://doi.org/10.1109/SITIS.2015.18>

- Ghorbel F, Hamdi F, Metais E (2019) Ontology-based representation and reasoning about precise and imprecise time intervals. In: 2019 IEEE international conference on fuzzy systems (FUZZ-IEEE), pp 1–8. <https://doi.org/10.1109/FUZZ-IEEE.2019.8859019>
- Gimenez-Garcia JM, Zimmermann A, Maret P (2017) Representing contextual information as fluents. In: Ciancarini P, Poggi F, Horridge M, Zhao J, Groza T, SuarezFigueroa MC, DAquin M, Presutti V (eds) Knowledge engineering and knowledge management, vol 10180, pp 119–122. [https://doi.org/10.1007/978-3-319-58694-6\\_13](https://doi.org/10.1007/978-3-319-58694-6_13)
- Grandi F (2016) Dynamic class hierarchy management for multi-version ontology-based personalization. *J Comput Syst Sci* 82(1):69–90. <https://doi.org/10.1016/j.jcss.2015.06.001>
- Harbelot B, Arenas H, Cruz C (2015) LC3: a spatio-temporal and semantic model for knowledge discovery from geospatial datasets. *J Web Semant* 35(1):3–24. <https://doi.org/10.1016/j.websem.2015.10.001>
- Kessler C, Farmer CJQ, Keunitedler C, Farmer CJQ (2015) Querying and integrating spatial-temporal information on the web of data via time geography. *Web Semant* 35(P1):25–34. <https://doi.org/10.1016/j.websem.2015.09.005>
- Kitchenham B (2004) Procedures for performing systematic reviews. Keele University, Keele, p 33
- Klusck M, Meshram A, Schuetze A, Helwig N (2015) ICM-hydraulic: semantics-empowered condition monitoring of hydraulic machines. In: Proceedings of the 11th international conference on semantic systems, pp 81–88. <https://doi.org/10.1145/2814864.2814865>
- Kondylakis H, Papadakis N (2018) EvoRDF: evolving the exploration of ontology evolution. *Knowl Eng Rev*. <https://doi.org/10.1017/S0269888918000140>
- Kozierkiewicz A, Pietranik M (2019) A formal framework for the ontology evolution. In: Nguyen NT, Gaol FL, Hong TP, Trawinski B (eds) Intelligent information and database systems, ACIIDS 2019, PT I, vol 11431, pp 16–27. [https://doi.org/10.1007/978-3-030-14799-0\\_2](https://doi.org/10.1007/978-3-030-14799-0_2)
- Krieger H-U, Peters R, Kiefer B, van Bekkum MA, Kaptein F, Neerinx MA (2016) The federated ontology of the PAL project interfacing ontologies and integrating time-dependent data. In: Fred A, Dietz J, Aveiro D, Liu K, Bernardino J, Filipe J (eds) KEOD: proceedings of the 8th international joint conference on knowledge discovery, knowledge engineering and knowledge management, vol 2, pp 67–73. <https://doi.org/10.5220/0006015900670073>
- Li Z, Feng Z, Wang X, Li Y, Rao G (2015) Analyzing the evolution of ontology versioning using metrics. In: 2015 12th web information system and application conference (WISA), pp 112–115. <https://doi.org/10.1109/WISA.2015.70>
- Li W, Song M, Tian Y (2019) An ontology-driven cyberinfrastructure for intelligent spatiotemporal question answering and open knowledge discovery. *ISPRS Int J Geo-Inf*. <https://doi.org/10.3390/ijgi8110496>
- Mahfoudh M, Forestier G, Thiry L, Hassenforder M (2015) Algebraic graph transformations for formalizing ontology changes and evolving. *Knowl-Based Syst* 73:212–226. <https://doi.org/10.1016/j.knosys.2014.10.007>
- Meditskos G, Dasiapoulou S, Kompatsiaris I (2016) MetaQ: a knowledge-driven framework for context-aware activity recognition combining SPARQL and OWL 2 activity patterns. *Pervasive Mob Comput* 25:104–124. <https://doi.org/10.1016/j.pmcj.2015.01.007>
- Mihindukulasooriya N, Poveda-Villalón M, Garcia-Castro R, Gomez-Perez A (2017) Collaborative ontology evolution and data quality—an empirical analysis. In: Dragoni M, PovedaVillalón M, JimenezRuiz E (eds) OWL: experiences and directions—reasoner evaluation, OWLED 2016, vol 10161, pp 95–114. [https://doi.org/10.1007/978-3-319-54627-8\\_8](https://doi.org/10.1007/978-3-319-54627-8_8)
- Osborne F, Motta E (2018) Pragmatic ontology evolution: reconciling user requirements and application performance. In: Vrandečić D, Bontcheva K, SuarezFigueroa MC, Presutti V, Celino I, Sabou M, Kaffee LA, Simperl E (eds) Semantic web—ISWC 2018, PT I, vol 11136, issue I, pp 495–512. [https://doi.org/10.1007/978-3-030-00671-6\\_29](https://doi.org/10.1007/978-3-030-00671-6_29)
- Paré G, Trudel M-C, Jaana M, Kitsiou S (2015) Synthesizing information systems knowledge: a typology of literature reviews. *Inf Manag* 52(2):183–199. <https://doi.org/10.1016/j.im.2014.08.008>
- Peixoto R, Cruz C, Silva N (2016) Adaptive learning process for the evolution of ontology-described classification model in big data context. In: Proceedings of the 2016 SAI computing conference (SAI), pp 532–540
- Piovesan L, Anselma L, Terenzianni P (2015) Temporal detection of guideline interactions. In: Proceedings of the international joint conference on biomedical engineering systems and technologies, vol 5, pp 40–50. <https://doi.org/10.5220/0005186300400050>
- Sad-Houari N, Taghezout N, Nador A (2019) A knowledge-based model for managing the ontology evolution: case study of maintenance in SONATRACH. *J Inf Sci* 45(4):529–553. <https://doi.org/10.1177/0165551518802261>
- Saeed NTM, Weber C, Mallak A, Fathi M, Obermaisser R, Kuhnert K (2019) ADISTES ontology for active diagnosis of sensors and actuators in distributed embedded systems. In: 2019 IEEE international

- conference on electro information technology (EIT), pp 572–577. <https://doi.org/10.1109/EIT.2019.8834013>
- Shaban-Nejad A, Haarslev V (2015) Managing changes in distributed biomedical ontologies using hierarchical distributed graph transformation. *Int J Data Min Bioinform* 11(1):53–83. <https://doi.org/10.1504/IJDMB.2015.066334>
- Smoker TM, French T, Liu W, Hodkiewicz MR (2017) Applying cognitive computing to maintainer-collected data. In: 2017 2nd international conference on system reliability and safety (ICSRS), pp 543–551. <https://doi.org/10.1109/ICSRS.2017.8272880>
- Stavropoulos TG, Andreadis S, Kontopoulos E, Kompatsiaris I (2019) SemaDrift: a hybrid method and visual tools to measure semantic drift in ontologies. *J Web Semant* 54(2):87–106. <https://doi.org/10.1016/j.websem.2018.05.001>
- Steinegger M, Melik-Merkumians M, Zajc J, Schitter G (2017) A framework for automatic knowledge-based fault detection in industrial conveyor systems. In: 2017 22nd IEEE international conference on emerging technologies and factory automation (ETFA), pp 1–6. <https://doi.org/10.1109/ETFA.2017.8247705>
- Stojanovic L (2004) Methods and tools for ontology evolution [Doctoral dissertation, University of Karlsruhe, Karlsruhe]. [https://www.researchgate.net/publication/35658911\\_Methods\\_and\\_tools\\_for\\_ontology\\_evolution](https://www.researchgate.net/publication/35658911_Methods_and_tools_for_ontology_evolution)
- Tommasini R, Bonte P, Della Valle E, Mannens E, De Turck F, Ongenaes F (2017) Towards ontology-based event processing. In: Dragoni M, PovedaVillalon M, JimenezRuiz E (eds) Owl: experiences and directions—reasoner evaluation, OWLED 2016, vol 10161, pp 115–127. [https://doi.org/10.1007/978-3-319-54627-8\\_9](https://doi.org/10.1007/978-3-319-54627-8_9)
- Touhami R, Buche P, Dibie J, Ibanescu L (2015) Ontology evolution for experimental data in food. In: Garoufallou E, Hartley RJ, Gaitanou P (eds) Metadata and semantics research, MTSR 2015, vol 544, pp 393–404. [https://doi.org/10.1007/978-3-319-24129-6\\_34](https://doi.org/10.1007/978-3-319-24129-6_34)
- Tsalapati E, Stoilos G, Chortaras A, Stamou G, Koletsos G (2017) Query rewriting under ontology change. *Comput J* 60(3):389–409. <https://doi.org/10.1093/comjnl/bxv120>
- Wang Z, Wang K, Zhuang Z, Qi G (2015) Instance-driven ontology evolution in DL-lite. In: Proceedings of the twenty-ninth AAAI conference on artificial intelligence, pp 1656–1662
- Wang Z, Qian Y, Wang L, Zhang S, Luo X (2019) The extraction of hidden fault diagnostic knowledge in equipment technology manual based on semantic annotation. In: Proceedings of the 2019 8th international conference on software and computer applications, pp 419–424. <https://doi.org/10.1145/3316615.3316659>
- Zhang Y, Xu F (2018) A SPARQL extension with spatial-temporal quantitative query. In: 2018 13th IEEE conference on industrial electronics and applications (ICIEA), pp 554–559. <https://doi.org/10.1109/ICIEA.2018.8397778>
- Zhang R, Guo D, Gao W, Liu L (2016) Modeling ontology evolution via Pi-Calculus. *Inf Sci* 346:286–301. <https://doi.org/10.1016/j.ins.2016.01.059>
- Zheleznyakov D, Kharlamov E, Nutt W, Calvanese D (2019) On expansion and contraction of DL-lite knowledge bases. *J Web Semant*. <https://doi.org/10.1016/j.websem.2018.12.002>
- Zhou Y, Wang Z, He D (2016) Spatial-temporal reasoning of Geovideo data based on ontology. In: 2016 IEEE international geoscience and remote sensing symposium (IGARSS), pp 4470–4473. <https://doi.org/10.1109/IGARSS.2016.7730165>
- Ziembinski RZ (2016) Ontology learning from graph-stream representation of complex process. In: Burduk R, Jackowski K, Kurzynski M, Wozniak M, Zolnierek A (eds) In: Proceedings of the 9th international conference on computer recognition systems, Cores 2015, vol 403, pp 395–405. [https://doi.org/10.1007/978-3-319-26227-7\\_37](https://doi.org/10.1007/978-3-319-26227-7_37)

## Authors and Affiliations

Alda Canito<sup>1</sup>  · Juan Corchado<sup>2</sup>  · Goreti Marreiros<sup>1</sup> 

Juan Corchado  
corchado@usal.es

Goreti Marreiros  
mgt@isep.ipp.pt

<sup>1</sup> GECAD - Research Group on Intelligent Engineering and Computing for Advanced Innovation and Development, Institute of Engineering - Polytechnic of Porto, Rua Dr. António Bernardino de Almeida, No. 431, 4249-015 Porto, Portugal

<sup>2</sup> Department of Computer Science, University of Salamanca, Salamanca, Spain