CrossMark

ORIGINAL ARTICLE

Applications of ontologies in requirements engineering: a systematic review of the literature

**Diego Dermeval · Jéssyka Vilela · Ig Ibert Bittencourt ·
Jaelson Castro · Seiji Isotani · Patrick Brito ·
Alan Silva**

Abstract There is an increase use of ontology-driven approaches to support requirements engineering (RE) activities, such as elicitation, analysis, specification, validation and management of requirements. However, the RE community still lacks a comprehensive understanding of how ontologies are used in RE process. Thus, the main objective of this work is to investigate and better understand how ontologies support RE as well as identify to what extent they have been applied to this field. In order to meet our goal, we conducted a systematic literature review (SLR) to identify the primary studies on the use of ontologies in RE, following a predefined review protocol. We then identified the main RE phases addressed, the requirements modelling styles that have been used in conjunction with ontologies, the types of requirements that have been supported by the use of ontologies and the ontology languages that have been adopted. We also examined the types of contributions reported and looked for evidences of the benefits of ontology-driven RE. In summary, the main findings of this work are: (1) there are empirical evidences of the benefits of using ontologies in RE activities both in industry and academy, specially for reducing ambiguity, inconsistency and incompleteness of requirements; (2) the majority of studies only partially address the RE process; (3) there is a great diversity of RE modelling styles supported by ontologies; (4) most studies addressed only functional requirements; (5) several studies describe the use/development of tools to support different types of ontology-driven RE approaches; (6) about half of the studies followed W3C recommendations on ontology-related languages; and (7) a great variety of RE ontologies were identified; nevertheless, none of them has been broadly adopted by the community. Finally, we conclude this work by showing several promising research opportunities that are quite important and interesting but underexplored in current research and practice.

Keywords Ontologies · Requirements engineering · Systematic literature review

D. Dermeval (✉)
Systems and Computing Department, Federal University of Campina Grande, Campina Grande, PB 58109-900, Brazil
e-mail: diegodermeval@copin.ufcg.edu.br

D. Dermeval
Campus Arapiraca, Federal University of Alagoas, Penedo, AL 57200-000, Brazil

J. Vilela · J. Castro
Informatics Center, Federal University of Pernambuco, Recife, PE 50.740-560, Brazil
e-mail: jffv@cin.ufpe.br

J. Castro
e-mail: jbc@cin.ufpe.br

I. I. Bittencourt · P. Brito · A. Silva
Computing Institute, Federal University of Alagoas, Maceió, AL 57072-900, Brazil
e-mail: ig.ibert@ic.ufal.br

P. Brito
e-mail: patrick@ic.ufal.br

A. Silva
e-mail: alanpedro@ic.ufal.br

S. Isotani
Institute of Mathematics and Computational Sciences, University of São Paulo, São Carlos, SP 13566-590, Brazil
e-mail: sisotani@icmc.usp.br

1 Introduction

Requirements engineering (RE) is concerned with the elicitation, analysis, specification, validation and

management of requirements [41]. It is well known that RE contributes to the improvement of the quality of software development decreasing the risk of budget overrun, delays and project failures [58].

In the context of software development, there is a growing interest in the use of ontologies [23, 53], which is an "explicit specification of a conceptualization" [26]. It is "explicit" because of its classes and properties visibility. Conceptualization is understood to be an abstract and simplified version of the world to be represented. Moreover, ontologies can be logically reasoned and shared within a specific domain [28]. Every knowledge base, knowledge-based system, or knowledge-level agent is committed to some conceptualization, explicitly or implicitly [13]. Thus, ontologies are a standard form for representing the concepts within a domain, as well as the relationships between those concepts in a way that allows automated reasoning.

Due to such benefits, the concept of ontology has been used in RE to minimize or resolve different types of problems. It is used, for instance, to specify more complete and unambiguous requirements, to perform consistency analysis in requirements, to explicitly model domain knowledge, to manage requirements knowledge and requirements changes and so on [9, 60].

There are some studies that investigated the use of ontologies in Software Engineering [23, 57] and, specifically, the use of ontologies in RE [9]. However, these studies did not capture all the aspects and evidences that we are interested. Moreover, they did not perform any kind of systematic investigation of the literature covering ontologies in RE.

Hence, the objective of this work is to conduct a systematic review of the literature to find out how ontologies support RE. It is also important to investigate whether there are real evidences of the improvements of using ontologies to support RE activities. Moreover, we also need to understand: (1) which are the main phases of the RE process amenable to the use of ontologies; (2) what types of requirements modelling styles have been supported by ontologies; (3) if ontologies have been used to support both functional and non-functional requirements; (4) what RE problems have been addressed by ontology-driven contributions and the types of these contributions; (5) which languages have been used in the ontology-driven RE process; and (6) if the RE ontologies have been reused and how they have been reused.

In this paper, we use the systematic literature review (SLR) method [39] to identify, evaluate, interpret and synthesize the available studies to answer particular research questions on the symbiosis of ontologies and RE, and to establish the state of evidence with in-depth analysis. The SLR method has already been used in different topics within RE. For instance, RE education [51], use case models [20], transformation between user requirements and analysis models [66], technology transfer decision in RE [35] and stakeholder identification methods in requirements elicitation [52].

Our purpose is to better understand how ontologies support RE and identify in what ways it has being applied to this field. This paper extends a previous SLR that we conducted [14] considering a reduced number of research questions (only five) on the ontology-driven RE topic. We also extend the selection process of the previous SLR by reviewing the references lists of the papers included in it. Additionally, we perform a more detailed and quantitative analysis of the papers' results.

In this way, this paper presents the results of an extended SLR on studies published from January 2007 to October 2013 and was conducted following a predefined review protocol. Our decision on such period was made to reduce repetitive effort and make use of existing work, since there is a general survey on the use of ontologies in software engineering. Ruiz and Hilera [57] summarize research about the use of ontology to support software engineering/RE until 2006.

This paper is organized as follows. Section 2 describes the SLR method used in this review and the context within it is inserted. Section 3 first presents the results of the quality assessment and an overview of the studies. It then reports the findings of the review along with a detailed analysis and discussion of each research question. Section 4 discusses the scope of this SLR, some related works and threats to the validity of our work and points out further research to be explored on the use of ontologies in RE. Finally, Sect. 5 presents conclusions and future works.

2 Methods

A SLR is a means of identifying, evaluating and interpreting the available research findings related to a research question, topic area, or phenomenon. The main purpose for conducting a systematic review is to gather evidence on which to base conclusions [39].

In order to perform this SLR, the guidelines and the systematic review protocol template proposed by Kitchenham and Charters [39] were used. According to these guidelines, the SLR process includes several activities, which can be grouped in three main phases: planning of the SLR, conducting the SLR and reporting the SLR. It consists of the following steps: (1) identification of the need for a systematic review; (2) formulation of a focused review question; (3) a comprehensive, exhaustive search for primary studies; (4) quality assessment of included studies; (5) identification of the data needed to

answer the research question; (6) data extraction; (7) summary and synthesis of study results; (8) interpretation of the results to determine their applicability; and (9) report-writing.

A software tool was used to support the SRL protocol definition. The tool, called StArt (State of the Art through Systematic Reviews) [43], is used to provide support to researchers conducting SLRs. StArt has been empirically evaluated, and it was demonstrated that such tool had positive results in the execution of SLRs [31].

Before describing the research questions of this review, we present the context (RE and ontologies) in which these questions are included.

2.1 Requirements engineering

The RE process is composed by several activities: Requirements Elicitation, Requirements Analysis and Negotiation, Requirements Documentation and Requirements Validation [41].

The requirements elicitation phase involves understanding the application domain, the specific problem to be solved, the organizational needs and constraints and the specific facilities required by the system stakeholders.

Requirements analysis and negotiation are concerned with the high-level statement of requirements elicited from stakeholders. The objective of this activity is to establish an agreed set of requirements which are complete and consistent.

Requirements validation is the final stage of RE. The aim of this phase is to check the final draft of a requirements document to certify it represents an acceptable description of the system to be implemented. The inputs to the validation process are the requirements document, organizational standards and implicit organizational knowledge. The outputs are a list of requirements problems and agreed actions to address these problems.

In addition to these activities, Kotonya and Sommerville [41] also emphasize the need to manage requirements. Requirements management is the process of managing changes in systems' requirements. Therefore, this process supports others RE and system development activities. Besides, it is carried out in parallel with other RE activities and continues after the first version of the requirements document has been delivered.

2.2 Ontologies

The term ontology comes from a branch of philosophy that deals with the nature of being [12]. The term was introduced in computer science by artificial intelligence researchers who constructed computer models with some kind of automated reasoning. From the 90's, ontologies began to be treated as an integral part of knowledge-based systems, defined as an explicit specification of conceptualization [26].

In the computer and information science context, an ontology defines a set of representational primitives in a particular knowledge area [49]. The usually adopted representational primitives are classes, attributes and relationships, including their meanings and restrictions. Ontologies are typically specified with languages that allow some kind of abstraction from data structures and from implementation strategies [27].

Ontologies languages are used for domain formalization by defining classes and properties for these classes, individuals (that instantiate the classes), properties of individuals, and statements on these individuals. It also allows to reason about these classes and individuals according to formal semantics defined by the language. On the one hand, the defined models gain a formal representation, which can be useful to support software engineering activities. On the other hand, using ontologies can support the automated reasoning and inference on such models.

Ontologies can be written down in a wide variety of languages and notations, such as Description logics [6], First-order logics, Relational model, UML and so on. However, ontologies are generally represented using one of the variants of the Web Ontology Language (OWL) [48], which is part of the technologies stack defined by the World Wide Web Consortium (W3C) for Semantic Web.

The OWL language has its roots in Description Logics and provides formal and clear semantics for the definition of concepts and their relationships. OWL ontologies are often serialized using an RDF/XML representation—also part of the stack of W3C technologies—which is a triple format that models information using triples in the form of subject-predicate-object expressions. The information represented in RDF format (e.g., OWL ontologies) can be queried using a standard RDF query language called SPARQL [54], which is an SQL-like language. Additionally, the Semantic Web Rule Language (SWRL) [32] proposal extends the set of OWL axioms to include Horn-like rules in order to provide even more inference capabilities to it.

2.3 Research questions

This systematic review's purpose is to better understand how ontologies support RE and identify to what extent they have been applied to this field. Thus, we intend to answer the main research question:

> How are ontologies supporting the Requirements Engineering process?

Based on the main research question, specific questions were raised according to RE aspects that we are interested.

Table 1 Research questions and motivations

Research question	Description and motivation
RQ1. What phases of the requirements engineering process have been supported by the use of ontologies?	This question provides a starting point to understand what are the main phases (elicitation, analysis, specification, validation and management) of the requirements engineering process supported by the use of ontologies
RQ2. What styles (scenario-based, goal-oriented, etc) of software requirements modelling have been supported by the use of ontologies?	The answer to this question allows the identification of main styles of software requirements modelling (e.g., scenario-based, goal-oriented, textual requirements and so on) that have been supported by the use of ontologies. It may help to identify which requirements styles are attracting more attention to ontology community.
RQ3. What types (functional and/or non-functional) of requirements have been supported by the use of ontologies?	This question intends to identify what is the distribution of the studies with respect to the types of requirements (functional and/or non-functional) addressed. It is important to investigate whether ontologies have been used to improve both functional and non-functional requirements
RQ4. How are ontologies contributing to the solution of requirements engineering problems?	This question aims to describe contributions to solution of well-known RE problems. It is important because it provides a set of contributions regarding the use of ontologies to address some well-known RE research problems, which can be useful to researchers that might be interested in using ontologies in RE
RQ4.1 What are the types of these contributions?	This sub-question intends to classify the contributions by its type, for instance, model, tool, process and method proposed in the study.
RQ5. Which ontology languages have been used in the ontology-driven requirements engineering methods?	This question identifies which are the main ontology languages (e.g., OWL, SPARQL, SWRL, UML and so on) been used to support requirements engineering methods. The answer to this question is also important because it can serve as a guide to researchers that might use some specific ontology language in RE
RQ6. Which studies have reused requirements engineering ontologies?	The answer to this question indicates the existing RE ontologies and also presents how they are been reused in the studies included in the review. Thus, it is important because it identifies a set of RE ontologies which may be reused by researches on the use of ontologies in RE
RQ7. Are there evidences of benefits of the use of ontologies in the RE Process?	This question intends to analyse whether such studies provide some evidence that the use of ontologies benefits the requirements engineering process. These evidences should consider positive and negative results including empirical and non-empirical evaluation. They are important since they form a knowledge base about the use of ontologies in RE

These questions, their descriptions and motivations are described in Table 1.

2.4 Inclusion and exclusion criteria

The aim of defining a criterion is to identify those primary papers which provide direct evidence about the research questions and also to reduce the likelihood of bias [39]. It is worth noting that we consider as primary papers the studies which present some kind of proposal to the area and/or present some kind of empirical validation of its contributions, whereas secondary papers are studies which only review a topic area, i.e., a SLR.

The studies were eligible for inclusion in the review if they presented a peer-reviewed primary study, published since January 2007 to October 2013 and that presented some contribution on the use of ontology to support RE activities. They should also satisfy the minimum quality threshold (see Sect. 2.6). Our decision on such period was made to reduce repetitive effort and to make use of existing work, since there is a general survey on the use of ontologies in software engineering [57]) published in 2006 covering the ontology-driven RE studies before this period.

Studies were excluded if they were secondary, short papers, non-peer-reviewed, duplicated, non-English written, grey literature papers (e.g., books, theses, dissertations and so on), redundant papers of same authorship and if their focus was not using ontologies to support RE activities. Furthermore, this research is concerned with generic and technology/paradigm independent contributions to software development, for this reason, domain-specific papers, Service-Oriented Architecture (SOA) [34], Agent-Oriented Software Development (AOSE) [65], Business Process Models (BPM) [63] and Information Technology (IT) exclusive papers were also excluded. Ontology engineering [25] papers were also out of the scope of this review. The summarized inclusion and exclusion criteria are presented in Table 2.

2.5 Sources selection and search

The search strategy included only electronic databases and was validated by experts on the ontology and RE areas. By using a search string and based on [10], the following electronic databases were automatically searched:

Table 2 Inclusion/exclusion criteria

#	Inclusion criterion
1	Primary studies
2	Peer-reviewed studies
3	Studies that use ontology to support requirements engineering
4	Study published between January 2007 and October 2013
5	Satisfies the minimum quality threshold

#	Exclusion criterion
6	Secondary studies
7	Short papers (≤5 pages)
8	Non-peer-reviewed studies
9	Duplicated studies (only one copy of each study was included)
10	Non-English written papers
11	Service-oriented architecture (SOA) exclusive papers
12	Agent-oriented software development (AOSE) exclusive papers
13	Domain-specific papers
14	Business process models (BPM) exclusive papers
15	Studies that do not use ontology in software development
16	Information technology (IT) exclusive papers
17	Grey literature
18	Ontology engineering papers
19	Redundant paper of same authorship
20	Studies that do not use ontology in requirements engineering

ScienceDirect,[1] ISI Web of Science,[2] Scopus,[3] SpringerLink,[4] ACM Digital Library,[5] IEEE Xplore[6] and Compendex.[7]

Figure 1 shows the systematic review process and the number of papers identified at each stage. Before describing these stages, it is worth emphasizing that, although the scope of this paper is reviewing the use of ontologies in RE process, this research is part of an ongoing work which intends to review the use of ontologies in the entire software engineering process. Hence, the search and selection strategy (i.e., the search string and Steps 1–5) aims to capture studies related to the main aspects of software engineering area for instance, software requirements, software design, software implementation, software testing, software management and so on. As such they will be useful for several other studies. The papers related to RE, which are the focus of this review, are only identified and

selected in Step 6 of the process, as it will be further described.

In Step 1, the studies were obtained from electronic databases using the following search terms:

1. ontology OR ontologies
2. "software engineering"
3. "requirements" OR "requirements engineering" OR "software analysis" OR "domain analysis"
4. "software architecture" OR "software design" OR "architectural design"
5. "software testing" OR "software verification" OR "software validation" OR "unit testing"
6. "software maintenance" OR "software evolution"
7. "software configuration management"
8. "software quality"
9. "software metric" OR "software measurement"
10. "software process" OR "process model" OR "software process model"
11. "software management"
12. "software reuse"
13. "software modelling" OR "software modelling"
14. "software programming"
15. "software engineering tool" OR "software engineering method"
16. "ontology-driven software"

These search terms for different types of software engineering articles were combined in the following way:

(1 AND 2 AND (3 OR 4 OR 5 OR 6 OR 7 OR 8 OR 9 OR 10 OR 11 OR 12 OR 13 OR 14 OR 15)) OR 16

The definition of these terms was based on three important references: (1) the taxonomies about the use of ontology in Software Engineering presented by Ruiz and Hilera [57]; (2) the survey authored by Gazevic et al. [24] about the use of ontology in software engineering; and (3) the topics within scope of the Information and Software Technology Journal [21], which encourages the conduction and publication of SLRs. Furthermore, as Ruiz and Hilera and Gazevic et al. summarize research about the use of ontology to support software engineering/RE until 2006, our search only considered the period between January 2007 and October 2013, which is an inclusion criterion, as described in Sect. 2.4.

The search results (12,270 papers) were automatic downloaded and were entered into and organized with the aid of StArt tool. Figure 1 depicts the steps of the selection process showing the number of studies in each one these steps. Besides, the two frames on the right side of Fig. 1 presents the exclusion criteria which were exclusively applied to the studies in Steps 4 and 5.

At Step 2, duplicated papers were automatically detected and removed using the StArt tool, remaining a set of 9,764 papers. Then, in Step 3 authors reviewed the
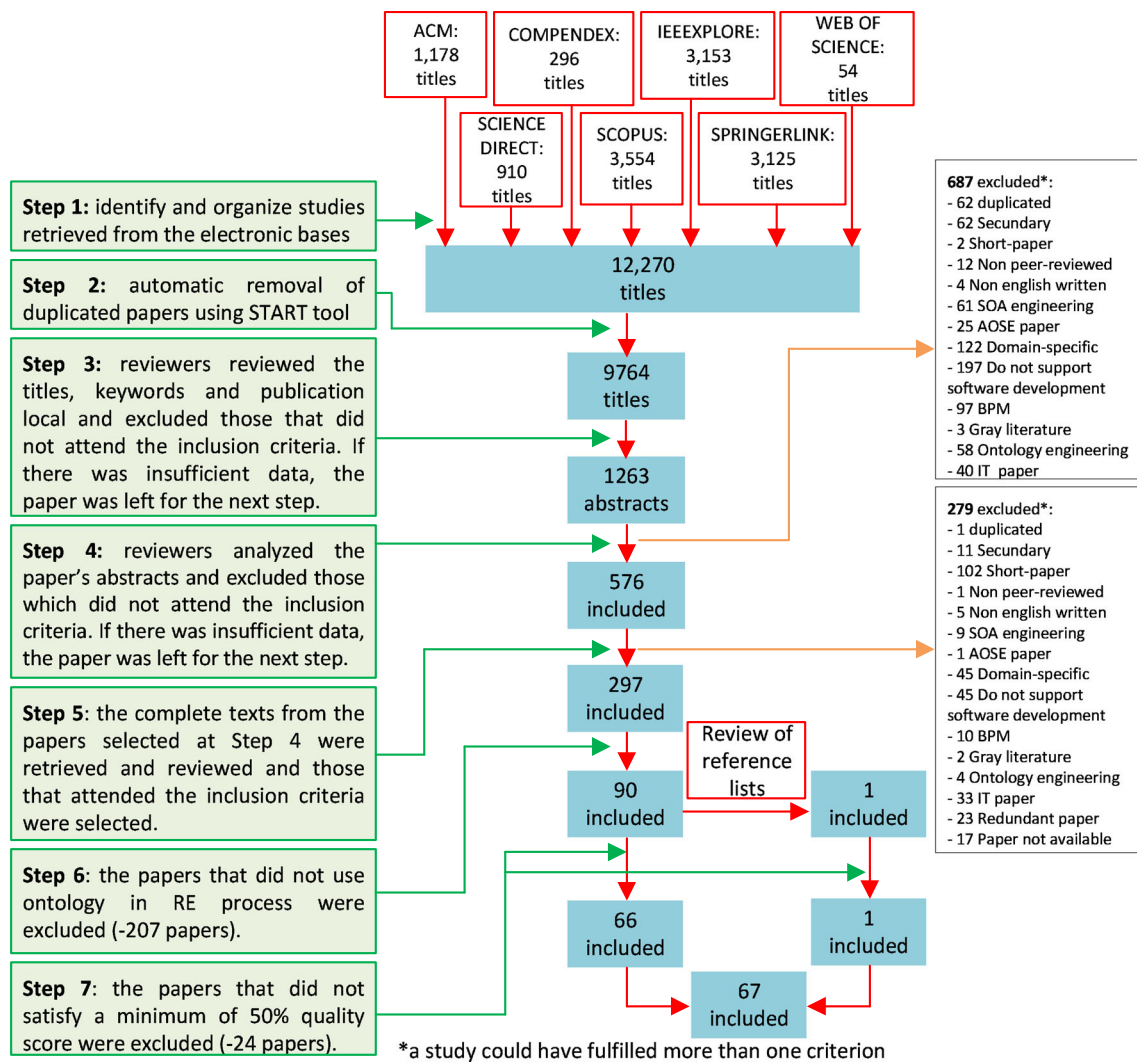
ACM: 1,178 titles

COMPENDEX: 296 titles

IEEEXPLORE: 3,153 titles

WEB OF SCIENCE: 54 titles

SCIENCE DIRECT: 910 titles

SCOPUS: 3,554 titles

SPRINGERLINK: 3,125 titles

Step 1: identify and organize studies retrieved from the electronic bases

Step 2: automatic removal of duplicated papers using START tool

Step 3: reviewers reviewed the titles, keywords and publication local and excluded those that did not attend the inclusion criteria. If there was insufficient data, the paper was left for the next step.

Step 4: reviewers analyzed the paper's abstracts and excluded those which did not attend the inclusion criteria. If there was insufficient data, the paper was left for the next step.

Step 5: the complete texts from the papers selected at Step 4 were retrieved and reviewed and those that attended the inclusion criteria were selected.

Step 6: the papers that did not use ontology in RE process were excluded (-207 papers).

Step 7: the papers that did not satisfy a minimum of 50% quality score were excluded (-24 papers).

12,270 titles

9764 titles

1263 abstracts

576 included

297 included

90 included

Review of reference lists

1 included

66 included

1 included

67 included

687 excluded*:
- 62 duplicated
- 62 Secundary
- 2 Short-paper
- 12 Non peer-reviewed
- 4 Non english written
- 61 SOA engineering
- 25 AOSE paper
- 122 Domain-specific
- 197 Do not support software development
- 97 BPM
- 3 Gray literature
- 58 Ontology engineering
- 40 IT paper

279 excluded*:
- 1 duplicated
- 11 Secundary
- 102 Short-paper
- 1 Non peer-reviewed
- 5 Non english written
- 9 SOA engineering
- 1 AOSE paper
- 45 Domain-specific
- 45 Do not support software development
- 10 BPM
- 2 Gray literature
- 4 Ontology engineering
- 33 IT paper
- 23 Redundant paper
- 17 Paper not available

*a study could have fulfilled more than one criterion

Fig. 1 Paper selection flowchart

titles, keywords and publication local of each paper and excluded those that were not related to the research questions (−8,501). If there was insufficient data, the paper was left for the next assessment. After finishing the Step 3, 1,263 papers remained in the selection process and reviewers analysed, in Step 4, the papers abstracts and excluded those according to 13 exclusion criteria (−687 papers), as indicated on the right side of Fig. 1. If there was insufficient data, the paper was left for the next step.

In Step 5, the complete texts from the papers selected at Step 4 (576 papers) were retrieved, the introduction and conclusion of each paper was read and each paper was full-screened. Papers were excluded according to 15 exclusion criteria (−279 papers), as also indicated in the bottom frame on the right side of Fig. 1.

Until Step 5, any SE paper supported by the use of ontologies was considered to be included in the review.

Recall that this is intentional, as we may use the studies identified so far for several researches under development. Hence, the specific exclusion criterion was only applied, in Step 6, to the 297 remaining studies of Step 5, in order to filter the papers exclusively related to RE (the focus of this paper).

After finishing Step 6, 207 papers that do not use ontology in RE—but use ontology in other software engineering subdomains—were excluded from this review, remaining 90 studies. As an additional search, the reference lists of these RE papers were also reviewed in order to identify studies which satisfy the inclusion criteria and led to the inclusion of 1 (one) study, resulting in a total of 91 RE papers to be assessed in the next step.

At last, in Step 7, it was performed a quality assessment in the included papers (91) and those that do not satisfy a minimum quality score were excluded (see next section), remaining a set of 67 papers to be considered in this review.

Table 3 Study quality assessment criteria

#	Questions	Possible answers
1	Is there a rationale for why the study was undertaken? [47]	$Y = 1, N = 0, P = 0.5$
2	Is the paper based on research (or is it merely a "lessons learned" report based on expert opinion)? [17]	$Y = 1, N = 0$
3	Is there a clear statement of the goals of the research? [17]	$Y = 1, N = 0, P = 0.5$
4	Does the study reuse an existing ontology?	$Y = 1, N = 0$
5	Is the proposed technique clearly described? [2]	$Y = 1, N = 0, P = 0.5$
6	Is there an adequate description of the context (industry, laboratory setting, products used and so on) in which the research was carried out? [17, 47]	$Y = 1, N = 0, P = 0.5$
7	How many phases of the requirements engineering process does the study support?	$\frac{\text{Num of Phases}}{\text{Total of Phases}}$
8	Is the study supported by a tool?	$Y = 1, N = 0$
9	Was the study empirically evaluated?	$Y = 1, N = 0$
10	Is there a discussion about the results of the study?	$Y = 1, N = 0, P = 0.5$
11	Are the limitations of this study explicitly discussed? [15]	$Y = 1, N = 0, P = 0.5$
12	Does the research also add value to the industrial community? [2, 17]	$Y = 1, P = 0.5$

2.6 Quality assessment

The quality assessment (QA) of selected studies was achieved by a scoring technique to evaluate the credibility, completeness and relevance of the selected studies. All papers were evaluated against a set of 12 quality criteria. Seven of them were adapted from existing study quality assessment criteria used in the literature, the remaining five questions were proposed according to the scope and research questions of this SLR. The assessment instrument used is presented in Table 3. Q1, Q2, Q3, Q5, Q6, Q11 and Q12 were adopted from the literature, while Q4, Q7, Q8, Q9 and Q10 were proposed.

We relied on SLRs published in a high reputation venue (i.e., Information and Software Technology Journal) in the context of empirical software engineering research to define seven of the quality assessment criteria. In particular, we adapted some of our criteria following the works by Mahdavi–Hezavehi [47] (Q1 and Q6), Dyba and Dingsyr [17] (Q2, Q3, Q6 and Q12), Achimugu et al. [2] (Q5 and Q12) and Ding et al. [15] (Q11).

The scores of questions Q2, Q4, Q8 and Q9 were determined using a two-grade scale score (Yes/No). If the answer was Yes, the study received 1 point in this question; otherwise, it received 0 point. Besides these alternatives, the questions Q1, Q3, Q5, Q6, Q10, Q11 also allowed a third one. If the contribution was not so strong, the study received 0.5 point, consisting in a three-grade scale score to these questions. Q12 receives 1 point if the study is applied in industry and 0.5 point if its setting is academy.

We also want to ascertain the contribution of the selected studies to RE through its coverage. For this, we used the RE process defined by Kotonya and Sommervile [41]. According to the authors, this process is divided in four phases: requirements elicitation, requirements analysis and negotiation, requirements documentation, and requirements validation. Moreover, the authors also emphasize the requirements management[8] as an important activity of the RE process. Thus, the score of Q7 was determined through the following proportion:

$\frac{\text{Number of Phases}}{\text{Total of Phases}}$, where the numerator is the number of RE phases covered by the study and the denominator is the total of phases considered in the RE process (five).

The study quality score is computed by finding the sum of all its scores of the answers to the questions. Each selected paper was assessed independently by the authors. All discrepancies on the scores were discussed among the authors, and the study was reevaluated in cases of non-agreement with the aim of reaching consensus.

2.7 Data extraction and synthesis

After the definition of the search and the selection processes, the data extraction process was performed by reading the introduction and conclusion; and full-text screening each one of the selected papers. In order to guide this data extraction, the data collection from Kitchenham and Charters [39] was adopted. During this stage, data were extracted from each of the 67 primary studies included in this systematic review according to a predefined extraction form (see Table 14 in "Appendix 1"). This form enabled us to record full details of the papers under review and to be specific about how each of them addressed our research questions. As well the selection process, the data extraction was full aided by the StArt tool.

---

[8] In this paper we also consider requirements management as a requirements phase.

The data were tabulated to show general information about the studies: identifier, authors, year, paper type, application context, research method and quality assessment scores. Furthermore, we also present a bubble chart considering the application context, research method and publication year information of the studies in order to provide a deeper visualization of these information. Additionally, the data regarding each research question were tabulated. The results of RQ2 and RQ7 are also graphically presented using bubble charts. Particularly, in the discussion of RQ7 (Sect. 3.9), we present a deeper analysis of ontologies benefits in RE with the aid of bubble and bar charts.

3 Results and analysis

A total of 67 studies met the inclusion criteria and their data were extracted. Prior to presenting the results and analysis for each research question, we depict the quality assessment results and give a detailed overview of the general characteristics of the studies.

3.1 Quality assessment results

The quality assessment of the selected studies is useful to increase the accuracy of the data extraction results. This evaluation helped to determine the validity of the inferences proffered and in ascertaining the credibility and coherent synthesis of results.

The quality assessment results are showed in Table 4 according to the assessment questions described in Table 3. The scores of all studies are no less than 50%, and the average score is 7.82. We chose the minimum of 50% quality with the aim to establish an acceptable quality threshold for the articles. Regarding the averages of specific quality criteria, Q2, Q1, Q3 and Q4 received the highest average scores (>0.9). Q6, Q10, Q12 and Q8 received, in average, intermediary scores between 0.57 and 0.62. Q7, Q11, Q9 and Q4 criteria received the lowest average scores ($\leq$0.5). It is worth noting that the lowest average score—of Q4, with 0.37—indicates that the great part of the studies is not concerned with reuse of ontologies developed by other authors.

The overall quality of the selected studies is acceptable. Taken together, these 12 criteria provided a measure of the extent to which we could be confident that a particular study's findings could make a valuable contribution to this review.

3.2 Overview of the studies

In following, we depict general characteristics of the studies included in the review: year of publication, type of source, research method and application context.

3.2.1 Publication year

The reviewed papers were published between 2007 and 2013. From a temporal point of view (Fig. 2), an increasing number of publications in the context of this review is observed since 2007. 2010 (19.4 %) is the year with most publications, followed by 2009 (17.91 %), 2012 (16.42 %), 2011 (14.93 %), 2013 (11.94 %), 2008 (10.45 %) and 2007 (8.96 %).

Although the apparent increasing of the number of studies on the use of ontologies in RE, it is difficult to point out that there is a trend in such topic. Indeed, we can observe that researchers are currently concerned with the topic, but we can not state that there is some kind of tendency. It is also worth noting that, as the search process of this review was performed in October 2013, a slight decrease in the number of publications would be expected in 2013 because some papers might be in press.

3.2.2 Application context

The study settings were categorized either as an industry or academic context. The majority of the papers (77.6 %; 52 studies) are considered academic. However, it is worth noting that a considerable amount of the studies were conducted in an industrial setting (22.4 %; 15 studies), indicating that, even though the concept of ontology is not widespread within the RE community, its use has also been significantly investigated in industry, as it will be further discussed in Sect. 3.8.

3.2.3 Research method

The classification of publications was based on the categories (controlled experiment, quasi-experiment, case study, survey research, ethnography and action research) defined by Easterbrook et al. [19]. However, we have defined two extra categories: illustrative scenario and not applicable. The first one is appropriate for papers that just evaluate their contributions using small examples. The second one refers to the papers that do not present any kind of research method in the study.

Illustrative scenarios (55.22 %; 37 studies) constitute majority of the studies, followed by controlled experiments (20.19 %; 14 studies), case studies (16.42 %; 11 studies), not applicable (5.97 %; 4 studies) and quasi-experiment (1.49 %; 1 study). There were no survey research, ethnography and action research papers in our classification.

Figure 3 presents a bubbleplot distributed over three dimensions regarding three characteristics of the studies: year of publication, application context and research method. The left part in Fig. 3 denotes the relationship between studies and year of publication. The number in a bubble represents the number of studies on a specific type

Table 4 List of papers included in the review along with their quality scores

ID	Author	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12	Total Score	Qual. (%)
S01	Al Balushi et al. [67]	1.0	1.0	1.0	0.0	1.0	1.0	1.0	1.0	1.0	0.5	0.0	0.5	9.0	75.0
S02	Anwer et al. [68]	1.0	1.0	0.5	1.0	1.0	0.0	1.0	0.0	0.0	0.0	0.0	1.0	6.5	54.2
S03	Aranda et al. [69]	0.5	1.0	1.0	0.0	1.0	1.0	0.2	0.0	1.0	1.0	1.0	0.5	8.2	68.3
S04	Assawamekin et al. [70]	1.0	1.0	1.0	1.0	1.0	0.5	0.4	1.0	1.0	1.0	0.0	0.5	9.4	78.3
S05	Bagheri et al. [71]	1.0	1.0	1.0	0.0	1.0	0.0	0.2	1.0	0.0	1.0	0.0	0.5	6.7	55.8
S06	Bicchierai et al. [72]	1.0	1.0	1.0	1.0	1.0	1.0	0.6	1.0	0.0	1.0	0.0	1.0	9.6	80.0
S07	Boukhari et al. [73]	1.0	1.0	0.5	0.0	1.0	0.5	0.2	1.0	0.0	0.5	0.0	0.5	6.2	51.7
S08	Cardei et al. [74]	0.5	1.0	1.0	0.0	1.0	0.0	0.6	1.0	0.0	0.5	0.0	0.5	6.1	50.8
S09	Castañeda et al. [75]	1.0	1.0	1.0	0.0	1.0	0.5	0.4	0.0	0.0	1.0	0.0	0.5	6.4	53.3
S10	Castillo et al. [76]	1.0	1.0	1.0	0.0	1.0	0.5	0.4	0.0	0.0	1.0	0.0	0.5	6.4	53.3
S11	Chicaiza et al. [77]	1.0	1.0	1.0	1.0	1.0	0.5	0.4	0.0	0.0	0.5	0.5	0.5	7.4	61.7
S12	Daramola et al. [78]	1.0	1.0	1.0	0.0	1.0	1.0	0.4	1.0	1.0	1.0	1.0	0.5	9.9	82.5
S13	Daramola et al. [79]	1.0	1.0	1.0	0.0	1.0	1.0	0.6	1.0	1.0	1.0	1.0	1.0	10.6	88.3
S14	de Lima et al. [107]	1.0	1.0	1.0	1.0	0.5	0.5	0.4	1.0	0.0	0.5	0.0	0.5	7.4	61.7
S15	Diallo et al. [80]	1.0	1.0	1.0	0.0	1.0	0.5	0.2	0.0	0.0	1.0	0.0	0.5	6.2	51.7
S16	Dzung et al. [82]	0.5	1.0	1.0	1.0	0.5	0.5	0.4	1.0	1.0	0.5	0.5	0.5	8.4	70.0
S17	Dzung et al. [81]	0.5	1.0	0.5	0.0	0.5	1.0	0.4	1.0	1.0	1.0	0.0	0.5	7.4	61.7
S18	Elahi et al. [83]	1.0	1.0	1.0	0.0	1.0	0.5	0.2	0.0	0.0	0.5	0.5	0.5	6.2	51.7
S19	Farfeleder et al. [84]	1.0	1.0	1.0	0.0	0.5	1.0	0.4	0.0	0.0	1.0	0.5	0.5	7.9	65.8
S20	Gailly et al. [85]	1.0	1.0	1.0	0.0	1.0	0.5	0.4	0.0	0.0	0.5	0.5	0.5	6.4	53.3
S21	Gandhi et al. [86]	1.0	1.0	1.0	1.0	1.0	1.0	0.2	1.0	1.0	1.0	1.0	1.0	11.2	93.3
S22	Ghaisas et al. [87]	1.0	1.0	1.0	0.0	1.0	1.0	0.2	1.0	1.0	1.0	0.5	1.0	9.7	80.8
S23	Groner et al. [88]	1.0	1.0	1.0	0.0	1.0	1.0	0.4	0.0	0.0	0.5	0.0	0.5	6.4	53.3
S24	Guizzardi et al. [89]	1.0	1.0	1.0	1.0	1.0	0.5	0.2	0.0	0.0	0.5	0.0	0.5	6.7	55.8
S25	Guo et al. [90]	1.0	1.0	1.0	0.0	1.0	1.0	0.4	1.0	1.0	1.0	1.0	0.5	9.9	82.5
S26	Hayashi et al. [91]	1.0	1.0	1.0	0.0	1.0	1.0	0.4	1.0	1.0	1.0	0.5	0.5	9.4	78.3
S27	He et al. [92]	1.0	1.0	1.0	0.0	1.0	0.5	0.4	0.0	0.0	0.5	0.5	0.5	6.4	53.3
S28	Hickey et al. [93]	1.0	1.0	1.0	0.0	1.0	0.0	0.4	0.0	0.0	1.0	1.0	0.5	6.9	57.5
S29	Hoss et al. [94]	1.0	1.0	0.5	1.0	1.0	0.5	0.6	0.0	0.0	0.5	0.0	0.5	6.6	55.0
S30	Hu et al. [95]	1.0	1.0	1.0	0.0	0.5	0.5	0.4	0.0	0.0	0.5	1.0	0.5	6.4	53.3
S31	Innab et al. [96]	1.0	1.0	1.0	0.0	1.0	0.5	0.6	0.0	0.0	0.5	0.0	0.5	6.1	50.8
S32	Jin et al. [97]	1.0	1.0	1.0	1.0	1.0	0.5	0.4	0.0	0.0	0.0	0.0	0.5	6.4	53.3
S33	Kaiya et al. [98]	1.0	1.0	1.0	1.4	1.0	1.0	0.2	1.8	1.0	1.0	1.0	0.5	10.7	89.2
S34	Karwowski et al. [99]	1.0	0.0	1.0	0.0	1.0	0.5	0.2	1.0	0.0	0.0	0.5	1.0	6.2	51.7
S35	Kassab et al. [100]	1.0	1.0	1.0	1.0	1.0	1.0	0.2	1.0	1.0	1.0	0.5	1.0	10.7	89.2
S36	Kitamura et al. [101]	1.0	1.0	1.0	1.0	1.0	1.0	0.6	1.0	1.0	1.0	1.0	0.5	11.1	92.5
S37	Kof et al. [102]	1.0	1.0	1.0	0.0	1.0	1.0	0.6	0.0	1.0	1.0	1.0	0.5	8.9	74.2
S38	Kossmann et al. [40]	1.0	1.0	1.0	1.0	0.5	1.0	0.4	1.0	1.0	1.0	0.5	1.0	10.4	86.7
S39	Kroha et al. [103]	0.5	1.0	1.0	1.0	1.0	0.0	0.4	1.0	0.0	0.5	0.5	0.5	7.4	61.7
S40	Lasheras et al. [104]	1.0	1.0	1.0	0.0	1.0	0.5	0.4	0.0	0.0	0.5	0.0	1.0	6.4	53.3
S41	Li et al. [105]	1.0	1.0	1.0	0.0	1.0	1.0	0.4	0.0	0.0	0.0	0.5	0.5	6.4	53.3
S42	Lima et al. [108]	1.0	1.0	0.5	0.0	1.0	1.0	0.4	0.0	1.0	1.0	0.0	1.0	7.9	65.8
S43	Liu [109]	1.0	1.0	1.0	1.0	1.0	0.5	0.6	1.0	0.0	0.0	0.5	0.5	8.1	67.5
S44	Liu et al. [110]	1.0	1.0	1.0	0.0	1.0	0.5	0.6	1.0	1.0	0.5	0.5	1.0	9.1	75.8
S45	López et al. [111]	1.0	1.0	1.0	1.0	0.5	0.0	0.4	1.0	0.0	0.5	0.5	0.5	7.4	61.7
S46	Machado et al. [112]	1.0	1.0	1.0	1.0	1.0	0.5	0.2	0.0	0.0	0.0	0.5	0.5	6.7	55.8
S47	Moser et al. [113]	1.0	1.0	1.0	0.0	1.0	1.0	0.4	1.0	1.0	1.0	1.0	1.0	10.4	86.7
S48	Moshirpour et al. [114]	1.0	1.0	1.0	0.0	1.0	0.5	0.4	1.0	0.0	0.0	0.0	0.5	6.4	53.3

Table 4 continued

ID	Author	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12	Total Score	Qual. (%)
S49	Mtibaav[115]	1.0	1.0	1.0	0.0	1.0	0.5	0.2	1.0	0.0	0.0	0.0	0.5	6.2	51.7
S50	Nguyen et al. [116]	1.0	1.0	1.0	0.0	1.0	1.0	0.8	1.0	1.0	0.5	0.5	1.0	9.8	81.7
S51	Niemela et al. [117]	1.0	1.0	0.5	0.0	1.0	0.0	0.6	1.0	0.0	0.5	0.5	0.5	6.6	55.0
S52	Omoronyia et al. [118]	1.0	1.0	1.0	0.0	0.5	0.5	0.4	0.0	1.0	1.0	1.0	1.0	8.4	70.0
S53	Osis et al. [119]	1.0	1.0	1.0	0.0	1.0	0.0	0.2	1.0	0.0	0.0	0.5	0.5	6.2	51.7
S54	Pires [120]	1.0	1.0	1.0	0.0	1.0	1.0	0.6	1.0	1.0	1.0	1.0	0.5	10.1	84.2
S55	Polpinij et al. [121]	1.0	1.0	1.0	0.0	1.0	0.5	0.4	0.0	1.0	0.5	0.0	0.5	6.9	57.5
S56	Rashwan et al. [122]	1.0	1.0	1.0	0.0	1.0	0.5	0.2	0.0	1.0	0.5	0.0	0.5	6.7	55.8
S57	Reinhartz-Berger et al. [123]	1.0	1.0	1.0	1.0	0.5	0.5	0.4	0.0	0.0	0.0	0.5	0.5	6.4	53.3
S58	Riechert et al. [124]	1.0	1.0	1.0	1.0	0.5	0.0	0.6	1.0	0.0	0.5	0.5	0.5	7.6	63.3
S59	Saeki et al. [125]	1.0	1.0	1.0	1.0	0.5	1.0	0.6	1.0	0.0	1.0	0.5	1.0	9.6	80.0
S60	Schugerl et al. [126]	1.0	1.0	1.0	0.0	1.0	0.5	0.6	1.0	0.0	1.0	0.0	0.5	7.6	63.3
S61	Shibaoka et al. [59]	1.0	1.0	0.5	0.0	1.0	0.5	0.8	1.0	1.0	0.5	0.5	0.5	8.3	69.2
S62	Souag et al. [127]	1.0	1.0	1.0	1.0	1.0	0.5	0.6	0.0	1.0	0.5	1.0	0.5	9.1	75.8
S63	Wang et al. [129]	1.0	1.0	1.0	0.0	1.0	1.0	0.4	1.0	0.0	1.0	0.0	0.5	7.9	65.8
S64	Wang et al. [128]	1.0	1.0	1.0	0.0	1.0	0.5	0.4	1.0	0.0	0.0	0.0	0.5	6.4	53.3
S65	Yu et al. [130]	1.0	1.0	1.0	0.0	1.0	0.5	0.4	0.0	0.0	1.0	0.5	0.5	6.9	57.5
S66	Zhang et al. [131]	1.0	1.0	1.0	1.0	1.0	0.5	0.4	0.0	0.0	0.0	0.0	0.5	6.4	53.3
S67	Li et al. [106]	1.0	1.0	1.0	0.0	1.0	1.0	0.6	0.0	0.0	0.0	0.0	0.5	6.1	50.8
	Average	0.96	0.99	0.95	0.37	0.92	0.62	0.43	0.57	0.39	0.62	0.40	0.61	7.82	65.2
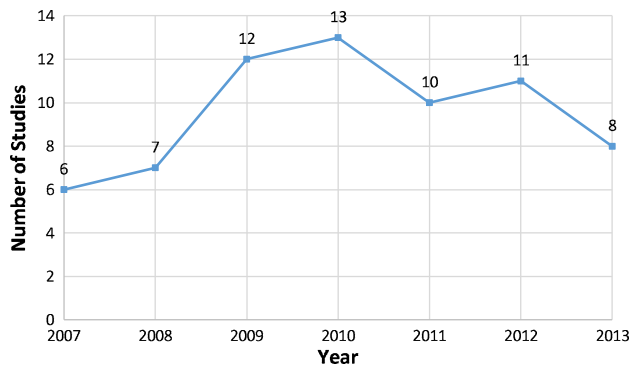
Fig. 2 Temporal view of the studies

of research method published in a certain year. Similarly, in the right part of Fig. 3, the number in a bubble represents the number on a specific research method in a certain application context.

An interesting aspect that we might note in Fig. 3, is that studies are significantly concerned in conducting empirical researches (e.g., case study, quasi-experiment and controlled experiment) on the topic been investigated, i.e., using ontologies in RE, specially in the last five years.

3.2.4 Type of source

The studies included in this review may be of journal, conference, workshop or book chapter publications. The majority of studies are conference papers (56.72 %; 38 studies), followed by journal publications (23.88 %; 16 studies), workshop papers (11.94 %; 8 studies) and book chapter publications (7.46 %; 5 studies).

Table 15 (in "Appendix 2") presents the distribution of selected studies over publication sources, including the publication name, type, count (i.e.,the number of selected studies from each source), and the percentage of selected studies. The 67 selected studies are distributed over 56 publication sources, suggesting that the use of ontologies in RE have been a widespread concern in the research community. As shown in Table 15, the leading venues in this study topic are APSEC, ER, ER workshops, ENASE, RCIS, ICSC, IJSEKE, Managing Requirements Knowledge (MARK) book and OTM workshops. These venues indicate the presence of sources of software engineering, information systems and artificial intelligence areas.

3.3 RQ1: Phases of RE process supported by ontologies

3.3.1 Results

The purpose of this research question was to identify the main phases of the RE process that have been supported by the use of ontologies. We categorized these phases according to the RE process defined by Kotonya and Sommervile [41]: elicitation, analysis and negotiation,

Fig. 3 *Bubble plot* with year of publication, application context and research method dimensions
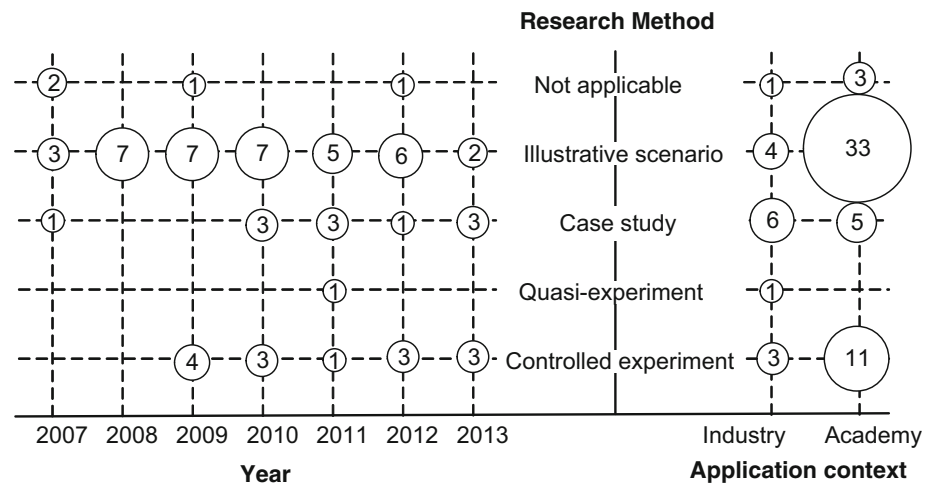
specification, validation and management (see Table 5). The predominant phase that we identified was *Specification* (83.6 %), followed by *Analysis and Negotiation* (58.2 %), *Management* (35.8 %), *Elicitation* (25.4 %) and *Validation* (6 %). It is worth noting that a study could have met more than one phase of the RE process, thus the sum of percentages can be greater than 100 %.

3.3.2 Analysis and discussion

In summary, the results shown in Table 5 indicate that all RE phases are covered by the studies. The *Specification* phase is addressed by more than 80 % of the studies. In fact, to some extent, this result was expected, since ontologies may be used as means to formally specify RE artefacts and activities. *Analysis and Negotiation* phase also encompasses a great number of the studies (more than 50 %). This result is also interesting because it shows that a great part of the studies relied on ontologies to perform some formal requirements analysis, taking advantage of reasoning capabilities provided by ontologies. Furthermore, 26 studies (almost 40 % of papers included) met both *Specification* and *Analysis and Negotiation* phases in the same paper, indicating the interest of using ontology not only to specify some RE artefact or activity but also to conduct some analysis or reasoning.

The *Management* phase is also significantly representative in the included studies. This result was somewhat expected, since ontologies aim to provide explicit conceptualization within a domain knowledge (e.g., requirements knowledge) and allows to automatic reason in such knowledge, thus been potentially useful in the proper management of different kinds of requirements changes.

There is also an interest in the use of ontologies in the *Elicitation* phase aiming to provide some guidance to requirements elicitation realization. For instance, by representing domain knowledge with the aid of ontologies in order to guide the elicitation activity. It is worth noting that most of the papers which met *Elicitation* phase also addressed *Specification* phase, except for S03 and S28.

On the other hand, the use of ontologies in the *Validation* phase is not significant (6 %; four studies). This result may also indicate that the RE researches are less concerned with the research in requirements validation in comparison with other RE research aspects and, hence, also in the use of ontologies in such phase.

Among all 67 studies, only two papers (S01 and S02) addresses all phases of RE process and only two studies met at least four phases of RE process (S50 and S61). In view of the potentials of using ontologies in many phases of RE (as shown in Table 5), this results suggest that the use of ontologies to support the entire RE process needs to be further investigated (see Sect. 4.4).

3.4 RQ2: Software requirements modelling styles

3.4.1 Results

The purpose of this research question was to identify the types of software requirements modelling styles used together with ontologies. The classification of the styles was made after the data extraction of the studies, i.e., during the extraction, plain text data about the software requirements modelling styles used in the study was captured. Next, in the syntheses step of this review, the categories presented in Table 6 were defined according to the distribution of the studies.

Figure 4 depicts the number of studies considering the requirements modelling styles over the RE phases. Note that, the sum of the numbers of studies on specific phases or styles exceeds the total number of studies within a specific category, because one study can use several requirements modelling styles and a study could also have addressed several RE phases.

As shown in Table 6, the predominant style that we identified was *Textual requirements* (49.3 %; 33 studies), followed by *UML* (23.9 %; 16 studies), *Scenario-based* (19.4 %; 13 studies) and *Goal-oriented* (17.9 %; 12 studies). *Features models* category has 6.0% (4 studies), followed by *Non-specific* (4.5 %; 3 studies), *Formal language* (3.0 %; 2 studies), *Aspects* and *Problem frames*, each one with 1.5 % (1 study).

3.4.2 Analysis and discussion

The results of this research question show that we identified a great variety of software requirements modelling styles which are been supported in some extent by ontologies.

Textual requirements category is the most frequent type of RE modelling style addressed by the studies; this category included studies on all RE phases, but the majority of studies are concerned with requirements specification, analysis of textual requirements and management of requirements, as can be seen in Fig. 4. It also encompasses studies that specify requirements documents using a requirements document template (e.g., Software Requirements Specification—SRS). The studies identified in this category frequently involve the use of artificial intelligence techniques, for instance, Information Retrieval (IR), Machine Learning (ML) and Natural Language Processing (NLP).

The *UML* category encompasses studies that use ontology to support the modelling of, for instance, use case models (S07, S14, S22, S44 and S53), activity models (S14, S27, S30, S31 and S43), sequence diagrams (S18, S27 and S31), statecharts models (S23, S27 and S31) and class diagrams (S14, S18 and S54). Furthermore, this category includes studies which address the use of ontologies in four RE phases (see Fig. 4), with an emphasis on specification and analysis phases.

Scenario-based category is also expressive in the studies, it encompasses 13 papers and includes requirements styles such as use case diagrams, misuse cases models (e.g., S18)—for security-specific modelling—and ScenarioML models (e.g., S15). The studies within this category were used in all five RE phases, but with greater frequencies in specification and analysis phases.

Table 5 Use of ontologies in RE phases

Phase	Studies	Count	%
Elicitation	S01, S02, S03, S17, S19, S20, S28, S31, S36, S41, S50, S52, S58, S59, S61, S62, S67	17	25.4
Analysis and Negotiation	S01, S02, S04, S06, S08, S09, S10, S12, S13, S16, S23, S25, S27, S29, S30, S31, S32, S36, S37, S39, S40, S43, S44, S45, S47, S48, S50, S51, S53, S54, S57, S59, S60, S61, S62, S63, S65, S66, S67	39	58.2
Specification	S01, S02, S05, S06, S07, S08, S09, S10, S11, S12, S13, S14, S15, S16, S17, S18, S19, S20, S21, S23, S24, S25, S26, S27, S29, S30, S31, S32, S33, S36, S38, S39, S40, S41, S42, S43, S44, S45, S49, S50, S51, S52, S54, S55, S56, S57, S58, S59, S60, S61, S62, S63, S64, S65, S66, S67	56	83.6
Validation	S01, S02, S37, S48	4	6.0
Management	S01, S02, S04, S06, S08, S11, S13, S14, S22, S26, S29, S34, S35, S38, S42, S43, S44, S46, S47, S51, S54, S60, S61, S64	24	35.8

Table 6 Software requirements modelling styles

Software requirement styles	Studies	Count	%
Textual requirements	S03, S04, S06, S09, S10, S11, S12, S13, S16, S17, S19, S21, S26, S28, S33, S34, S35, S36, S37, S38, S40, S41, S42, S46, S47, S49, S52, S54, S55, S56, S58, S60, S67	33	49.3
UML	S07, S08, S10, S14, S18, S22, S23, S27, S30, S31, S39, S43, S44, S53, S54, S64	16	23.9
Scenario-based	S07, S14, S15, S18, S22, S31, S32, S42, S44, S48, S49, S53, S57	13	19.4
Goal-oriented	S01, S02, S07, S18, S20, S24, S45, S50, S59, S61, S62, S65	12	17.9
Feature models	S05, S22, S25, S63	4	6.0
Non-specific	S29, S51, S66	3	4.5
Business process models	S22, S53	2	3.0
Formal language	S49, S64	2	3.0
Aspects	S10	1	1.5
Problem frames	S32	1	1.5

Fig. 4 Requirements modelling styles over RE phases

TR = Textual requirements   NS = Not specific
UML = Unified modeling language   BPM = Business process models
SC = Scenario-based   FL = Formal languages
GO = Goal-oriented models   AS = Aspects
FM = Feature models   PF = Problem frames
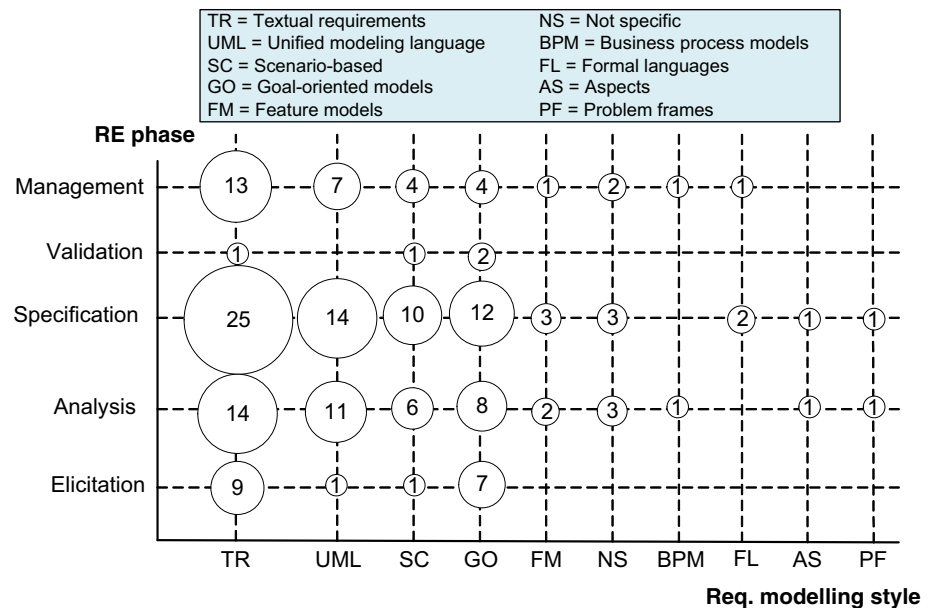
Goal-oriented models category also contains 12 studies and includes papers which use, for instance, the *i** framework (e.g., S07, S18, S20, S24, S62 and S65), the KAOS approach (e.g., S50) and the NFR framework (e.g., S45). The studies within this category addressed, in total, all five RE phases, but with more frequency in the elicitation, analysis and specification phases.

Feature models category is represented by four studies, using ontologies in the context of software reuse (e.g., software product lines). These studies may include the specification, analysis and management of feature models, as shown in Fig. 4. *Non-specific* category includes the studies where we could not identify any specific requirements modelling style. *Business process models*[9] were also addressed by the use of ontologies in two studies which use it in conjunction with use cases in analysis and management of requirements. The use of ontologies also appeared in the included studies along with *Formal languages* (e.g., Z language and EB3), using ontologies to support specification and management of requirements. Lastly, *Aspects* and *Problem frames* modelling styles were also identified within the studies, but with a single study in each category in order to specify and analyse requirements by the use of ontologies.

In summary, we can note that there is a great diversity of RE modelling styles associated with the use of ontologies. We identified ten modelling styles, in which four of them, i.e., the top four frequent studies, are more expressively addressed by the use of ontologies. By contrast, the last categories (non-including the Non-specific category) were

much less satisfied by the included studies. Additionally, as presented in Fig. 4, the specification phase was addressed by almost all modelling styles, except for *Business process models* category. Similarly, the analysis phase was met by almost all styles, except for *Formal languages*. The management phase was also addressed by the great majority of modelling styles, except for *Aspects* and *Problem Frames*. The elicitation phase was only met by the top four styles, especially by the *Textual requirements* and *Goal-oriented* styles. Note that, the validation phase was much less met by the studies (as already discussed in Sect. 3.3), only the *Textual Requirements*, *Scenario-based* and *Goal-oriented* styles addressed such phase.

It is well known in the RE community that natural language requirements are much more prone to inconsistencies and ambiguities. However, even though such fact, the *Textual Requirements* style is the most frequent one used along with ontologies. We may speculate a reason for it. This style is much more used in the Specification phase (see Fig. 4); thus, we can note (Table 6) that the great part of the studies relied exclusively on the capabilities of ontologies to specify requirements, without needing to use a well-defined requirements style. Anyway, this result must be further investigated. In Sect. 4.4, we present some research directions on this topic.

3.5 RQ3: Use of ontologies to describe functional and non-functional requirements

3.5.1 Results

The purpose of this research question was to identify how the use of ontologies is distributed concerning the type of

---

[9] It is worth noting that even though BPM exclusive papers were excluded from this review in the selection process, some papers were included in the review because they used BPM together with other RE modelling styles in the study.

requirement, i.e., functional or non-functional. Thus, we created three categories: (1) functional requirements; (2) non-functional requirements (NFRs); and (3) both types. Table 7 presents the distribution of the studies within these categories.

As seen in Table 7, nearly half of the studies only deal with functional requirements (52.24 %), followed by the studies which deal with both functional and non-functional requirements (44.78 %) and by the studies which only consider non-functional requirement (2.99 %).

3.5.2 Analysis and discussion

The results presented in Table 7 indicate that, in general, functional and non-functional requirements are addressed by the studies. As expected, the great majority of the papers addressed functional requirements (almost 97 %). It is also worth noting that a great number of studies (44.78 %) consider both functional and non-functional requirements (in the same study) on the use of ontologies, showing their concern with a more complete requirements specification. However, 55.23 % of the studies only address one type of requirement, in which 2.99 % are of NFR type.

In fact, even though the considerable amount of papers addressing both types of requirements, the results suggest that the use of ontologies along with non-functional requirements is still underexplored. It is well known that improving NFRs activities contributes to the success development of software systems, leading to software with better quality. Thus, in Sect. 4.4, we present an open question on this topic that might be further investigated.

3.6 RQ4: Existing contributions in ontology-driven RE

3.6.1 RQ4: Results

The purpose of this research question was to identify the existing approaches in ontology-driven RE. The classification of such approaches was based on the analysis of RE problems addressed by the studies. The main RE problems that were identified are depicted in Table 8. The majority of studies addressed the ambiguity, inconsistency and/or incompleteness requirements problem (56.72 %; 38 studies),

followed by the requirements management or evolution problem (35.82 %; 24 studies), domain knowledge representation (26.87 %; 18 studies), integration between requirements and software architecture (4.48 %; three studies). The requirements communication, requirements interoperability, distributed requirements elicitation and goal decomposition problems equally appear with two studies (2.99 %). The goal decomposition (related to Goal-oriented approaches) and selection of elicitation techniques categories are equally represented by 1 study (1.49 %) each. Note that a study could have met more than one RE problem.

In sequel, we briefly depict the objective of each one of the studies within the categories identified in this research question. These RE problems are discussed in RQ7 (Sect. 3.9) along with a discussion about the empirical and non-empirical benefits of using ontologies in RE problems.

P1: Ambiguity, inconsistency and/or incompleteness

As shown in Table 8, this RE problem includes 37 studies which are discussed in the sequel.

S02 proposes an unified Goal-oriented RE (GORE) process, based on a GORE ontology. Such process aims to standardize roles, activities, artefacts and their relationships. This study also addresses the requirements management problem. S04 presents the MUPRET framework (Multiperspective Requirements Traceability) which deploys ontology as a knowledge management (P2) mechanism to intervene mutual understanding without restricting the freedom in expressing requirements differently. S05 proposed a Semantic Web-oriented language, called Semantic Annotations for Feature Modelling Description Language that provides the means to semantically describe feature models. This framework uses ontology to provide a standard terminology (i.e., a shared conceptualization) of feature models.

S07 proposes a conceptual ontology-driven approach to reduce the heterogeneities between user requirements formalisms, i.e., specifying a common terminology and also to facilitate the interoperability (P6). S08 describes a methodology for requirements specification that aims to produce models for functional requirements that can be automatically validated for completeness and consistency. This methodology is part of the Requirements Driven

Table 7 Functional and non-functional requirements distribution

Types of requirements	Studies	Count	%
Functional req. only	S03, S06, S08, S09, S11, S12, S13, S14, S16, S18, S21, S22, S23, S25, S26, S27, S29, S30, S31, S32, S33, S34, S37, S39, S41, S43, S47, S48, S52, S53, S57, S58, S63, S66, S67	35	52.24
Non-functional req. only	S45, S51	2	2.99
Both	S01, S02, S04, S05, S07, S10, S15, S17, S19, S20, S24, S28, S35, S36, S38, S40, S42, S44, S46, S49, S50, S54, S55, S56, S59, S60, S61, S62, S64, S65	30	44.78

Table 8 Existing contributions in ontology-driven RE

#	RE problem	Studies	Count	%
P1	Ambiguity, inconsistency and/or incompleteness	S02, S04, S05, S07, S08, S09, S10, S12, S13, S16, S17, S18, S19, S21, S22, S24, S25, S27, S30, S31, S32, S37, S39, S40, S42, S43, S45, S47, S48, S49, S50, S55, S56, S57, S63, S65, S66, S67	38	56.72
P2	Requirements management/evolution	S01, S02, S04, S06, S08, S11, S13, S14, S19, S22, S26, S29, S34, S35, S38, S42, S43, S44, S46, S47, S51, S54, S60, S64	24	35.82
P3	Domain knowledge representation	S17, S19, S20, S29, S33, S36, S37, S41, S49, S50, S52, S53, S59, S61, S62, S64, S67	18	26.87
P4	Integration between requirements and architecture	S15, S45, S51	3	4.48
P5	Requirements communication	S13, S54	2	2.99
P6	Requirements models interoperability	S07, S23	2	2.99
P7	Distributed requirements elicitation	S03, S58	2	2.99
P8	Goal decomposition	S59, S61	1	1.49
P9	Selection of elicitation technique	S28	1	1.49

Design Automation framework (RDDA). The contribution of the S09 paper is an ontology-based approach for defining a good Software Requirement Specification (SRS). The main component of this approach is an ontology for conceptualizing the SRS called OntoSRS whose main objective is to guide stakeholders in the definition of a SRS with the following properties: consistency, unambiguity, correctness, organization and traceability. By contrast, the goal of the S10 work is to integrate (defining a common terminology) AOSD concepts, classic RE notions, and the new standard ISO/IEC 25030 on software quality requirements. The main result of this study is the REASQ (REquirements, Aspects and Software Quality) conceptual model, expressed in UML.

The S12 paper proposes a tool-based framework that is capable of supporting the security requirements specification process. The aim of such study is to assist the requirements engineer in the process of security specifications so that their quality can be enhanced and the effort reduced. S13 presents an approach for hazard identification (HazId) based on requirements and reuse-oriented safety analysis. The approach offers a starting point for the identification of potential system safety concerns from the RE phase of development, to facilitate communication of requirements (P5) and also to manage such requirements (P2). S16 presents a checking method of elicited software requirements using requirements ontology. In the S17 paper, the authors use a requirements ontology as domain knowledge (addressing the P3 problem) for requirements elicitation. They introduce a checking method using requirements ontology to detect errors in requirements. S18 proposes a vulnerability-centric modelling ontology, which aims to integrate empirical knowledge of vulnerabilities into the system development process.

S19 presents the prototype of a semantic guidance system used to assist requirements engineers with capturing requirements using a semi-formal representation. The semantic guidance system uses concepts, relations and axioms of a domain ontology to provide a list of suggestions; the requirements engineer can build on to define requirements. Such study addresses the P1, P2 and P3 problems. The S21 paper outlines a stepwise methodology to discover and understand the multidimensional correlations among regulatory security requirements and relies on a RE framework to explain regulatory requirements and related domain concepts from natural language documents and represent them using ontological domain modelling techniques. The resulting Problem Domain Ontology (PDO) establishes the semantics of each requirement through its relationships with domain concepts in a socio-technical environment. S22 developed a Knowledge-assisted Ontology-based Requirements Evolution (K-RE) method and tool which combine the social software principles and semantic web concepts to enable a knowledge-assisted RE (also addressing the P2 problem).

S24 proposes the use of UFO (Unified Foundational Ontology) to provide an ontological foundation to the i* framework. In such paper, one particular construct, means-end links, is taken as subject of research. The S25 paper presents an approach to deal with inconsistencies in feature models (FM) evolution scenarios reducing the impact of change to the smallest part of an FM that changes. Such study formalizes FMs from an ontological perspective and define constraints that must be satisfied in FMs to be consistent. S27 introduces a method of ontology-based semantic verification of UML behavioural models. The S30 paper proposes a hybrid method which aims to detect behavioural requirement interactions in order to improve the quality of software system. The proposed method includes

two main parts: an ontological modelling process for requirement specification and a set of conflict detection rules derived from behavioural analysis.

The aim of the S31 paper is to develop an ontology that diminishes the complexity, difficulty and ambiguity of understanding the different concepts of modelling notations used in software RE. S32 uses the concept of projection from relational algebra and combines it with concepts from the problem frames (PF) and scenario-based approaches to present a conceptual model for conducting problem projection in RE. S37 introduces means of requirements validation, extracting domain-specific abstractions (also addressing P3 problem) and relationships between requirements from documents which are assembled into an ontology that acts as a conceptual model of the problem domain. Then, such descriptions are taken, in the form of scenarios or use case descriptions, and used to construct a set of Message Sequence Charts (MSCs). The S39 paper designed and implemented an approach as an ontology-based component in a requirements specification tool, named TESSI, that helps an analyst to write a UML model and, hence, to improve it and to reduce misunderstandings.

In the S40 paper, an ontology-based framework for representing, storing and reusing security requirements is presented. This framework combines a risk analysis ontology, based on methods of risk analysis and security standards, and a requirements ontology. By contrast, the objective of S42 is to propose the use of a basic ontology based on the artefacts to standardize and manage (also addressing P2) software requirements that can be used on any project developed in any organization. The S43 paper presents the CDADE tool, which can help designers to detect conflicts in activity diagram evolution (also addressing P2 problem). The CDADE tool is composed of ontologies, metadata, and conflict detection rules. S45 proposes to address fragment identification as a problem best served with interactive aids and presents a faceted exploration approach to explore NFR solutions and identify reusable model fragments where NFRs and trade-offs are represented as ontologies. Such paper is also concerned with the integration between requirements and software architecture (P4).

In the S47 paper, its proposed the OntRep, an automated ontology-based reporting approach for requirements categorization, conflict analysis and tracing (P2) based on ontologies and semantic reasoning mechanisms. The S48 paper introduces an approach to systematically extract knowledge from system requirements to construct different views of ontologies for the system as a part of a comprehensive framework to analyse and validate software requirements and design. Such study also addresses the requirements management problem (P2). S49 proposes a

multi-representation ontology to solve conflicts and problems of the multi-context requirements specification. Such study also uses ontologies for domain knowledge representation (P3). The S50 paper proposes a goal-directed RE process, named KBRE model, that centralizes domain knowledge and semantics of requirements. Using such model, requirements can be elicited, elaborated, and inconsistencies and other related requirements problems can be detected.

The S55 paper proposes a method for simplifying ambiguity of requirement specification documents through two concepts of ontology-based probabilistic text processing: Text classification and Text Filtering. Such study also address the P2 problem. S56 proposes an automated analysis of SRS (Software Requirements Specification) documents for different NFR types and contains two significant contributions towards this goal: a corpus containing annotations for different NFR types, based on a requirements ontology and a support vector machine (SVM) classifier to automatically categorize requirements sentences into different ontology classes. In the S57 paper, a framework for classifying external variability types based on ontological principles is introduced. The framework defines the external view of software in terms of the behaviour of the application domain. Behaviour is formalized as state changes in response to external stimuli.

The S63 paper presents an approach for modelling and verifying feature diagrams using Semantic Web OWL ontologies. In the S65 paper, the authors argue that a social ontology at the core of a RE process can be the basis for integrating security into a requirements driven software engineering process. For this, they describe the i* framework and show how it can be used to model and reason about security concerns and responses. Similarly to other studies in the P1 category, the S66 paper also proposes to represent a requirement specification through a requirement ontology that is described by Description Logics. Last but not least, S67 suggests a formal approach to precisely describe ontology using Description Logics, and then model the integrity rules and derivation rules which restrict the business behaviour. This study also met the P3 problem.

P2: Requirements management/evolution

This RE problem includes 24 studies which are described hereafter.

In the S01 paper, a quality-driven RE framework and a tool that applies knowledge management techniques and quality ontologies to support RE activities are presented. The ontology implements the quality characteristics and metrics prescribed by the ISO/9126 quality model, providing a common vocabulary to address quality concerns/aspects across RE activities. The S02 also addresses the requirements management problem through the proposal of

an unified GORE process, as it was explained in the previous section. In the S04 paper, the MUPRET framework explained in P1, is used to trace requirements, i.e., been useful to manage requirements. S06 presents an approach for manipulation of data involved in the SW-FMEA (Software—Failure Modes and Effects Analysis) process, introducing an ontological model which formalizes concepts involved in the analysis.

In the S08 paper, the RDDA framework (explained in the previous section) is to used to manage the consistency of requirements. S11 address the difficulty for end-users in finding documentation related to software requirements. Thus, an architecture is proposed for enhanced resources search, combining the strengths of the social (social annotations) and semantic (semantic metadata) technologies. The S13 paper, as already explained in P1, proposes an approach to manage requirements related to security requirements. S14 developed a software development process support tool, using ontologies, to promote the discovery of software artefacts relationships, mixing the knowledge offered by the business ontology and the software engineering ontology.

The S19 paper, as already explained in the previous section, assists the management of requirements by the use of a semantic guidance system. Also explained in previous section, the S22 paper uses the knowledge-assisted ontology-based requirements evolution (K-RE) method and tool to address requirements management. S26 specifies the domain knowledge as ontologies (also meeting P3) capturing concepts and relations of the problem domains. Such study uses these ontologies to detect code fragments corresponding to the requirements sentences, providing traceability between requirements and code. The goals of S29 paper are: to recover high-level artefacts such as requirements and design models; to simplify modification of those models; to reuse software design and domain knowledge contained within models; and to integrate those models with enhancements via a novel combination of ontological and model weaving concepts.

The S34 paper presents a conceptualization of a requirement management system. The system (referenced as the CARM—Computer Aided Requirements Management framework) makes use of semantic networks by expressing requirements through ontologies. In the S35 paper, the authors extend previous work on NFR ontology by proposing a change management mechanism for tracing the impact of NFRs on the other constructs in the ontology such as FR or NFR operationalization and vice versa, and providing a traceability mechanism. The S38 paper applies the ontology-driven RE methodology (OntoREM) to a case study within the aerospace context in order to evaluate the potential of such ontology-driven approach in knowledge-driven RE.

As explained in the previous section, the S42 paper uses a basic RE ontology that can be used to manage requirements in an organization. The S43 paper, as also previously explained, uses the CDADE tool, to detect conflicts in activity diagram management/evolution. S44 study proposes an ontology-based blog to detect and to manage the conflicts using extended use case models between change requests and the existing system requirements. In the S46 paper, the Infrastructure for semantic document management (ISDM) is used to support requirements evolution.

The OntRep, proposed in the S47 paper and already explained in previous section, also aims at lowering the effort for requirements management, while keeping high requirements consistency by using ontologies. The S51 paper aids the NFR requirements management by allowing to define metrics for quality attributes as ontologies, to specify execution qualities as profiles according to a quality variability model and ontologies, and to model quality properties as an integrated part of software architecture. Such study also relates requirements and software architecture (P4). In the S54 work, the authors propose the use of an iterative and incremental model-driven RE process (maintaining traceability of models) combined with the employment of different notations such as controlled natural language and ontology in several activities of RE process.

In the S60 research, the authors focus on assessing nonfunctional quality requirements, specifically evolvability, through semantic modelling of relevant software artefacts and introduce the SE-Advisor that supports the integration of knowledge resources typically found in software ecosystems by providing a unified ontological representation. Finally, in the S64 paper, a methodology with tool support is proposed, aiming to manage requirements, for automatically deriving ontological metadata from formal software models and semantically describe them.

P3: Domain knowledge representation

As shown in Table 8, this RE problem includes 18 studies which are described hereafter.

As previously explained, the S17 paper uses a requirements ontology as domain knowledge to aid requirements elicitation. Similarly, the S19 paper—also explained in P1—uses concepts, relations and axioms of a domain ontology to provide a list of suggestions the requirements engineer can build on to define requirements. In the S20 paper, a mechanism which integrates domain knowledge in domain-independent RE languages is proposed.

The S26 paper proposes a technique for recovering the links using domain ontologies and uses such knowledge to manage requirements, as already explained. The S29 paper, which also addresses the P1 problem, reuse software design and domain knowledge contained within models via

combination of ontological and model weaving concepts. S33 proposes a method and a tool to enhance an ontology of domain knowledge for requirements elicitation by using Web mining. Similarly, the S36 paper proposes the usage of a domain ontology as domain knowledge during the requirements elicitation process and a supporting tool based on this technique.

The S37 paper, as explained above, uses domain-specific abstractions and relationships between them extracted from a document or documents that are representative of the application domain that are assembled in ontologies. In the S41 paper, the authors propose an ontology-based approach for requirement elicitation in process centred problem domain. In this method, based on upper level requirement representation ontology, the study merges the ontology acquiring process and the requirement elicitation process together. The S49 paper, which addresses the P1 problem, makes use of a multi-representation domain ontologies.

In the S50 paper, domain knowledge is represented by the use of ontologies in a goal-directed RE process, as described in P2. The S52 work also investigates an approach for building domain ontologies suitable for guiding requirements elicitation. S53 proposes to analyse use cases by means of natural language processing (NLP) and rules can be defined for validating use cases against a given domain ontology. In the S59 study, a method for goal-oriented security requirements analysis using security domain knowledge ontologies which is derived from several security targets compliant to common criteria is presented. Such study also uses ontologies to support goal decomposition, i.e., the P8 problem. In a similar way, the S61 paper proposes a method called GOORE where a domain knowledge ontology is utilized to support goal decomposition.

The S62 proposes a method that exploits security and domain ontologies dynamically through a collection of heuristic production rules. In the S64 paper, a methodology is proposed to automatically derive domain ontological metadata from formal software models and semantically describe them, which is also used for requirements management, as explained previously. The S67 paper suggests an approach to describe domain ontology to be used for consistency checking, as explained in the P1 section.

P4: Integration between requirements and architecture

This kind of RE problem was addressed by three studies, two of them were already explained, S45 and S51, which met, respectively, the P1 problem and P2 problem. S45 presents an ontology-based search approach to identify valuable NFR knowledge fragments. The search outputs can be organized by valuable knowledge fragments. Architects can choose which kind of classification criteria and knowledge fragments they are interested in. S51 allows to define metrics for quality attributes as ontologies, to specify execution qualities as quality profiles according to a variability model and ontologies, and to model quality properties as an integrated part of software architecture. By contrast, the S15 paper proposes an approach that maps domain events, classes, and individuals used in requirements scenarios (described using ScenarioML) to architectural components (described using xADL).

P5: Requirements communication

This RE problem category was met by two studies: S13 and S54. Both studies were explained previously, whereas S13 addresses the P1 and P2 RE problems and the S54 work meet the P2 problem. Particularly, the approach proposed in S13 captures the knowledge contained in both the requirements document and previously documented HazOp projects in order to attain a reduction in the cost of safety analysis by using established technologies such as ontology, case-based reasoning and natural language processing. The S54 also adopts ontologies and controlled natural language to aid activities of knowledge acquisition and knowledge representation. It provides a notation close to the client native language, which is fully understandable by both requirements engineer and client, thus facilitating the comprehension of and validation of the requirements specification.

P6: Requirements models interoperability

This RE problem category was met by two studies: S07 and S23. The S07 proposes a conceptual ontology-driven approach to facilitate interoperability and to reduce the heterogeneities between user requirements formalisms. It uses a pivot model allowing the integration of different semi-formal models. This work also addresses the P1 problem. In the S23 paper, the authors use OWL to represent statechart models which must be interoperated with other statechart models.

P7: Distributed requirements elicitation

This category is represented by the S03 and S29 studies. S03 proposes an ontology-based framework for global requirements elicitation aiming to reduce misunderstandings among stakeholders, and therefore helps to achieve more committed requirements. Conversely, the S58 paper presents a classification of Wiki-based approaches to RE and introduces the ontology of an approach that aims at supporting the collaboration of stakeholders in order to enable large stakeholder groups to elicit, semantically structure and classify requirements.

P8: Goal decomposition

This category was addressed by two studies: S59 and S61. Both studies also met the P3 problem and were explained

in the P3 section. The S59 paper uses ontology to suppport goal decomposition in the context of goal-oriented security requirements analysis. The S61 proposes the GOORE method which is also used to support goal decomposition.

P9: Selection of elicitation technique

This RE problem category is only met by the S28 study. This study developed an ontology and explored how well it supports requirements elicitation and elicitation techniques selection.

3.6.2 RQ4.1: Types of contributions results

The purpose of this research question was to identify the types of existing contributions which were explained above. The classification of such approaches was based on the work by Petersen et al. [55] and includes the method, model, tool and process categories (see Table 9). We only consider as process contributions, papers that use ontology in all five RE phases (elicitation, analysis and negotiation, specification, validation and management). Whereas the method contributions include the papers that presented some ontology-based approach. Note that, similarly to other research questions, this question also allows a study to be included in more than one category. The predominant contribution that we identified was *Method* (88.1 %; 59 studies), followed by *Model* (76.1 %; 51 studies), *Tool* (56.7 %; 38 studies) and *Process* (3.0 %; 2 studies).

3.6.3 RQ4.1: Analysis and discussion

The results shown in Table 9 indicate that the four types of contributions considered in this research question are contemplated by the studies. Different kinds of *Method* contributions of using ontologies in some RE activity are proposed by more than 80 % of the studies. A *Model* contribution is proposed by more than 75 % of the studies, showing that the great majority of studies model some RE activity by the use of ontologies. Furthermore, 56.7 % of the studies provide some kind of *Tool* support for their contributions; this finding was somewhat positively surprising since it demonstrates the concern of studies with the automatization of their contributions and also to be effectively used by potential users. Only two studies (S01 e S02) presented *Process* contributions, this information is complementary to the fact that only the same studies met all phases of RE process, as stated in RQ1 (see Sect. 3.3).

It is worth highlighting that 29 studies propose in the same paper the three main types of contribution: model, method and tool; it indicates that these studies use ontologies to model some RE aspect and to also perform some manipulation in such model aided by a tool support.

Moreover, the S01 paper proposes in the same study a model, a process and a tool to meet the entire RE process.

3.7 RQ5: Ontology-related languages used in RE

3.7.1 Results

The purpose of this research question was to identify the main ontology-related languages that have been used to support RE activities. The initial set of categories was defined according to the semantic web technology recommendations of the World Wide Web Consortium (W3C) [61] (e.g., OWL, XML, RDF, SPARQL and so on). However, during the data extraction of the studies, others languages were found and inserted as categories to help to answer this research question. The distribution of studies within these categories is presented in Table 10.

OWL (Web Ontology Language) is used by the majority of studies (49.3 %; 33 studies) in the included papers, followed by *SPARQL* query language (10.4 %; 7 studies), *Description Logics* (7.5%; 5 studies) and *SWRL* rule language (4.5 %; 3 studies). *RDF*, *Foundational Ontology* and *UML*[10] have, each one, two studies (3 %). Whereas, *XML*, *XSPARQL*, *First-order Logic* and *Relational model-based* categories have 1 study (1.5 %) each one. We encountered also papers where we could not identify a specific ontology-related language (37.3 %; 25 studies) or they did not use any language (i.e., propose some kind of taxonomy), the so-called *Non-specific* category. They were, however, included in the review as they provided data points relevant to the other research questions. Note that, in this research question, a study could also have used more than one ontology language.

3.7.2 Analysis and discussion

The results of this research question are analysed and discussed from the viewpoint of adopting the languages to specify and handle ontologies recommended by the consortium that administers the Web (W3C). Among the categories used to classify the studies (see Table 10), *OWL*, *SPARQL*, *XSPARQL*, *SWRL*, *RDF*, *XML* are recommended by the stack of web semantic technologies of the W3C, whereas *Description logics*, *Foundational ontology*, *UML*, *First-order logic* and *Relational model-based* are not. The *Non-specific* category is also inserted as a not recommended technology, since each paper classified in this

---

[10] Note that the *UML* category in this research is different from the UML category of RQ2. In this question UML modelling styles are modelled by the use of ontologies, by contrast, in RQ2, UML (e.g., class diagram) is used as a language to represent ontologies.

Table 9 Type of contributions on ontology-driven RE

Types of contributions	Studies	Freq.	%
Method	S03, S04, S05, S06, S07, S08, S11, S12, S13, S14, S15, S16, S17, S18, S19, S20, S21, S22, S23, S25, S26, S27, S29, S30, S32, S33, S34, S35, S36, S37, S38 S39, S40, S41, S43, S44, S45, S46, S47, S48, S49, S50, S51, S52, S53, S54, S55, S56, S57, S58, S59, S60, S61, S62, S63, S64, S65, S66, S67	59	88.1
Model	S01, S04, S05, S06, S07, S08, S09, S10, S11, S12, S13, S14, S16, S18, S19, S20, S21, S22, S23, S24, S25, S26, S27, S28, S29, S31, S32, S36, S38, S39, S40, S41, S42, S43, S44, S45, S50, S51, S54, S55, S57, S58, S59, S60, S61, S62, S63, S64, S65, S66, S67	51	76.1
Tool	S01, S04, S05, S06, S07, S08, S12, S13, S14, S16, S17, S19, S21, S22, S25, S26, S33, S34, S35, S36, S38, S39, S43, S44, S45, S47, S48, S49, S50, S51, S53, S54, S58, S59, S60, S61, S63, S64	38	56.7
Process	S01, S02	2	3.0

Table 10 Ontology-related languages

Ontology-related languages	Studies	Count	%
OWL (W3C)	S01, S05, S06, S07, S08, S09, S10, S11, S12, S13, S19, S22, S23, S27, S30, S34, S38, S39, S40, S42, S45, S46, S47, S48, S50, S51, S53, S54, S56, S58, S60, S63, S64	33	49.3
SPARQL (W3C)	S05, S06, S09, S11, S40, S45, S56	7	10.4
Description Logics	S23, S27, S50, S66, S67	5	7.5
SWRL (W3C)	S27, S30, S39	3	4.5
RDF (W3C)	S22, S58	2	3.0
Foundational Ontology	S24, S57	2	3.0
UML	S18, S28	2	3.0
XML (W3C)	S14	1	1.5
XSPARQL (W3C)	S05	1	1.5
First-order Logic	S04	1	1.5
Relational model-based	S35		1.5
Non-specific	S02, S03, S15, S16, S17, S20, S21, S25, S26, S29, S31, S32, S33, S36, S37, S41, S43, S44, S49, S52, S55, S59, S61, S62, S65	25	37.3

category used arbitrary ways to handle ontologies within the study.

Concerning the use of W3C recommended technologies, it can be observed that 34 studies used one or more of these recommended languages. By contrast, 35 studies used at least one non-recommended language. It is worth noting that two studies (S27 and S50) used languages of both categories, i.e., used languages of the W3C recommended languages and one language of the set of non-recommended categories in the same study. Specifically, S27 used *OWL* and *SWRL* together with *Description logics* and S50 used *OWL* also together with *Description logics*.

It can be noted that there is a great concern in the studies to use W3C recommended languages to specify and manipulate ontologies. However, we stress that although OWL is the most expressive and broadly accepted ontology language, it was only used by almost half of the included studies in this review. Thus, one can state that many studies which propose to use ontologies in RE process are not

following the proper guidelines and standard technologies to specify and handle formal ontologies.

3.8 RQ6: Reuse of ontologies

3.8.1 Results

The purpose of this research question was to identify the studies which reused ontologies developed in others studies in the context of RE. The distribution of studies within these categories is presented in Table 11. It is worth noting that the answer to this question was also considered as a quality criterion (see Sect. 3.1).

As shown in Table 11, the great part of the studies (66 %; 44 papers) did not reuse any ontologies, i.e., they specified their own ontology. By contrast, several papers (34 %; 23 studies) reused ontologies in their contributions. In the following, we depict the ontologies which were reused by the studies (within the *Yes* category).

Table 11 Reuse of ontologies

Reuse of ontologies	Studies	Count	%
No	S01, S03, S05, S07, S08, S09, S10, S12, S13, S15, S17, S18, S19, S20, S22, S23, S25, S26, S27, S28, S30, S31, S34, S37, S40, S41, S42, S44, S47, S48, S49, S50, S51, S52, S53, S54, S55, S56, S60, S61, S63, S64, S65, S67	44	66
Yes	S02, S04, S06, S11, S14, S16, S21, S24, S29, S32, S33, S35, S36, S38, S39, S43, S45, S46, S57, S58, S59, S62, S66	23	34

3.8.2 Analysis and discussion

A wide variety of ontologies is reused by the 23 studies identified in Table 11. We grouped these ontologies according to their purposes (Table 12).

We identified six studies which reuse *generic requirements engineering ontologies*, i.e., which represent general aspects within RE. The S04 study reuses the ontology for RE and software development [5]. S29 uses the Ontology for Software Specification and Design (OSSD) [33]. S39 uses the ontology of requirements [36]. The S46 study reuses the Software Requirements Ontology [22]. S58 uses the SWORE (SoftWiki Ontology for RE) [42] and S66 reused the requirements ontology proposed by [1].

Furthermore, five studies used *domain knowledge ontologies* for RE purposes. In the S21 paper, the Problem Domain Ontology (PDO) [44] is reused. S33 and S36 uses the ORE (Ontology-based Requirements Elicitation) [37]. The S39 paper reuses the domain ontology (besides reusing an ontology of requirements) proposed by [36] and S59 reuses the Security Ontology in an Application Domain (SOAD) [59].

Four of the twenty-three studies reuse some type of *foundational ontology* within their contribution. In the S24

Table 12 Groups of ontologies reused

Ontology purpose	Studies
Generic RE ontology	S04, S29, S39, S46, S58, S66
Domain knowledge ontology	S21, S33, S36, S39, S59
Foundational ontology	S24, S29, S62, S57
Security ontology	S06, S59
Functional req. ontology	S16
Goal-oriented ontology	S02
Open educational resources-CC ontology	S11
Business ontology	S14
Scenario-extended problem ontology	S32
Non-functional req. ontology	S35
Ontology-driven requirements engineering methodology	S38
Speech act ontology	S43
Design rationale (NDR) ontology	S45

paper, the UFO ontology [29] is reused. Besides reusing a generic ontology, the S29 paper also uses the Standard Upper Merged Ontology (SUMO) [50]. The S62 paper reuses an upper ontology proposed in [7] and, lastly, the S57 paper reuses the Bunge's ontology [62].

In addition to these groups, two studies focused on the reuse of *security ontologies*: S06 and S59. The first reuses the ontology specified in [16] and the last reuses the SOAD ontology [59]. By contrast, one single study reused a *functional requirements ontology*, S16. Such paper reuses the functional requirements ontology for an Education Management System (EMS) [18].

Moreover, S02 reuses the *Goal-oriented requirements engineering (GORE) ontology* [4]. S11 reuses open educational resources (OER)-CC ontology [56]. S14 reuses the *Business ontology* [8]. S32 reuses the *scenario-extended problem ontology* [11]. In the S35 paper, the *NFRs ontology* [38] is reused. The S38 paper reuses the ontology-driven RE methodology (OntoREM) [40]. S43 reuses the speech act ontology [45]. Finally, S45 reuses the design rationale (NDR) ontology [46].

3.9 RQ7: Ontologies benefits to the RE process

In order to answer this research question, we firstly synthesize (Sect. 3.9.1) and analyse (Sect. 3.9.2) the research method of the studies along with the studies' indication of positive or negative results of the use of ontologies. Then, we discuss the empirical and non-empirical benefits (Sect. 3.9.3) identified in this review.

3.9.1 Results

The purpose of this research question was to gather and classify evidences to state how the use of ontologies benefits requirements engineering. The classification of the studies was defined according to the presence or absence of empirical evaluation in the study and by the positive or negative indication that using ontologies benefits RE activities. The defined categories are as follows: positive with empirical evaluation, positive without empirical evaluation, negative with empirical evaluation and negative without empirical evaluation (see Table 13). No study was included in the *Negative without empirical evaluation* category.

Positive without empirical evaluation (61.2 %; 41 studies) constitutes majority of the studies, i.e., studies that did not empirically evaluated their contributions. This category is followed by *Positive with empirical evaluation* (37.3 %; 25 studies) and *Negative with empirical evaluation* (1.5 %; 1 study).

3.9.2 Analysis and discussion

The results of this research question are of outmost importance to verify whether there are evidences to state that the use of ontologies effectively benefits RE process. These results show that the majority of the studies provide only a positive argumentation about the benefits of the use of ontology in the RE process, i.e., they did not present any empirical evidence, but rather presented some line of argumentation and/or illustrated their contributions using small examples. However, there was a significant amount of studies that presented empirical evidences of such benefits. In the following, we analyse and discuss these specific studies from two dimensions: (1) type of empirical evidence, determined by the research method used in the study and (2) context on which the study was conducted, academic or industry.

As shown in Table 13, most of these empirical studies used controlled experiments (14 papers), followed by case studies (11 papers) and a quasi-experiment (1 study). Moreover, looking up in the studies which performed controlled experiments, we can observe that 3 of them (S13, S22 and S52) reported evidences from industry, whereas 10 studies report evidences from an academic context. By analysing the empirical evidences through the use of case studies, it can be observed that six studies are from industry and five studies academic. The quasi-experiment study was also conducted in industry settings.

The empirical studies can also be analysed over the RE phases that they met. Figure 5 presents a bubbleplot chart distributed over three dimensions: study settings, research method and RE phases. The left part in Fig. 5 denotes the relationship between empirical studies' settings and the phase that they met. The number in a bubble represents the number of studies on a specific setting addressing a certain phase. By contrast, the right part of Fig. 5 denotes the relationship between studies' research methods and RE phases. The number in a bubble represents the number of studies on a specific empirical research method in a certain RE phase. Note that, the sum of the numbers of studies on a specific research method or settings category exceeds the total number of studies within a specific category, because a same study could meet more than one RE phase.

By analysing Fig. 5, we can draw some important information about the empirical studies identified in this review: (1) the elicitation phase is only and equally addressed by case studies and controlled experiments, with more emphasis on academic studies; (2) the analysis phase is slightly more addressed by case studies than by controlled experiments and is not met by quasi-experiment studies. Moreover, the majority of studies within this phase are academic; (3) the specification phase is the top addressed phase by empirical studies, with a greater contribution from controlled experiment studies than from case studies papers, no quasi-experiment studies met this phase. In addition, the settings of the papers in this phase are in great part academic, but with significant presence of industrial papers; (4) the validation phase is only addressed by two academic case studies; and (5) the majority of studies in management phase are case studies, followed by controlled experiments and a single quasi-experiment study. Furthermore, the great part of studies was conducted in industrial settings.

On the other hand, it was found only one study that presented negative results about the use of ontologies in RE. S03 proposed a framework for global requirements elicitation (P7 problem; RQ4) in the context of Global Software development (GSD). However, after the conduction of a controlled experiment, the results did not present a significant increase in the quality of elicited requirements by the use of ontologies in comparison with other approaches without the use of ontology. These results did not coincide with the paper author's expectations which may have been caused by the planning and conduction of the experiment, as explained by the authors.

Table 13 Evidences of ontologies benefits to the RE process

Evidences	Studies			Freq.	%
Positive without empirical evaluation	S02, S05, S06, S07, S08, S09, S10, S11, S14, S15, S18, S19, S20, S23, S24, S27, S28, S29, S30, S31, S32, S34, S39, S40, S41, S43, S45, S46, S48, S49, S51, S53, S57, S58, S59, S60, S63, S64, S65, S66, S67			41	61.2
Positive with empirical evaluation	Case study	Controlled exp.	Quasi-exp.	25	37.3
	S01, S04, S21, S37, S38, S42, S44, S47, S50, S61, S62	S12, S13, S16, S17, S22, S25, S26, S33, S36, S52, S54, S55, S56	S35		
Negative with empirical evaluation	S03			1	1.5
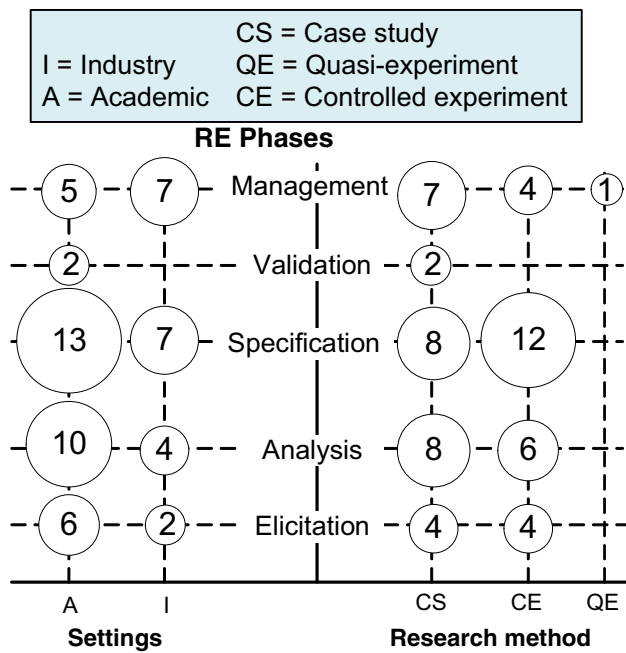
Fig. 5 Research method and context of empirical studies over RE phases

In the following subsection, we analyse and discuss the main benefits, according to the RE problems explained in Sect. 3.6, that are been supported by the use of ontologies in the empirical and non-empirical studies included in this review.

3.9.3 Empirical and non-empirical benefits

Positive results reported in the studies are classified according to the research method that was conducted in the study. Figure 6 illustrates the empirical benefits (i.e., the results presented by the studies which used some empirical evaluation) and the non-empirical benefits, i.e., the results reported by the studies which did not conduct any empirical evaluation in the study.

Figure 6 presents the group of benefits which were categorized according to the RE problems categories presented in Sect. 3.6. Analysing Fig. 6, we can note that the "Reduce ambiguity, inconsistency and/or incompleteness" (B1), "Aid requirements management" (B2), "Domain knowledge representation for guiding requirements elicitation" (B3), "Facilitate requirements communication" (B5) and "Support goal decomposition" (B8) are the empirical benefits that we identified. By contrast, the non-empirical benefits include these benefits (except for B5) and also encompass the "Integration between requirements and software architecture" (B4), "Distributed requirements elicitation" (B7) and "Aid selection of elicitation technique" (B9). In the cases where the benefit is reported by both empirical and non-empirical studies (e.g., B1, B2, B3

and B8), the distribution of the studies is very similar; the more expressive difference appears in the B1 benefit which contains 24 non-empirical studies and 14 empirical studies. Note that, such benefit is reported by the majority of studies, followed by B2, B3, B4, B5, B6, B8, B7 and B9.

Concerning the empirical studies—which, in fact, characterize evidences that the use of ontologies benefits RE—we identified five groups: B1, B2, B3, B5 and B8. The first group refers to the studies S04, S12, S13, S16, S21, S22, S25, S37, S42, S50, S55 and S56 and suggests that ontologies may be very useful for standardization and understanding of requirements (reducing ambiguity), for automatic error checking and conflict analysis of requirements (verifying inconsistency and incompleteness of requirements with less effort) and so on. The second group (S17, S33, S36, S52, S61 and S62 studies) presented evidences that ontologies are good to represent domain knowledge for guiding requirements elicitation and hence to produce requirements specifications with better quality. The studies (S04, S13, S26, S38, S42, S44, S47) in the third group showed that using ontologies is effective for managing requirements changes in general and NFR in specific (e.g., S01 and S35 studies). The fourth grouped is represented by the S13 and S54 studies which shows that the use of ontology may facilitate the communication between stakeholders (e.g., customer, development team, etc) during the RE process. The last group is specific to the S61 study and uses ontologies to support the goal decomposition in the context of goal-oriented models.

Additionally, we also analyse these empirical benefits according to their application context: industry and academic. Figure 7 presents a bubbleplot chart distributed over the empirical benefits and evidence/context dimensions. The number in a bubble represents the number of studies on a specific empirical benefit in a certain pair of research method and application context. As shown in Fig. 7, the B1 benefit is mostly reported by academic controlled experiments (six papers) and by industrial case studies (four papers). Moreover, the B2 benefit is addressed by case study, controlled experiment and quasi-experiment research methods in both industrial and academic settings. The B3 benefit is mainly met by academic controlled experiments (four papers) and academic case studies (three papers). The B5 benefit is reported by studies which conducted controlled experiments in academy and industry. Finally, the B8 benefit was met by a single academic case study.

In summary, we can highlight that, despite the concept of ontology is not widespread in the RE community, the results found in this study suggest that there are empirical evidences to state that ontologies benefit RE activities in both academic and industrial settings. However, the strength of these evidences is somewhat limited to the context (for instance, artefact used, activity addressed by the study,

B1=Reduceambiguity, inconsistency or/and incompletenes   B6=Support req.models interoperability
B2=Aidreq.management/evolution   B7=Distributed requirement selicitation
B3=Domain knowledge representation to guide elicitation   B8=Support goal decomposition
B4=Integration between req. and swarchitecture   B9=Aid selection of elicitation technique
B5=Facilitate req. communication
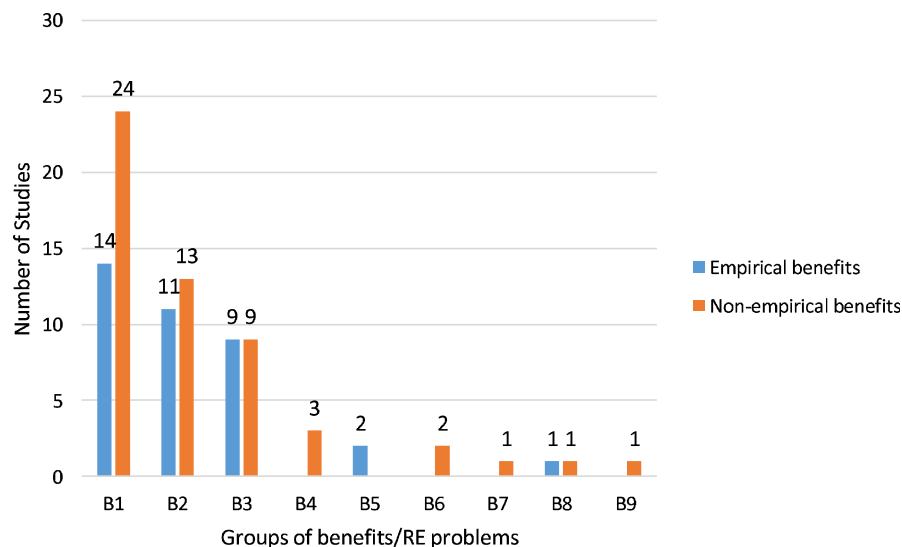
Fig. 6 Empirical and non-empirical benefits of using ontologies in RE

ontology language used, RE phase(s) addressed and so on) on which the studies were performed.

4 Discussion

This section starts with a discussion about the scope of this review. Next, we summarize and discuss some related works and depict the threats to validity of this work. Then, we finish discussing further research on the use of ontologies in RE.

4.1 Scope of this systematic review

This SLR focuses on how ontologies are employed in RE, in terms of their use in the RE process, the software requirements modelling styles been used, the types of requirements, the existing contributions and their types, the ontology-related languages, reuse of RE ontologies and empirical and non-empirical benefits. When conducting this SLR, we use jointly the terms "ontology" and "requirements engineering" as part of the search terms to maximize coverage of potentially-relevant studies retrieval and consequently ensure that the results of this SLR can cover all studies that use ontologies in RE activities.

This paper focuses exclusively on the use of ontologies in RE. The "ontology" term has a special meaning in computer science, since it carries with it a whole body of

knowledge that is different, for instance, from conceptual model. For this reason, we do not consider synonyms for ontologies in the paper. Furthermore, as explained in Sect. 2.4, this SLR focuses on ontology-driven RE studies in the context of software engineering with generic and technology/paradigm independent contributions, thus we did not consider contributions for specific domains and also for SOA, AOSE, BPM and IT areas.

4.2 Related works

There are some studies that investigate the use of ontologies in Software Engineering [3, 23, 30, 57] and the use of ontologies in RE [9].

Ruiz and Hilera [57] presents a taxonomy in order to assist Software Engineering and Technology (SET) professionals and researchers on the use of ontologies in software development. They present an overview of ontology's use in many fields of software engineering such as software maintenance, software quality, software measurement, system modelling, component-based software engineering, requirement engineering and so on. Some proposals of the use of ontologies at development time and at runtime are also described in their work.

Hapel and Seedorf [30] present some examples of ontology applications throughout the Software Engineering lifecycle. They discuss the advantages of ontologies in each case and provide a framework for classifying the usage of
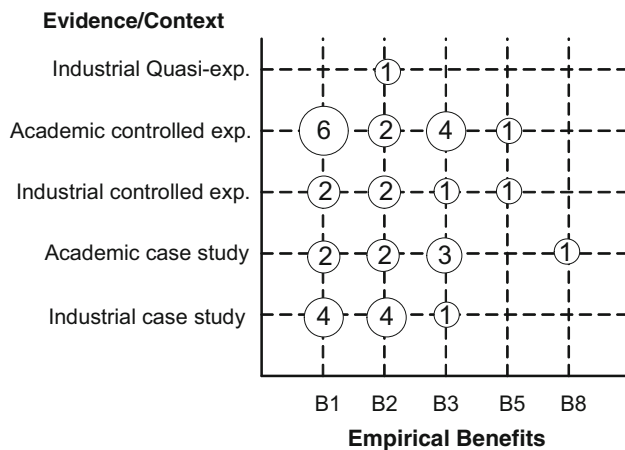
Evidence/Context

Fig. 7 Empirical benefits over evidence/context

ontologies in Software Engineering. In a similar way, Ahmed [3] describes some ontology-based approaches in software engineering (ordered by their position in the development life cycle). Examples of techniques from the entire software engineering life cycle (e.g., analysis and design, deployment and run-time, and maintenance) are presented in his work.

Gasevic et al. [23] define a framework that identifies places in software lifecycle where ontologies can contribute to improve software engineering practices. They apply the framework to analyse the use of ontologies in different phases of software life cycle.

Regarding the use of ontologies in RE, Castañeda et al. [9] presents a classification of approaches that include ontologies within RE and describes the expected benefits and challenges that must be faced when using ontologies in RE activities.

As explained previously, despite the existence of some works that surveyed works jointly using ontologies in RE and software engineering, to the best of our knowledge, there is no previous systematic investigation of the literature on the topic. In fact, we have presented a systematic review on the use of ontologies in RE [ref. omitted for blind review] before, but that review considered a limited number of research questions (five) and provided a shallow analysis of the results. Thus, the surveys that were identified and our first review on the topic do not capture all the aspects and evidences that we are interested, for instance, types of requirements been covered, RE problems been addressed, reused ontologies, empirical benefits of using ontologies in RE and so on.

4.3 Threats to validity

This section describes concerns that must be improved in future replications of this study and other aspects that must be taken into account in order to generalize the results of the

SLR performed in this work. In order to organize this section, the threats to validity were classified using the Internal, External, Construct and Conclusion categories [64].

Construct validity: The main constructs in this review are the two concepts "ontologies" and "requirements engineering". For the first concept, we use the term "ontology" and its synonym "ontologies" to make sure that all selected studies are related to ontology approaches. For the second concept, the terms "requirements", "requirements engineering", "software analysis" and "domain analysis" are used to ensure high coverage of potentially-relevant studies on the influence to RE activities from database search. A complementary manual search was not performed in the SLR due to the fact there are no conferences and journals specifically focused on the joint use of these concepts. This threat is mitigated by including the general intervention term "ontologies" along with "requirements" in the terms for the search in seven reputable database.

As a threat to the internal validity, some subjective decisions may have occurred during paper selection and data extraction since some primary studies did not provide a clear description or proper objectives and results, making difficult the objective application of the inclusion/exclusion criteria or the impartial data extraction. In order to minimize selection and extraction mistakes, the selection process was performed in an iterative way and the data extraction was realized collaboratively by reviewers, and any conflicts were discussed and resolved by all the authors. In this way, we tried to mitigate the threats due to personal bias on study understanding. It is also worth noting that the first and second authors are, respectively, PhD and MSc students in RE and the remaining four authors are experienced researchers with expertise in RE, Ontology or Software Engineering.

External validity is concerned with establishing the generalizability of the SLR results, which is related to the degree to which the primary studies are representative for the review topic. In order to mitigate external threats, the search process described in Sect. 2.5 was defined after several trial searches and validated with the consensus of all authors. We tested the coverage and representativeness of retrieved studies, including automatic database search and references scan.

Conclusion validity: It is possible that some studies excluded in this review should have been included. To mitigate this threat, the selection process and the inclusion and exclusion criteria were carefully designed and discussed by authors to minimize the risk of exclusion of relevant studies. Furthermore, in the final round of study selection, reviewers conducted the selection process in parallel and independently, and then harmonized their selection results to mitigate the personal bias in study selection caused by individual reviewers. It is worth

highlighting that we specified the time period of published studies for this SLR from January 2007 to October 2013, when we started this SLR. As mentioned in Sect. 2, for the best of our knowledge, there is currently no SLR on the use of ontologies in RE. Ruiz and Hilera made a general survey about the use of ontologies in software engineering published before 2007 [57]. In order to reduce repetitive effort and make use of existing work, we set the starting time of the published studies included in this SLR to January 2007.

4.4 Further research

This SLR has generated several promising research directions that are very important but underexplored in current research and practice:

1. How can we employ ontologies to improve the entire RE process (**RQ1**)? This research has not received much attention and the claims lack evidence support. In fact, only two studies address all five phases of the RE process. Moreover, only one (S01) was empirically evaluated by the conduction of an academic case study.

2. Why ontologies have not been much used along with the validation phase (**RQ1**)? This research includes the investigation about for the lack of use of ontologies in the validation phase, since such RE phase was rarely addressed by the studies. We suspect that might be some resistance on the use of ontologies when it is need to deal with stakeholders. However, some empirical investigation (e.g., survey) on the topic is encouraged in order to understand why it occurs.

3. How can ontologies be better used in requirements modelling (**RQ2**)? This SLR showed that ontologies are much more used along with textual requirements. In fact, textual requirements are inherently more prone to inconsistencies and ambiguities than well-structured requirements modelling languages, such scenario-based and GORE approaches. The effort required to reason with textual requirements is greater – generally involving the use of AI techniques. Thus, the use of ontologies can be also further explored along with non-textual requirements modelling styles.

4. How can ontologies be more used with NFRs (**RQ3**)? This research has not received sufficient attention by the studies. It is well known in the RE community that NFRs are specially important to the success of software systems, but it was addressed by less than half of the studies included in this review. The use of ontologies may be further explored in the elicitation, specification, analysis, validation and management of non-functional requirements both in academic and industrial settings.

5. How can ontologies improve the integration between requirements and architecture (**RQ4**)? This research has not received much attention by the studies and the claims lack evidence. Only three studies addressed this RE problem and none of them empirically evaluated their contributions. The use of ontologies in such problem appears to be potentially useful, because it can support different aspects which can favour the interplay between requirements and architecture. For instance, ontologies could be used to represent requirements and architectural knowledge and to establish traceability links between these knowledge as well as to support reasoning.

6. How can ontologies facilitate the communication during the RE process (**RQ4**)? This research has also received insufficient attention by the studies. In fact, only two studies addressed the RE communication problem and were also empirically evaluated (reporting positive claims). The knowledge conceptualization and sharing capabilities inherent to ontologies (especially when it is used some implementation language, for example OWL) suggest that the communication problem between RE players can be better explored with the aid of ontologies.

7. How can ontologies improve requirements models interoperability (**RQ4**)? As explained in RQ4 (Sect. 3.6), this research was addressed by only two studies. However, since ontologies provide a shared conceptualization in a specific domain, it can also be further explored to promote the interoperability between requirements models. The use of ontologies can also be explored to interoperate requirement models and to help to generate code in the context of model-driven development (MDD).

8. How can ontologies improve distributed requirements elicitation (**RQ4** and **RQ7**)? This research involves requirements elicitation in distributed settings, i.e., where the RE actors are geographically distant. Similarly to items (4), (5) and (6), this research has not received much attention by the studies included in this review. Besides that, as explained in Sect. 3.9.2, one paper found a non-expected empirical negative result on the use of ontologies in distributed requirements elicitation. In this sense, additional empirical evaluations are also required in this kind of RE research problem in order to verify whether such result remains valid.

9. How can W3C ontology-related technologies be better used in RE (RQ5)? The RQ5 results presented in Sect. 3.7 showed that nearly half of the studies have not been supported by W3C ontology-related technologies. In fact, the use of these technologies to support different activities of the RE process still

needs to be further explored. For instance, the OWL and RDF languages can be used to guide requirements elicitation. Furthermore, with the use of SWRL and SPARQL languages, some analysis, validation and management could also be performed.

10. How can we promote wider reuse of RE ontologies (RQ6)? As shown in Table 11 (Sect. 3.8), few works have reused ontologies specified by other authors. Thus, more researches are required in order to define agreed conceptualizations within the RE community regarding several RE aspects (e.g., NFRs ontology, GORE ontology and so on) aiming to promote the effective reuse of RE ontologies and avoiding the construction of redundant ontologies.

11. How to measure the costs and benefits of using ontologies in RE in a qualitative or quantitative way (RQ7)? The benefit identified in this SLR is classified without qualitative or quantitative information. Hence, we need more research on the qualitative or quantitative measurement of the benefits of using ontologies in RE. Regarding the costs of using ontologies in RE, we could not explicitly identify metrics to assess the impact that the use of ontologies brings to the RE process. Thus, there is also necessary to conduct more researches on the qualitative and quantitative measurement of the costs of using ontologies in RE.

5 Conclusions

In this work, we conducted a SLR to investigate the use of ontologies in RE. Our goal was to improve the understanding of how ontologies support RE as well as to identify evidences of its use in this field.

Sixty-seven studies were finally included, in which five main RE process phases, nine requirements modelling styles, nine main RE problems, eleven ontology-related languages, twenty-five ontologies have been reused by the studies, five groups of empirical benefits from twenty-six empirical studies and seven groups of non-empirical benefits from forty-two non-empirical studies were identified.

Three categories of benefits out of nine distinct groups deserve special attention, since they were present in the majority of evidences found in the empirical studies: (1) reduction of ambiguity, inconsistency and/or incompleteness of requirements; (2) support for domain knowledge representation to guide requirements elicitation; and (3) assistance in requirements management/evolution. The review results also suggest that ontologies can potentially be useful to deal with several RE problems such as, integration between requirements and software architecture,

requirements communication, requirements models interoperability, distributed requirements elicitation and goal decomposition in the context of GORE approaches.

Among the five RE phases, the validation phase is the one which was less addressed by the use of ontologies. Moreover, the majority of studies only partially addresses the RE process. In fact only two papers considered all five phases. There is a great diversity of RE styles that have been supported by ontologies, highlighting the top four frequent categories. Furthermore, even though being much more prone to inconsistencies and ambiguities, textual requirements category is the top frequent modelling style found in this review. Majority of contributions also addressed functional requirements, but there is also a significant number of papers that covers both functional and non-functional requirements. In general, papers mainly addressed ambiguity, inconsistency and incompleteness of requirements, management/evolution of requirements as well as domain knowledge representation.

Additionally, a great number of contributions are supported by a tool, indicating the studies' concerns with automated support for potential users. Moreover, half of the studies followed W3C recommendations on ontology-related languages, in which the use of OWL was much more significant. Regarding the reuse of ontologies, despite the existence of a great variety of RE ontologies, none was identified as being broadly adopted. Finally, it was observed that the empirical benefits are mainly focused on the top three RE problems highlighted above. However, it is worth noting that the requirements communication and goal decomposition problems are also considered in the set of studies.

The results presented in this systematic review can be very useful to the RE community, since it gathers evidences from the primary studies included in the review, forming a body of knowledge regarding the use of ontologies in RE. As future work we intend to further investigate some of the research directions presented in this paper, with an emphasis on the integration between requirements and software architecture through the use of ontologies with the aid of W3C ontology-related languages. Moreover, we intend to continue this systematic review to explore how ontologies have been supporting the entire software development process.

Appendix 1: Extraction form

See Table 14.

Table 14 Extraction form

#	Study data	Description	Relevant RQ
1	Study identifier	Unique id for the study	Study overview
2	Date of data extraction		Study overview
3	Authors, Year, Title, Country		Study overview
4	Article source		Study overview
5	Type of article	Journal, conference, workshop, book chapter	Study overview
6	Application context	Industrial, academic	Study overview
7	Research method (based on [19])	Controlled experiment, case study, survey, ethnography, action research, illustrative scenario, not applicable	Study overview
8	Req. phase (based on [41])	What were the RE phases addressed by the study? (Elicitation, analysis and negotiation, specification, validation and management)	RQ1
9	Software requirement style	What were the software requirements modelling styles addressed in the study?	RQ2
10	Type of requirement	What were the types of requirements addressed in the study?	RQ3
11	Name of contributions	What are the contributions names of the study?	RQ4
12	Objectives	What were the objectives of the study?	RQ4
13	Type of contribution (based on [55])	What were the types of contributions of the study? (Method, tool, process, model)	RQ4.1
14	Ontology language	What were the ontology representation languages used in the study? (OWL, RDF, RDF Schema and so on)	RQ5
15	Reused ontologies	What were the reused ontologies by the study?	RQ6
16	Evidences	What were the evidences which indicate that the use of ontologies benefits the requirements engineering? (Negative argumentation, negative with empirical evaluation, positive argumentation, positive with empirical evaluation)	RQ7
17	Benefits	What were the main benefits?	RQ7

Appendix 2: Publication sources

See Table 15.

Table 15 Distribution of studies over publication sources

Publication source	Type	Count	%
Asia Pacific Software Engineering Conference (APSEC)	Conference	4	6.0
Conceptual Modelling—ER	Conference	2	3.0
ER Workshops	Workshop	2	3.0
International Conference on Evaluation of Novel Approaches to Software Engineering (ENASE)	Conference	2	3.0
International Conference on Research Challenges in Information Science (RCIS)	Conference	2	3.0
International Conference on Semantic Computing (ICSC)	Conference	2	3.0
International Journal of Software Engineering and Knowledge Engineering (IJSEKE)	Journal	2	3.0
Managing Requirements Knowledge	Book	2	3.0
OTM Workshops	Workshop	2	3.0
A Handbook of Ontologies	Book	1	1.5
ACM Transactions on Software Engineering and Methodology	Journal	1	1.5
Annual Computer Software and Applications Conference (COMPSAC)	Conference	1	1.5
Annual Computer Software and Applications Conference Workshops (COMPSACW)	Workshop	1	1.5
Annual IEEE Systems Conference (SysCon)	Conference	1	1.5
Architecting Dependable Systems V	Book	1	1.5

Table 15 continued

Publication source	Type	Count	%
Asia-Pacific Services Computing Conference (APSCC)	Conference	1	1.5
Conference of the Center for Advanced Studies on Collaborative Research (CASCON)	Conference	1	1.5
Expert Systems	Journal	1	1.5
Expert Systems with Applications	Journal	1	1.5
Extended Semantic Web Conference (ESWC)	Conference	1	1.5
IASTED International Conference on Software Engineering	Conference	1	1.5
ICSE Workshop on Wikis for Software Engineering (WIKIS4SE)	Workshop	1	1.5
Iet Software	Journal	1	1.5
Information Systems Frontiers	Journal	1	1.5
Integrating Security and Software Engineering: Advances and Future Visions	Book	1	1.5
International Conference on Advanced Information Systems Engineering (CAiSE)	Conference	1	1.5
International Conference on Advances in Semantic Processing (SEMAPRO)	Conference	1	1.5
International Conference on Computational Collective Intelligence (ICCI)	Conference	1	1.5
International Conference on Computational Science (ICCS)	Conference	1	1.5
International Conference on Computer Safety, Reliability, and Security (SAFECOMP)	Conference	1	1.5
International Conference on Computer Supported Cooperative Work in Design (CSCWD)	Conference	1	1.5
International Conference on ENTERprise Information Systems (CENTERIS)	Conference	1	1.5
International Conference on Formal Engineering Methods (ICFEM)	Conference	1	1.5
International Conference on Global Software Engineering	Conference	1	1.5
International Conference on Information Reuse and Integration	Conference	1	1.5
International Conference on Information Science and Technology (ICIST)	Conference	1	1.5
International Conference on Knowledge Engineering and Management by the Masses (EKAW)	Conference	1	1.5
International Conference on Knowledge Science, Engineering and Management (KSEM)	Conference	1	1.5
International Conference on Quality Software (QSIC)	Conference	1	1.5
International Conference on Signal-Image Technology and Internet-Based Systems (SITIS)	Conference	1	1.5
International Conference on Software and Data Technologies (ICSOFT/ENASE)	Conference	1	1.5
International Conference on Software Engineering and Knowledge Engineering (SEKE)	Conference	1	1.5
International Journal of Software Engineering and Its Applications	Journal	1	1.5
International Symposium of the International Council on Systems Engineering (INCOSE)	Conference	1	1.5
International Symposium on Leveraging Applications of Formal Methods, Verification and Validation (ISoLA)	Conference	1	1.5
Journal of Object Technology (JOT)	Journal	1	1.5
Journal of Research and Practice in Information Technology (JRPIT)	Journal	1	1.5
Journal of Web Semantics: Science, Services and Agents on the World Wide Web (JWS)	Journal	1	1.5
Knowledge and Information Systems (KIS)	Journal	1	1.5
Knowledge-Based Systems (KBS)	Journal	1	1.5
Monterey Workshop	Workshop	1	1.5
Requirements Engineering Journal (REJ)	Journal	1	1.5
Requirements Engineering: Foundation for Software Quality (REFSQ)	Conference	1	1.5
Software Quality Journal (SQJ)	Journal	1	1.5
Workshop on Requirements Engineering (WER)	Workshop	1	1.5
World Academy of Science, Engineering and Technology	Journal	1	1.5

References

1. Abran A, Cuadrado J, Garca-Barriocanal E, Mendes O, Snchez-Alonso S, Sicilia M (2006) Engineering the ontology for the swebok: issues and techniques. In: Calero C, Ruiz F, Piattini M (eds) Ontologies for software engineering and software technology. Springer, Berlin, pp 103–121

2. Achimugu P, Selamat A, Ibrahim R, Mahrin MN (2014) A systematic literature review of software requirements prioritization research. Inf Softw Technol 56(6):568–585

3. Ahmed E (2008) Use of ontologies in software engineering. In: SEDE, pp 145–150

4. Anwer S, Ikram N (2006) Goal oriented requirement engineering: a critical study of techniques. In: Software engineering

conference, 2006. APSEC 2006. 13th Asia Pacific. IEEE, pp 121–130

5. Assawamekin N, Sunetnanta T, Pluempitiwiriyawej C (2008) Automated multiperspective requirements traceability using ontology matching technique. In: SEKE. Knowledge Systems Institute Graduate School, pp. 460–465

6. Baader F, Calvanese D, McGuinness DL, Nardi D, Patel-Schneider PF (eds) (2003) The description logic handbook: theory, implementation, and applications. Cambridge University Press, New York

7. Bjorner D (2010) Role of domain engineering in software development-why current requirements engineering is flawed!. In: Pnueli A, Virbitskaite I, Voronkov A (eds) Perspectives of systems informatics, vol 5947. Lecture notes in computer science. Springer, Berlin, pp 2–34

8. Cappelli C, Baião F, Santoro F, Iendrike H, Lopes M, Nunes V (2007) Uma abordagem de construção de ontologia de domínio a partir do modelo de processos de negócio

9. Castañeda V, Ballejos L, Caliusco ML, Galli MR (2010) The use of ontologies in requirements engineering. Glob J Res Eng 10(6):2–8

10. Chen L, Babar MA, Zhang H (2010) Towards an evidence-based understanding of electronic data sources. In: Proceedings of the 14th international conference on evaluation and assessment in software engineering, EASE'10, British Computer Society, Swinton, pp 135–138

11. Chen X, Jin Z, Yi L (2007) An ontology of problem frames for guiding problem frame specification. In: Zhang Z, Siekmann J (eds) Knowledge science, engineering and management, lecture notes in computer science, vol. 4798. Springer, Berlin, pp 384–395. doi:10.1007/978-3-540-76719-0_38

12. Corazzon R Notes on the history of ontology, descriptive and formal ontology: a resource guide to contemporary research. www.formalontology.it

13. Davis R, Shrobe H, Szolovits P (1993) What is a knowledge representation? AI Mag 14(1):17

14. Dermeval D, Vilela J, Bittencourt II, Castro J, Isotani S, Brito P (2014) A systematic review on the use of ontologies in requirements engineering. In: Brazilian symposium on software engineering (SBES)

15. Ding W, Liang P, Tang A, van Vliet H (2014) Knowledge-based approaches in software documentation: a systematic literature review. Inf Softw Technol 56(6):545–567

16. Dittmann L, Rademacher T, Zelewski S (2004) Performing fmea using ontologies. In: 18th international workshop on qualitative reasoning, Evanston, pp 209–216

17. Dyb T, Dingsyr T (2008) Empirical studies of agile software development: a systematic review. Inf Softw Technol 50(9–10):833–859

18. Dzung DV, Ohnishi A (2009) Ontology-based reasoning in requirements elicitation. In: 2009 seventh iEEE international conference on software engineering and formal methods, pp 263–272

19. Easterbrook S, Singer J, Storey MA, Damian D (2008) Selecting empirical methods for software engineering research. In: Shull F, Singer J, Sjberg D (eds) Guide to advanced empirical software engineering. Springer, New York, pp 285–311

20. El-Attar M, Miller J (2012) Constructing high quality use case models: a systematic review of current practices. Requir Eng 17(3):187–201

21. Elsevier (2014) Information and software technology journal. http://www.journals.elsevier.com/information-and-software-technology/. Accessed 28 March 2014

22. Falbo R, Nardi J (2008) Evolving a software requirements ontology. In: XXXIV Conferencia Latinoamericana de Informática, Santa Fe, Argentina, pp 300–309

23. Gasevic D, Kaviani N, Milanovic M (2009) Ontologies and software engineering. In: Staab S, Studer R (eds) Handbook on ontologies, international handbooks on information systems. Springer, Berlin, pp 593–615

24. Gazevic D, Djuric D, Devedzic V, Selic B (2006) Model driven architecture and ontology development. Springer, Secaucus

25. Gómez-Pérez A, Fernández-López M, Corcho O (2007) Ontological engineering: with examples from the areas of knowledge management, e-commerce and the semantic web. In: Advanced information and knowledge processing. Springer, Secaucus

26. Gruber TR (1993) A translation approach to portable ontology specifications. Knowl Acquis 5(2):199–220

27. Gruber TR (1995) Toward principles for the design of ontologies used for knowledge sharing. Int J Hum Comput Stud 43(5–6):907–928

28. Guarino N (1998) Formal ontology in information systems: Proceedings of the 1st international conference June 6–8, 1998, Trento, Italy, 1st edn. IOS Press, Amsterdam

29. Guizzardi G (2005) Ontological foundations for structural conceptual models. CTIT, Centre for Telematics and Information Technology

30. Happel HJ, Seedorf S (2006) Applications of ontologies in software engineering. In: 2nd international workshop on semantic web enabled software engineering (SWESE 2006), held at the 5th international semantic web conference (ISWC 2006. Citeseer)

31. Hernandes EM, Zamboni A, Fabbri S, Thommazo AD (2012) Using GQM and tam to evaluate start—a tool that supports systematic review. CLEI Electron J 15(1):1–13

32. Horrocks I, Patel-Schneider PF, Boley H, Tabet S, Grosof B, Dean M (2004) Swrl: a semantic web rule language combining owl and ruleml. W3c member submission, World Wide Web Consortium

33. Hoss A, Carver D (2006) Ontological approach to improving design quality. In: 2006 IEEE aerospace conference, p 12

34. Huhns M, Singh M (2005) Service-oriented computing: key concepts and principles. Internet Comput IEEE 9(1):75–81. doi:10.1109/MIC.2005.21

35. Ivarsson M, Gorschek T (2009) Technology transfer decision support in requirements engineering research: a systematic review of REj. Requir Eng 14(3):155–175

36. Janetzko R (2009) Applying ontology for checking of requirements specification. Master's thesis, Faculty of Computer Science, TU Chemnitz

37. Kaiya H, Saeki M (2006) Using domain ontology as domain knowledge for requirements elicitation. In: 14th IEEE international conference, requirements engineering, pp 189–198. doi:10.1109/RE.2006.72

38. Kassab M, Ormandjieva O, Daneva M (2009) An ontology based approach to non-functional requirements conceptualization. In: Fourth international conference on software engineering advances, 2009. ICSEA '09, pp 299–308

39. Kitchenham B, Charters S (2007) Guidelines for performing systematic literature reviews in software engineering. In: Technical report EBSE 2007–001, Keele University and Durham University Joint Report

40. Kossmann M, Odeh M (2010) Ontology-driven requirements engineering-a case study of ontoREM in the aerospace context. In: 20th Annual International Symposium of the International Council on Systems Engineering, vol 2., INCOSE 2010IL, Chicago, pp 1000–1012

41. Kotonya G, Sommerville I (1998) Requirements engineering—processes and techniques. Wiley, New York

42. van Lamsweerde A (2001) Goal-oriented requirements engineering: a guided tour. In: Proceedings of fifth IEEE international symposium on requirements engineering, 2001, pp 249–262

43. LAPES (2014) Start-state of the art through systematic review tool. http://lapes.dc.ufscar.br/tools/start_tool. Accessed October 2013

44. Lee SW, Gandhi RA (2006) Requirements as enablers for software assurance. CrossTalk J Def 19(12):20–24

45. Liu CL (2009) Ontology-based requirements conflicts analysis in activity diagrams. In: Computational science and its applications-ICCSA 2009. Springer, New York, pp 1–12

46. López C, Cysneiros L, Astudillo H (2008) NDR ontology: Sharing and reusing NFR and design rationale knowledge. In: First international workshop on managing requirements knowledge, 2008. MARK '08, pp 1–10

47. Mahdavi-Hezavehi S, Galster M, Avgeriou P (2013) Variability in quality attributes of service-based software systems: a systematic literature review. Inform Softw Technol 55(2):320–343. Special Section: Component-Based Software Engineering (CBSE), 2011

48. McGuinness DL, Harmelen FV (2004) Owl web ontology language overview. http://www.w3.org/TR/owl-features/

49. Mika P, Akkermans H (2004) Towards a new synthesis of ontology technology and knowledge management. Knowl Eng Rev 19:317–345

50. Niles I, Pease A (2001) Towards a standard upper ontology. In: Proceedings of the international conference on formal ontology in information systems, vol 2001. FOIS '01, ACM, New York, pp 2–9

51. Ouhbi S, Idri A, Fernández-Alemán JL, Toval A Requirements engineering education: a systematic mapping study. Requir Eng 1–20

52. Pacheco C, Garcia I (2012) A systematic literature review of stakeholder identification methods in requirements elicitation. J Syst Softw 85(9):2171–2181. Selected papers from the 2011 Joint working IEEE/IFIP conference on software architecture (WICSA 2011)

53. Pan JZ, Staab S, Aßmann U, Ebert J, Zhao Y (eds) (2013) Ontology-driven software development. Springer, Berlin

54. Pérez J, Arenas M, Gutierrez C (2009) Semantics and complexity of SPARQL. ACM Trans Database Syst 34:16:1–16:45

55. Petersen K, Feldt R, Mujtaba S, Mattsson M (2008) Systematic mapping studies in software engineering. In: Proceedings of the 12th international conference on evaluation and assessment in software engineering. EASE'08, British Computer Society, Swinton, pp 68–77

56. Piedra N, Chicaiza J, López J, Martínez O, Caro E (2010) An approach for description of open educational resources based on semantic technologies. In: 2010 IEEE education engineering (EDUCON), pp 1111–1119

57. Ruiz F, Hilera J (2006) Using ontologies in software engineering and technology. In: Calero C, Ruiz F, Piattini M (eds) Ontologies for software engineering and software technology. Springer, Berlin, pp 49–102

58. Selby RW (2007) Software engineering: Barry W. Boehm's lifetime contributions to software development, management, and research (practitioners). Wiley-IEEE Computer Society Pr, Canada

59. Shibaoka M, Kaiya H, Saeki M (2007) GOORE: goal-oriented and ontology driven requirements elicitation method. Springer, Berlin, pp 225–234

60. Siegemund K, Thomas EJ, Zhao Y, Pan J, Assmann U (2011) Towards ontology-driven requirements engineering. In: Workshop semantic web enabled software engineering at 10th international semantic web conference (ISWC), Bonn

61. W3C (2014) W3c semantic web activity. http://www.w3.org/2001/sw/. Accessed April 2014

62. Wand Y, Weber R (1995) On the deep structure of information systems. Inform Syst J 5(3):203–223

63. White S (2004) Business process modeling notation (bpmn) version 1.0. technical report, BPMI.org. http://www.bpmn.org/Documents/BPMN%20V1-0%20May%203%202004.pdf

64. Wohlin C, Runeson P, Höst M, Ohlsson MC, Regnell B, Wesslén A (2000) Experimentation in software engineering: an introduction. Kluwer Academic Publishers, Norwell

65. Wooldridgey M, Ciancarini P (2001) Agent-oriented software engineering: The state of the art. In: Ciancarini P, Wooldridge M (eds) Agent-oriented software engineering, vol 1957. Lecture Notes in Computer Science, vol 1957. Springer, Berlin, pp 1–28

66. Yue T, Briand LC, Labiche Y (2011) A systematic review of transformation approaches between user requirements and analysis models. Requir Eng 16(2):75–99

Systematic Literature Review References

67. Balushi Al, TH, Sampaio PRF, Loucopoulos P (2013) Eliciting and prioritizing quality requirements supported by ontologies: a case study using the ElicitO framework and tool. Expert Syst 30(2):129–151

68. Anwer Shahzad Ikram N (2008) A process for goal oriented requirement engineering. In: Proceedings of the IASTED international conference on software engineering, SE 2008. Innsbruck, Austria, pp 255–261

69. Aranda GN, Vizcaino A, Piattini M (2009) Analyzing ontology as a facilitator during global requirements elicitation. In: Proceedings of 2009 4th IEEE international conference on global software engineering, ICGSE 2009. Limerick, Ireland, pp 309–314

70. Assawamekin N, Sunetnanta T, Pluempitiwiriyawej C (2010) Ontology-based multiperspective requirements traceability framework. Knowledge and Information Systems 25(3):493–522

71. Bagheri E, Asadi M, Ensan F, Gašević D, Mohabbati B (2011) Bringing semantics to feature models with safmdl. In: Proceedings of the 2011 conference of the center for advanced studies on collaborative research, CASCON '11. IBM Corp, Riverton, pp 287–300

72. Bicchierai I, Bucci G, Nocentini C, Vicario E (2012) An ontological approach to systematization of SW-FMEA. In: Ortmeier F, Daniel P (eds) Computer safety, reliability, and security, vol 7612. Lecture notes in computer science. Springer, Berlin, pp 173–184

73. Boukhari I, Bellatreche L, Jean S (2012) An ontological pivot model to interoperate heterogeneous user requirements. In: Margaria S, Steffen B (eds) Leveraging applications of formal methods, verification and validation. Applications and case studies, vol 7610. Springer, Berlin, Heidelberg, pp 344–358

74. Cardei I, Fonoage M, Shankar R (2008) Model based requirements specification and validation for component architectures. In: 2nd annual IEEE systems conference, 2008, pp 1–8

75. Castañeda V, Ballejos L, Caliusco ML (2012) Improving the quality of software requirements specifications with semantic Web technologies. In: WER 2012–15th workshop on requirements engineering. CIDISI, UTN, FRSF, Lavaise 610, Santa Fe, Argentina

76. Castillo I, Losavio F, Matteo A, Boegh J (2010) REquirements, aspects and software quality: the REASQ model. J Object Technol 9(4):69–91

77. Chicaiza J, Lopez J, Piedra N, Martinez O, Tovar E (2010) Usage of social and semantic web technologies to design a searching architecture for software requirement artefacts. IET Softw 4(6):407–417

78. Daramola O, Sindre G, Moser T (2012) Ontology-based support for security requirements specification process. In: Herrero P,

Panetto H, Meersman R, Dillon T (eds) On the Move to Meaningful Internet Systems: OTM 2012 Workshops, vol 7567. Springer, Berlin, Heidelberg, pp 194–206

79. Daramola O, Stålhane T, Omoronyia I, Sindre G (2013) Using Ontologies and machine learning for hazard identification and safety analysis. In: Maalej W, Thurimella AK (eds) Managing Requirements Knowledge. Springer, Berlin, Heidelberg, pp 117–141

80. Diallo MH, Naslavsky L, Alspaugh T, Ziv H, Richardson D (2008) Toward architecture evaluation through ontology-based requirements-level scenarios. In: de Lemos R, Di Giandomenico F, Gacek C, Muccini H, Vieira M (eds) Architecting Dependable Systems, vol 5135. Springer, Berlin, Heidelberg, pp 225–247

81. Dzung DV, Ohnishi A (2009) Improvement of quality of software requirements with requirements ontology. In: Proceedings of international conference on quality software. Republic of Jeju, Korea, pp 284–289

82. Dzung DV, Ohnishi A (2012) A verification method of elicited software requirements using requirements ontology. In: Proceedings of Asia-Pacific Software Engineering Conference, vol 1. APSECHong Kong, China, pp 553–558

83. Elahi G, Yu E, Zannone N (2009) A modeling ontology for integrating vulnerabilities into security requirements conceptual foundations. Conceptual Modeling—ER 2009 NA

84. Farfeleder S, Moser T, Krall A, Stålhane T, Omoronyia I, Zojer H (2011) Ontology-driven guidance for requirements elicitation. In: Antoniou G, Grobelnik M, Simperl E, Parsia B, Plexousakis D, De Leenheer P, Pan J (eds) The semantic web: research and applications, vol 6644. Springer, Berlin, Heidelberg, pp 212–226

85. Gailly F, España S, Poels G, Pastor O (2008) Integrating business domain ontologies with early requirements modelling. In: Song I-Y, Piattini M, Chen Y-P, Hartmann S, Grandi F, Trujillo J, Opdahl A, Ferri F, Grifoni P, Caschera M, Rolland C, Woo C, Salinesi C, Zimányi E, Claramunt C, Frasincar F, Houben G-J, Thiran P (eds) Advances in conceptual modeling – challenges and opportunities, vol 5232. Springer, Berlin, Heidelberg, pp 282–291

86. Gandhi RA, Lee SW (2011) Discovering multidimensional correlations among regulatory requirements to understand risk. ACM Trans Softw Eng Methodol 20(4):1–37

87. Ghaisas S, Ajmeri N (2013) Knowledge-assisted ontology-based requirements evolution. In: Maalej W, Thurimella AK (eds) Managing requirements knowledge. Springer, Berlin, Heidelberg, pp 143–167

88. Gröner G, Staab S (2010) Specialization and validation of statecharts in OWL. In: Cimiano P, Pinto H (eds) Knowledge engineering and management by the masses, vol 6317. Springer, Berlin, Heidelberg, pp 360–370

89. Guizzardi R, Franch X, Guizzardi G (2012) Applying a foundational ontology to analyze means-end links in the i* framework. In: 2012 Sixth International Conference on research challenges in information science (RCIS), pp 1–11

90. Guo J, Wang Y, Trinidad P, Benavides D (2012) Consistency maintenance for evolving feature models. Expert Syst Appl 39(5):4987–4998

91. Hayashi S, Yoshikawa T, Saeki M (2010) Sentence-to-code traceability recovery with domain ontologies. In: Software engineering conference (APSEC), 2010 17th Asia Pacific, pp 385–394

92. He H, Wang Z, Dong Q, Zhang W, Zhu W (2013) Ontology-based semantic verification for UML behavioral models. Int J Softw Eng Knowl Eng 23(2):117–145

93. Hickey AM, Davis A (2007) An ontological approach to requirements elicitation technique selection. In: Sharman R, Kishore R, Ramesh R (eds) Ontologies, vol 14. Springer, US, pp 403–431

94. Hoss A, Carver D (2008) Towards combining ontologies and model weaving for the evolution of requirements models. In: Paech B, Martell C (eds) Innovations for requirement analysis. From stakeholders' needs to formal designs, vol 5320. Springer, Berlin, Heidelberg, pp 85–102

95. Hu H, Yang D, Ye C, Fu C, Li R (2010) Detecting interactions between behavioral requirements with OWL and SWRL. World Acad Sci Eng Technol 72:330–336

96. Innab N, Kayed A, Sajeev ASM (2012) An ontology for software requirements modelling. In: 2012 International conference on Information science and technology (ICIST), pp 485–490

97. Jin Z, Chen X, Zowghi D (2009) Performing projection in problem frames using scenarios. In: Proceedings of Asia-Pacific software engineering conference. APSEC, Penang, pp 249–256

98. Kaiya H, Shimizu Y, Yasui H, Kaijiri K, Saeki M (2010) Enhancing domain knowledge for requirements elicitation with Web mining. In: Proceedings of Asia-Pacific Software Engineering Conference, APSEC, Sydney, pp 3–12

99. Karwowski K, Wysota W, Wytrębowicz J (2009) Computer aided requirements management. In: Nguyen N, Kowalczyk R, Chen S-M (eds) Computational collective intelligence. Semantic web, social networks and multiagent systems, vol 5796. Springer, Berlin, Heidelberg, pp 389–400

100. Kassab M, Ormandjieva O, Daneva M (2011) Relational-model based change management for non-functional requirements: approach and experiment. In: 2011 fifth international conference on research challenges in information science (RCIS), pp 1–9

101. Kitamura M, Hasegawa R, Kaiya H, Saeki M (2009) A supporting tool for requirements elicitation using a domain ontology. In: Filipe J, Shishkov B, Helfert M, Maciaszek L (eds) Software and data technologies, pp 128–140

102. Kof L, Gacitua R, Rouncefield M, Sawyer P (2010) Ontology and model alignment as a means for requirements validation. In: 2010 IEEE fourth international conference on semantic computing (ICSC), pp 46–51

103. Kroha P, Janetzko R, Labra JE (2009) Ontologies in checking for inconsistency of requirements specification. In: Third international conference on advances in semantic processing, 2009. SEMAPRO '09, pp 32–37

104. Lasheras J, Valencia-García R, Fernández-Breis JT, Toval A (2009) Modelling reusable security requirements based on an ontology framework. J Res Pract Inf Technol 41(2):119–133

105. Li G, Jin Z, Xu Y, Lu Y (2011) An engineerable ontology based approach for requirements elicitation in process centered problem domain. In: Xiong H, Lee W (eds) Knowledge science, engineering and management, vol 7091. Springer, Berlin, Heidelberg, pp 208–220

106. Li Z, Wang Z, Zhang A, Xu Y (2007) The domain ontology and domain rules based requirements model checking. Int J Softw Eng Appl 1(1)

107. de Lima EJC, Nt JA, Xexeo GB, de Souza JM (2010) ARARA: a collaborative tool to requirement change awarenes. In: 14th international conference on Computer supported cooperative work in design (CSCWD), 2010, pp 134–139

108. Lima J, Garcia B, Amaral C, Caran G (2011) Building an ontological model for software requirements engineering. In: Cruz-Cunha M, Varajo J, Powell P, Martinho R (eds) ENTERprise Information systems, communications in computer and information science, vol 219. Springer, Berlin, pp 228–237

109. Liu CL (2010) CDADE: conflict detector in activity diagram evolution based on speech act and ontology. Knowl Based Syst 23(6):536–546

110. Liu CL, Yang HL (2012) Applying ontology-based blog to detect information system post-development change requests conflicts. Inf Syst Front 14(5)

111. López C, Astudillo H, Cysneiros L (2008) Semantic-aided interactive identification of reusable NFR knowledge fragments. In: Meersman R, Tari Z, Herrero P (eds) On the move to meaningful internet systems: OTM 2008 workshops, vol 5333. Springer, Berlin, Heidelberg, pp 324–333

112. Machado BN, De Oliveira Arantes L, De Almeida Falbo R (2011) Using semantic annotations for supporting requirements evolution. In: SEKE 2011—proceedings of the 23rd international conference on software engineering and knowledge engineering, pp 185–190. Ontology and Conceptual Modeling Research Group (NEMO), Computer Science Department, Federal University of Espirito Santo, Vitoria

113. Moser T, Winkler D, Heindl M, Biffl S (2011) Requirements management with semantic technology: An empirical study on automated requirements categorization and conflict analysis. In: Mouratidis H, Rolland C (eds) Advanced information systems engineering, vol 6741. Lecture notes in computer science. Springer, Berlin, pp 3–17

114. Moshirpour M, Mireslami S, Alhajj R, Far BH (2012) Automated ontology construction from scenario based software requirements using clustering techniques. In: Information Reuse and Integration (IRI), 2012 IEEE 13th International Conference, 8–10 Aug 2012, pp 541–547

115. Mtibaa A, Gargouri F (2009) A multi-representation ontology for the specification of multi-context requirements. In: Damiani E, Yetongnon K, Chbeir R, Dipanda A (eds) Advanced internet based systems and applications, vol no 4879. Springer, Berlin, Heidelberg, pp 259–269

116. Nguyen TH, Vo BQ, Lumpe M, Grundy J (2013) KBRE: a framework for knowledge-based requirements engineering. Softw Qual J 22(1):87–119

117. Niemelã E, Evesti A, Savolainen P (2008) Modeling quality attribute variability. In: ENASE 2008—proceedings of the 3rd international conference on evaluation of novel approaches to software engineering. VTT Technical Research Centre of Finland, Kaitovayla 1, 90571 Oulu, pp 169–176

118. Omoronyia I, Sindre G, Stålhane T, Biff S, Moser T, Sunindyo W (2010) A domain ontology building process for guiding requirements elicitation. In: Wieringa R, Persson A (eds) Requirements engineering: foundation for software quality, vol no 6182. Springer, Berlin, Heidelberg, pp 188–202

119. Osis J, Slihte A, Jansone A (2012) Using use cases for domain modeling. In: ENASE 2012—proceedings of the 7th international conference on evaluation of novel approaches to software engineering. Institute of Applied Computer Systems, Riga Technical University, Kalku iela 1, Riga, pp 224–231

120. Pires PF, Delicato FC, Cóbe R, Batista T, Davis JG, Song JH (2011) Integrating ontologies, model driven, and CNL in a multi-viewed approach for requirements engineering. Requir Eng 16(2)

121. Polpinij J (2009) An ontology-based text processing approach for simplifying ambiguity of requirement specifications. In: Services computing conference, 2009. APSCC 2009. IEEE Asia-Pacific, pp 219–226

122. Rashwan A, Ormandjieva O, Witte R (2013) Ontology-based classification of non-functional requirements in software specifications: a new corpus and svm-based classifier. In: IEEE 37th Annual computer software and applications conference (COMPSAC), 2013, pp 381–386

123. Reinhartz-Berger I, Sturm A, Wand Y (2011) External variability of software: classification and ontological foundations. In: Jeusfeld M, Delcambre L, Ling T-W (eds) Conceptual modeling - ER 2011, vol no 6998. Springer, Berlin, Heidelberg, pp 275–289

124. Riechert T, Berger T (2009) Leveraging semantic data wikis for distributed requirements elicitation. In: Proceedings—international conference on software engineering, Vancouver, BC pp 7–13

125. Saeki M, Hayashi S, Kaiya H (2013) Enhancing goal-oriented security requirements analysis using common criteria-based knowledge. Int J Softw Eng Knowl Eng 23(5):695–720

126. Schugerl P, Rilling J, Witte R, Charland P (2009) A quality perspective of software evolvability using semantic analysis. In: IEEE international conference on semantic computing, 2009. ICSC '09, pp 420–427

127. Souag A, Salinesi C, Wattiau I, Mouratidis H (2013) Using security and domain ontologies for security requirements analysis. In: 2013 IEEE 37th annual computer software and applications conference workshops (COMPSACW), pp 101–107

128. Wang HH, Damljanovic D, Sun J (2010) Enhanced semantic access to formal software models. In: Dong J, Zhu H (eds) Formal methods and software engineering, vol no 6447. Springer, Berlin, Heidelberg, pp 237–252

129. Wang HH, Li YF, Sun J, Zhang H, Pan J (2007) Verifying feature models using OWL. Web Semant Sci Serv Agents World Wide Web 5(2):117–129

130. Yu E, Liu L, Mylopoulos J (2007) A social ontology for integrating security and software engineering. In: Mouratidis H, Giorgini P (eds) Integrating security and software engineering: advances and future visions. Idea Group Publishing, Hershey, PA, pp 70–106

131. Zhang Y, Zhang W (2007) Description logic representation for requirement specification. In: Shi Y, Albada G, Dongarra J, Sloot P (eds) Computational science—ICCS 2007, vol 4488. Lecture notes in computer science, vol 4488. Springer, Berlin, pp 1147–1154