Reducing the Effort for Systematic Reviews in Software Engineering

Francesco Osborne [a,*], Henry Muccini [b], Patricia Lago [c], and Enrico Motta [d]

[a] *Knowledge Media Institute, The Open University, UK*
E-mail: francesco.osborne@open.ac.uk; ORCID: https://orcid.org/0000-0001-6557-3131
[b] *DISIM Department, University of L'Aquila, Italy*
E-mail: henry.muccini@univaq.it; ORCID: https://orcid.org/0000-0001-6365-6515
[c] *Department of Computer Science, Vrije Universiteit Amsterdam, The Netherlands*
E-mail: p.lago@vu.nl; ORCID: https://orcid.org/0000-0002-2234-0845
[d] *Knowledge Media Institute, The Open University, UK*
E-mail: enrico.motta@open.ac.uk; ORCID: https://orcid.org/0000-0003-0015-1952

Abstract. *Context.* Systematic Reviews (SRs) are means for collecting and synthesizing evidence from the identification and analysis of relevant studies from multiple sources. To this aim, they use a well-defined methodology meant to mitigate the risks of biases and ensure repeatability for later updates. SRs, however, involve significant effort.
Goal. The goal of this paper is to introduce a novel methodology that reduces the amount of manual tedious tasks involved in SRs while taking advantage of the value provided by human expertise.
Method. Starting from current methodologies for SRs, we replaced the steps of keywording and data extraction with an automatic methodology for generating a domain ontology and classifying the primary studies. This methodology has been applied in the software engineering sub-area of software architecture and evaluated by human annotators.
Results. The result is a novel Expert-Driven Automatic Methodology, EDAM, for assisting researchers in performing SRs. EDAM combines ontology-learning techniques and semantic technologies with the human-in-the-loop. The first (thanks to automation) fosters scalability, objectivity, reproducibility and granularity of the studies; the second allows tailoring to the specific focus of the study at hand and knowledge reuse from domain experts. We evaluated EDAM on the field of Software Architecture against six senior researchers. As a result, we found that the performance of the senior researchers in classifying papers was not statistically significantly different from EDAM.
Conclusions. Thanks to automation of the less-creative steps in SRs, our methodology allows researchers to skip the tedious tasks of keywording and manually classifying primary studies, thus freeing effort for the analysis and the discussion.

Keywords: software engineering, ontology learning, semantic web, software architecture, digital libraries, systematic reviews

1. Introduction

Understanding the state-of-the-art in research provides the foundation for building novelty. In particular, in software engineering topic areas, the acquisition of knowledge for this understanding follows a clear path: started with informal reviews and surveys, it is moving towards systematic searches of the literature. Kitchenham [15] clearly explains the reasons, the importance, and the advantages and disadvantages of using systematic reviews instead of informal ones. Various studies (e.g., [7, 49]) reveal

*Corresponding author. E-mail: francesco.osborne@open.ac.uk.

the growing interest of our community in systematic literature reviews and systematic mapping studies [47]. A number of articles and books have been written on how to perform such systematic studies [17, 42, 46].

A Systematic Review (or simply, SR) is *"a means of evaluating and interpreting all available research relevant to a particular research or topic area or phenomenon of interest"* [15]. Given a set of research questions, and by following a systematically defined and reproducible process, a SR helps selecting primary studies that contribute to provide an answer to them. Used in combination with keywording [30], a SR supports the systematic elicitation of an ontological classification framework [31].

A SR can help researchers and practitioners in creating a complete, comprehensive and valid picture of the state-of-the-art about a given theme when the search-space is bounded (e.g., when the search query returns few thousands of articles to scrutinize). However, it falls short when used to investigate the state-of-the-art on an entire research area (e.g., software architecture) where the returned entries are hundreds of thousands - hence clearly unmanageable. As reported by Vale et al. [41] while investigating the state-of-the-art of the Component-Based Software Engineering area through an SR, a *"... manual search* [restricted only to the most relevant journals and conferences related to the CBSE area] *was considered as the primary source, given the infeasibility of analyzing all studies collected from automatic search"*. Still, they had to select, read, and thoroughly analyze 1,231 primary studies.

In contrast to manually run SRs, several state of the art automated methods allow classifying a document in a certain category or topic [2, 4, 22, 39]. Unfortunately, most current techniques suffer from limitations that make them unsuitable for systematic reviews. The approaches which exploit keywords as proxy for research areas are unsatisfactory, as they fail to distinguish research topics from other terms that can be used to annotate papers (e.g., "user case", "scalability") and to take advantage of the relationships that hold between research areas (e.g., the fact that "Software Architecture" is a sub-area of "Software Engineering"). Probabilistic topic models (e.g., Latent Dirichlet Allocation [4]) are also unsuitable for this task since they produce cluster of terms that are not easy to map to research areas [28]. Crucially, it is often unfeasible to integrate these topic detection techniques with the needs and the knowledge of human experts. Another alternative is to apply entity linking techniques [22] to map papers to relevant entities in knowledge base. Unfortunately, we currently lack good granular and machine readable representation of research areas in many domains which could be used to this end.

Current techniques have complementary limitations when investigating the state-of-the-art of an entire research area: on the one hand side, SRs are *"human-intensive"*, as they require domain experts to invest a large amount of time to carry out manual tasks; on the other side, automated techniques keep the *humans "out of the loop"*, while human expertise is critical for the more conceptual analysis tasks.

This paper proposes an *expert-driven automatic methodology* for assisting systematic reviews that, while recognizing the essential value of human expertise, limits the amount of tedious tasks the expert has to carry out. Our methodology contributes with 1) automatically extracting an ontology of relevant topics, related to a given research area; 2) using experts to refine this knowledge base; 3) exploiting this knowledge base for classifying relevant papers that may be then further validated/analyzed by experts, and for computing analytics.

In summary, our contributions are:

- a novel methodology for supporting ontology-driven systematic reviews, which involves both automatic techniques and human experts;
- an implementation of this methodology which exploits the Klink-2 algorithm for generating the domain ontology in the field of software architecture;
- an illustrative analysis of the software architecture trends;

- an evaluation involving six human annotators, which shows that the classification of primary studies yielded by the proposed methodology is comparable to the one produced by domain experts (p=0.77).
- an automatically generated ontology of Software Engineering, which could support further systematic reviews in the field[1].

The rest of the paper is structured as follows. Section 2 introduces related works on systematic studies. Section 3 provides an overview of some preliminary evidence of the benefits brought by using EDAM to assist a mapping study. Section 4 then presents the EDAM methodology and its application to the research area of software architecture. The discussion is presented in Section 5, and the conclusions in Section 6.

2. Related Work

There are many guidelines for, and reports on, carrying out systematic studies in software engineering. Among them, we could identify a few aimed at supporting or improving the underlying process. In our perspective, they all enable researchers to focus more on the most creative steps of a systematic study by removing what is referred to as *manual work*.

With a motivation similar to ours, i.e. to improve the search step in systematic studies in software engineering research, Octaviano et al. [25] propose a strategy that automates part of the primary study selection activity. Mourão et al. [24] present a preliminary assessment of a hybrid search strategy for systematic literature reviews that combines database search and snowballing to reduce the effort due to searches in multiple digital libraries. Kuhrmann et al. [19] provide recommendations specifically for the general study design, data collection, and study selection procedures. Zhang et al. [50], in turn, systematically select and analyze a large number of SLRs. Their results have been then used to define a quasi-gold standard for future studies. In their validation, they were able to improve the rigor of the search process and provide guidelines complementing the ones already in use.

Ros et al. [34] propose a machine learning approach that classifies papers for SLRs by leveraging human experts, who iteratively validate set of publications produced by a classifier. Conversely, EDAM does not require experts to manually examine research papers, but only to review a taxonomy of research areas.

The need for guidelines in conducting empirical research has been addressed in other types of empirical studies, too. De Mello and Travassos [9] focus on opinion surveys and provide guidelines (in the form of a reference framework) aimed to at improving the representativeness of samples. Also on opinion surveys, Molleri et al. [23] provide recommendations based on an annotated bibliography instead.

Another interesting work by Felizardo et al. [11] investigates how the use of forward snowballing can considerably reduce the effort in updating SLRs in software engineering. Based on this result, complementing our method with automated forward snowballing suggests a very promising direction for future works as it could further reduce the effort for identifying relevant primary studies.

Marshall et al. [21] carried out an interview survey with experts in other domains (i.e. healthcare and social sciences) with the aim to identify tools that are generally used, or desirable, to ease which steps in systematic studies, and transfer the best practices to the software engineering domain. Among the results, data extraction and automated analysis emerge as top requirements for reducing the workload.

[1] http://rexplore.kmi.open.ac.uk/data/edam/SE-ontology.owl

In a similar vein, Hassler et al. [14] followed by Al-Zubidy et al. [1] consulted software engineering researchers conducting SLRs to identify and prioritize the necessary SLR tool features. The results identified *search & study selection* as the most desirable feature. Our work addresses the needs identified by both [21] and [14].

The idea of using ontologies for supporting SRs was discussed by few papers, but did not receive much attention. de Almeida Biolchini et al. [8] introduced the Scientific Research Ontology, a resource to organize the knowledge generated from SR. This ontology offers a conceptual framework with the aim of fostering the consistency between different studies, but does not directly assist the tasks involved in SR, such as the extraction of primary studies. Sun et al. [40] discussed the use of ontologies for supporting key activities in SRs and presented an experiment in which they automatically classified primary studies by means of COSONT, an ontology of methods for cost estimation. Unfortunately, their approach still required to manually check hundred of papers and the COSONT ontology was quite simplistic, being an handcrafted list of methods with no hierarchical structure. This is a common issue with manually generated ontology of research concepts, which are usually costly to produce, coarse-grained, and slow to evolve [27]. Conversely, EDAM takes advantage of recent ontology learning techniques to automatically generate complex multi-level ontologies (e.g., the SE ontology presented in this paper includes 956 topics and 5,461 relationships), exploits the resulting taxonomic structure to classify the primary studies, and does not require experts to manually review a large number of papers.

3. An Overview of the Benefits of Automatic SRs

Before entering the details of the EDAM methodology, this section provides an overview of the benefits such an automatic SR methodology can bring with respect to more traditional, manual SRs carried out according to predefined protocols.

We all agree that manual SRs based on well-defined systematic protocols help reducing (but not fully removing) subjective biases in the selection of the studies. They however are by and large unfeasible in reviewing a too large dataset (i.e. when the number of scientific publications is too large to be manually processed by the researcher).

In a similar vein, automatic SRs help reducing subjective biases (in this case by *implementing* the selection of the studies according to the predefined systematic protocol). Differently, they pose no limitation in terms of the size of the dataset of publications.

In our earlier work [48] we challenged these limitations and benefits by applying the automatic study selection to a manual SR carried out beforehand by other researchers [12]. In this way, we could compare and contrast the results of the manual SR with the results of our automatic SR. In this earlier work, we have studied the field of software sustainability within the software engineering domain. While at the time the EDAM methodology was not yet fully developed, we did use the same ontology-learning algorithms and a preliminary version of the ontology for the software engineering domain.

The observations gathered during this experiment are illustrated in Fig. 1, where we represented the primary studies selected manually (see the left-hand circles) and those selected automatically (see the right-hand ovals). The experiment underwent three phases:

Starting point: The already-completed manual SR had selected 116 primary studies. Before training the algorithm and tuning the domain ontology, from the Scopus dump of scientific publications we automatically selected 950 studies. While our automatic methodology is able to handle seamlessly any size of the base of publications, the selected studies did initially include a very large number

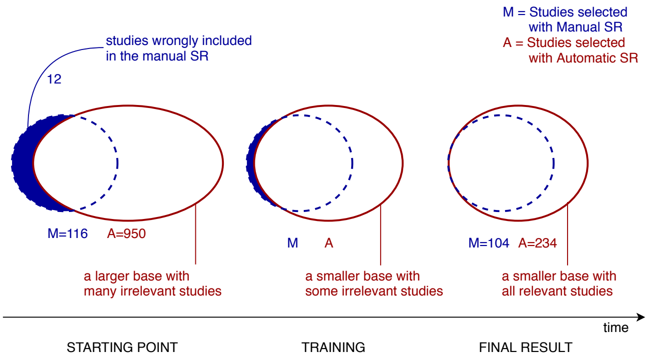

Fig. 1. Some evidence on the Benefits of Automated SRs

of false positives. However, it did also uncover that 12 studies selected in the manual SR where wrongly included. **Observation #1:** *in spite of systematic selection criteria and the involvement of multiple researchers, human errors in the manual study selection is still possible.*

Training: By treating the 104 primary studies (from the manual SR) as pilot studies, we trained our domain ontology and learning algorithm to automatically select the primary studies. **Observation #2:** *Automatic SR is able to automatize the selection criteria of systematic reviews while handling any size of the initial dataset of scientific publications.* As discussed in Section 5.1, the domain ontology is able to classify the primary studies as correctly as the human experts do, without needing further training. As such, the domain ontology can be reused for any study in the domain of software engineering.

Final result: The final result of the automatic selection converged to 234 studies which included the 104 pilot studies and *correctly* identified additional 130 studies that were missing in the original manual SR. **Observation #3:** *By handling a much larger base of publications, automatic SRs are able to uncover primary studies that are missed by manual SRs where such scale is unfeasible.*

4. An Expert-Driven Automatic Methodology

We propose a novel expert-driven automatic methodology (EDAM) for assisting systematic reviews like systematic literature reviews and mapping studies. EDAM allows to automatize the steps that are the most time and effort consuming while requiring the least creativity, such as *selection of relevant papers*, *keywording*, and *creation of a classification schema* [31], by exploiting ontology learning techniques and semantic technologies to foster scalability, objectivity, reproducibility, and granularity of the study (further discussed in Section 5.3). It also supports the generation of research trends, which are typical of data synthesis in mapping studies. In this paper, we illustrate how EDAM can support mapping studies, even though it can be evidently exploited in systematic literature reviews, too.

Figure 2 shows the steps of a mapping study using EDAM in contrast with the steps of a classic (manual) methodology - shown in Figure 3. The main difference is that in the classic methodology the researchers first select and analyze each primary study (steps 2-3) and then produce a taxonomy to classify them (step 4). When assisted by EDAM, instead, the researchers first use ontology learning methods over large scholarly datasets to generate an ontology of the field (steps 2-3), then refine the ontology with the help of domain experts (step 4), and finally exploit this knowledge base to automatically select and classify the primary studies (steps 5-6).

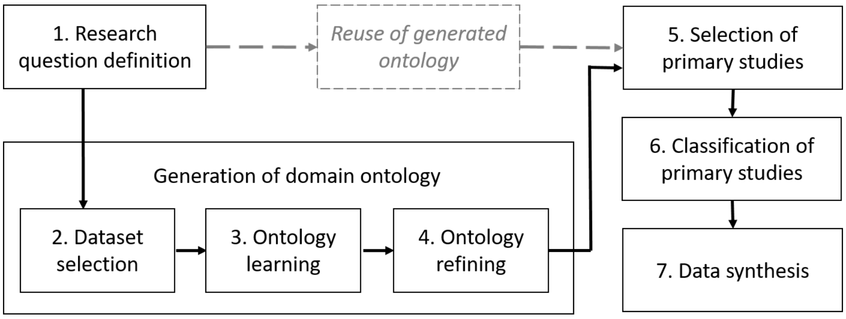

Fig. 2. Steps of a systematic mappings adopting the EDAM methodology. The gray-shaded elements refer to the alternative step of reusing the previously generated ontology.

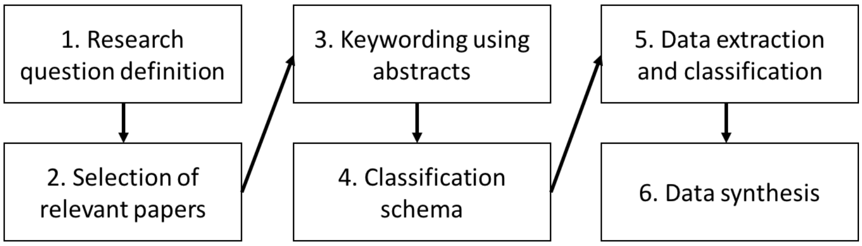

Fig. 3. Classic steps of systematic mappings (inspired by [31])

An alternative solution for steps 2-4 (Generation of domain ontology) is the reuse of an ontology crafted by a previous study with the same scope. Indeed, in the study discussed in Section 4.2 we have generated an ontology of Software Engineering (SE) research topics, with the hope that it will be re-used by the research community.

In Section 4.1, we describe EDAM and discuss its advantages over a classic methodology. In Section 4.2, we exemplify the application of EDAM specifically aimed at identifying publication trends of the software architecture research area in the specific SE domain.

4.1. EDAM Description

A SR assisted by EDAM is organized along the following steps (ref. Figure 2).

1. Research question definition. The researchers performing the study state the research questions (RQs). These will affect the aim of the study and thus its steps. It should be noted that EDAM is applicable only to research questions that could be answered by classifying publications, authors, venues, and other entities according to the ontology for producing relevant analytics. Other research questions should be addressed according to the standard methodology [31].

2. Dataset selection. The researchers select a dataset on which to apply the chosen ontology learning technique (further elaborated in step 3) for generating the domain ontology that will be used to select and classify the primary studies. The most important characteristic of this dataset is that it must be unbiased with respect to the focus of the study. For example, if the study wants to uncover the trends in research areas (e.g., software architecture), the dataset should not be biased with respect to any area in the domain (e.g., software engineering in our case). A good strategy to select unbiased datasets is considering either a full scholarly dataset of a very high-level field (e.g., all the Computer Science papers in Microsoft Academic Search[2] or in Scopus[3]) or a dataset including all the papers published in the main conferences and journals of the domain under analysis. In recent years, universities, organizations, and publishing companies have released an increasing number of open datasets that could assist in this task, such as CrossRef[4], SciGraph[5], OpenCitations[6], DBLP[7], Semantic Scholar[8], and others.

3. Ontology learning. The dataset is processed by an ontology learning technique that automatically infers an ontology of the relevant concepts.

We strongly advocate the use of an ontology learning technique that generates a full domain ontology and represents it with Semantic Web standards, such as the Web Ontology Language (OWL)[9]). The main advantage of adopting an ontology in this context is that it allows for a more comprehensive representation of the domain since it includes, in addition to hierarchical relationships, also other kinds of relationships (e.g., *sameAs*, *partOf*), which may be critical for classifying the primary studies. For example, an ontology allows to explicitly associate to each category a list of alternative labels or related terms that will be used in the classification phase. In addition, ontology learning techniques can infer very structured multi-level ontologies [27], and thus describe the domain at different levels of granularity.

The task of ontology and taxonomy learning was comprehensively explored over the last 20 years. Therefore, the researcher can choose among a variety of different approaches for this step, including:

- statistical methods for deriving taxonomies from keywords [20, 38];
- natural language processing approaches, e.g., FRED [13], LODifier [3], Text2Onto [6];
- approaches based on deep learning, e.g., recurrent neural networks [32];
- hybrid ontology learning frameworks [44];
- specific approaches for generating research topic ontologies, e.g., Klink-2 [27].

However, as discussed in the following step, researchers may also chose to skip this step and re-use a compatible ontology from a previous study.

It is useful to clarify why we suggest the adoption of an ontology learning approach, rather than the adoption of one of the currently available research taxonomies, such as the ACM computing classification system[10], the Springer Nature classification[11], Scopus subject areas[12], and the Microsoft Academic Search classification. Unfortunately, these taxonomies suffer from some common issues, which make them unfeasible to support most kinds of SRs. First, they are very coarse-grained and represent wide categories of approaches, rather than the fine-grained topics addressed by researchers [26]. Secondly, they are usually obsolete since they are seldom updated. For example, the 2012 version of the ACM classification was finalized fourteen years after the previous version. This is a critical point, since some

[2]http://academic.research.microsoft.com
[3]https://www.scopus.com/
[4]https://www.crossref.org/
[5]https://scigraph.springernature.com/explorer/downloads/
[6]http://opencitations.net
[7]http://dblp.uni-trier.de
[8]https://www.semanticscholar.org/
[9]https://www.w3.org/OWL/
[10]http://www.acm.org/publications/class-2012
[11]http://www.nature.com/subjects
[12]https://www.elsevier.com/solutions/scopus/content

interesting trends could be associated with recently emerged topics. In third instance, most ontology learning algorithms are not limited to learning research areas, but can be tailored to yield the outputs which are more apt to support a specific analysis.

4. Ontology refining. The ontology resulting from the previous step is corrected and refined by domain experts. During this phase, the experts are allowed to 1) delete an existent category, 2) add a new category, 3) delete an existent relationship, 4) add a new relationship. We suggest using at least three domain experts for addressing possible disagreements.

This step is critical for two reasons. First, it may correct some errors in the automatically-generated taxonomy. Secondly, it verifies that the data-driven representation aligns with the domain experts mental model and thus the outcomes will be understandable and reusable by their research community.

Refining a very large ontology is not a trivial task, therefore if the domain comprehends a large number of topics we suggest to split it in manageable sub branches to be addressed by different experts. Our experience suggests that a taxonomy of about 50 research areas can be reviewed in about 15-30 minutes by an expert of the field. For example, in [27] three experts reviewed a Semantic Web ontology of 58 topic in about 20 minutes. In the test study for this paper, three experts took about 20 minutes to examine and produce feedback on a taxonomy of 46 topics (and 71 terms considering synonymous such as “product line", “product-lines", “product-line", which were clustered automatically by the ontology learning algorithm). In both cases, we represented the ontology as tree diagram in a excel sheet[13] and included also a list of the most popular terms in the dataset, for supporting experts in remembering all the relevant research topics. The involved researchers had no problem in understanding this simple representation and modified the spreadsheet according to their expertise.

An alternative solution is to provide experts with ontology editors that could be used to directly modify the ontology, such as Protege[14], NeOn Toolkit[15], TopBraid Composer[16], Semantic Turkey[17], or Fluent Editor[18]. However, these tools are not always easy to learn and we thus believe that the adoption of a simple spreadsheet would be advisable in most cases. As highlighted by Figure 2, the aim of steps 2-4 is to generate an ontology apt to select and classify relevant papers and ultimately answer the RQs. It follows that these steps could be replaced by the adoption of an ontology previously generated and validated by a previous study with a consistent scope. For example, the ontology about software engineering generated for this paper’s example study (see Section 4.2) can be re-used to perform many kinds of mapping studies involving other research areas in SE. Naturally, the ontology may have to be further updated to include the most recent concepts and terms. This solution allows users with no access to vast scholarly databases or no expertise in ontology learning techniques to easily implement an EDAM study.

5. Selection of primary studies. The authors select a dataset of papers and define the inclusion criteria of the primary studies according to the domain ontology and other metadata of the papers (e.g., year, venue, language). The inclusion criteria need to be expressed as a query that can be run automatically over the dataset. Some examples of queries for the selection of primary studies include 1) “all the papers

---

[13] See an example at http://tinyurl.com/yal6h3wu

[14] http://protege.stanford.edu

[15] http://neon-toolkit.org/

[16] http://www.topquadrant.com/products/TB_Composer.html

[17] http://semanticturkey.uniroma2.it/

[18] http://www.cognitum.eu/Semantics/FluentEditor/

in the dataset published in a list of relevant conferences" or "all the papers in the dataset that contain a list of relevant terms from the ontology".

In most cases this dataset will be the same or a subset of the one used for learning the domain ontology. However, the authors may want to zoom on a particular set of articles, such as the ones published in the main venues of a field, in a geographical area, or by a certain demography. It is also possible to select a different dataset altogether, since the ontology would use generic topic labels and thus be agnostic with respect to the dataset. A possible reason to do so is the availability of the full text of the studies. Many ontology learning algorithms can be run on massive metadata dataset (e.g., Scopus, Microsoft Academic Search), but some research questions may require the full text. In this case, the author may want to perform the ontology learning step on the metadata dataset, which is usually larger in size and scope, and then either select a subset composed by publications which are available online or adopt for this phase a second dataset that includes the full text of the articles, such as Core [18]. The growth of the Open Access movement [43], which aims at providing free access to academic work, may alleviate this limitation in the following years.

6. Classification of primary studies. The authors define a function for mapping categories to papers based on the refined ontology. This step is important to foster reproducibility since the inclusion criteria (defined in the step 5), the mapping function, and the domain ontology should contain all the information needed for replicating the classification process. The function can also be associated to an algorithmic method (e.g., a machine learning classifier), provided the method is made available and is reproducible.

The simplest way for mapping categories to papers is to associate to each category each paper that contains the label of the category or of any of its sub-categories. This simple technique for semantically characterizing documents was applied with good results in a variety of fields, such as topic forecasting [36], automatic classification of proceeding books [29], sentiment analysis [35], recommender systems [10], and many others.

In addition, the authors can choose to create a more complex mapping function which exploits other semantic relationships in the ontology (e.g., *relatedTerm*, *partOf*).

7. Data synthesis. According to the RQs, this step may be automatic, semi-automatic or manual. Some straightforward analytics (e.g., the number of publications or citations over time) can be computed completely automatically by counting the previously classified papers or summing their number of citations. Other more complex analyses may require the use of machine learning techniques or the (manual) intervention of human experts. Starting from the groundwork formed by our research, a full analysis of the possible kinds of data synthesis and the way to automatize them will constitute interesting future works beneficial for the whole research community.

Overall, motivated by the need to reduce the amount of manual tedious tasks involved in SRs, **EDAM offers four main advantages over a classic methodology**. **First,** human experts are not required to manually analyze and classify primary studies, but they simply have to refine the ontology, choose the inclusion criteria, and define a mapping function for associating papers to categories in the ontology. This allows researchers to carry out large scale studies that involve thousands of research papers with relative ease. **Secondly,** since the domain ontology is created with a data-driven method, it should reflect the real trends of the primary studies, rather than arbitrary human decisions about which keywords to annotate and aggregate, even if the refinement step may still introduce a degree of arbitrariness. **Third,** the use of a formal machine-readable ontology language for representing the domain taxonomy should foster the reproducibility of the study and allow authors with no expertise in data science to perform studies using previously generated ontologies. **Fourth,** this methodology allows researchers to produce and exploit complex multi-level ontologies, rather than the simple two-level classifications used by many studies [41].

Naturally, EDAM is suitable for research questions that can be automatized by the ontology-driven classification process previously described, or that aim at giving an overview of the state-of-the-art or state-of-practice on a topic [45] by analysing all of the relevant research contributions in a specific research area. We will discuss further this and other limitations in Section 5.2.

4.2. EDAM Application

With the aim of presenting a reproducible pipeline and showing how EDAM can be applied, we present here an example as part of a possible systematic mapping study assisted by EDAM in the software architecture research area. We chose to study the research trends in this area, since trend analysis is typical of mapping studies [45] and it is one of the tasks that can be automatized by EDAM.

In the following, we describe how we instantiated the study example assisted by EDAM and discuss the specific technologies used to implement it. The data necessary for reproducing this study and using this same pipeline on other fields are available at http://tinyurl.com/ycgbyas9.

1. Research question definition. We wanted to focus on a task that is often addressed by mapping studies and could be completely automatized. Therefore our RQ is: “What are the trends of the main research topics of software architecture?".

2. Dataset selection. We selected all papers in a dump of the Scopus dataset about Computer Science in the period 2005-2013. The Scopus dataset we were given access by Elsevier BV includes papers in 1900-2013 interval, but the number of relevant articles before 2005 was too low to allow a proper trend analysis. Each paper in this dataset is described by title, abstract, keywords, venue, and author list.

3. Ontology learning. We applied the Klink-2 algorithm [27] on the Scopus dump for learning an ontology representing the main ’software architecture’ research area in SE.

Klink-2 is an algorithm that generates an ontology of research topics by processing scholarly metadata (titles, abstracts, keywords, authors, venues) and external sources (e.g., DBpedia, calls for papers, web pages). It is integrated in Rexplore[19] [28], a system that uses semantic technologies for exploring and making sense of scholarly data. In particular, Klink-2 periodically produces the Computer Science Ontology (CSO)[20] [37] that is currently used by Springer Nature for classifying proceedings in the field of Computer Science [29], such as the well-known Lecture Notes in Computer Science series[21]. The ontologies produced by Klink-2 use the Klink data model[22], which is an extension of the BIBO ontology[23] that in turn builds upon SKOS[24]. This model includes three semantic relations: *relatedEquivalent*, which indicates that two topics can be treated as equivalent for the purpose of exploring research data;

[19] http://skm.kmi.open.ac.uk/rexplore/
[20] http://cso.kmi.open.ac.uk/
[21] http://www.springer.com/gp/computer-science/lncs
[22] http://technologies.kmi.open.ac.uk/rexplore/ontologies/BiboExtension.owl
[23] http://purl.org/ontology/bibo/
[24] https://www.w3.org/2004/02/skos/

skos:broaderGeneric, which indicates that a topic is a subarea of another one; and *contributesTo*, which indicates that the research outputs of one topic significantly contribute to the research into another. In the following, we make use of the first two relationships for classifying studies according to their research topics.

We selected Klink-2 among the other previously discussed solutions for a number of reasons. First, it is the only approach to our knowledge that was specifically designed to generate taxonomy of research areas. Secondly, it was already integrated and evaluated on a dump of the Scopus dataset, which we adopted in this study, yielding excellent performance on the fields of artificial intelligence and semantic web [27]. In third instance, it permits to define a number of pre-determinate relationships as basis for a new taxonomy. In particular, a human user can define a subsumption relation (i.e., *skos:broaderGeneric*), a *relatedEquivalent* one, or specify that two concepts should not be in any relationships. This functionality allows us to easily incorporate expert feedback in the ontology learning process. Therefore, the next iterations of the ontology will benefit from the knowledge of previous reviewers.

We ran Klink-2 on the selected dataset, giving as initial seed the keyword “Software Engineering" and generated an OWL ontology of the field including 956 concepts and 5,461 relationships. We then selected the sub-branch of software architecture comprising 46 research areas and 71 terms (some research areas have multiple labels, such as “component based software" and “component-based software").

4. Ontology refining. We generated a spreadsheet, containing the Software Architecture (SA) ontology as a tree diagram[25] . In this representation each concept of the ontology was illustrated by its level in the taxonomy, its labels, and the number of papers annotated with the concepts. We also included a list of the 500 more popular terms in the papers that contained the keywords “Software Architecture" and “Software Engineering", to assist the experts in remembering other concepts or terms that the algorithm may have missed.

We sent it to three senior researchers and asked them to correct the ontology as discussed in Section 4.1. The task took about 20 minutes and produced three revised spreadsheets. The feedback from the experts was integrated in the final ontology[26]. In case of disagreement we went with the majority vote.

The most frequent feedback regarded: 1) the deletion sub-areas that were incorrectly classified under SA (e.g., “software evolution"), 2) the introduction of sub-areas that were neglected by Klink-2 (e.g, “architecture concerns"), and 3) the inclusion of alternative labels for some category (e.g., alternative ways to spell “component-based architecture").

5. Selection of primary studies. We then selected from the initial Scopus dump two datasets of primary studies to investigate the SA area: 1) **DSA** (Dataset SA, 3,467 publications), including all papers in the Scopus dataset that contain the terms “software architectures" or “software architecture" and include at least one of the subtopics of software architecture in the domain ontology, and 2) **DSA-MV** (Dataset SA - Main Venues, 1,586 publications), containing all the papers published in a list of well-known conferences and journals in the SE fields and in a particular in the SA area (see Table 1) and including at least one of the sub-topics of SA in the OWL ontology. We considered these two datasets since it may be interesting to analyze the discrepancy between generic SA papers and papers published in the main

[25]http://tinyurl.com/yal6h3wu
[26]http://rexplore.kmi.open.ac.uk/data/edam/SE-ontology.owl

venues.

6. Classification of primary studies. We defined the mapping function as follows. A paper was classified under a certain category (e.g., service-oriented architectures) if it contained in the title, abstract or keywords: 1) the label of the category (e.g., "service-oriented architectures"), 2) a *relevantEquivalent* of the category (e.g., "service oriented architecture"), 3) a *skos:broaderGeneric* of the category (e.g., "microservices"), or 4) a *relevantEquivalent* of any *skos:broaderGeneric* of the category (e.g., "microservice").

The advantage of this solution is that it allows us to map each category to a list of terms that can be automatically searched in the metadata of the papers. Therefore, the classification step can be handled automatically. In addition, it allows us to associate multiple categories to the same paper.

In practice, we indexed titles, abstracts and keywords in an ElasticSearch[27] instance and we ran a PHP script that imported the ontology, performed the relevant queries on the metadata, and saved the result in a MariaSQL database[28].

7. Data synthesis. Figure 4 shows the number of primary studies in the DSA and DSA-MV datasets. The DSA dataset follows the trend of the "software architecture" keyword in the Scopus dataset and decreases after 2010. Conversely, the size of DSA-MV grows steadily with the number of relevant conferences and journals.

We identified the main trends by running a script to count the number of studies about each subtopic in each year. Since the focus of the paper is the EDAM methodology, rather than a comprehensive analysis on these research sub-areas, we will briefly discuss only the main trends associated with the more popular subtopics (in terms of number of papers). The full results of this example study, however, are available at rexplore.kmi.open.ac.uk/data/edam and can be reused for supporting a more in-depth analysis of the field.

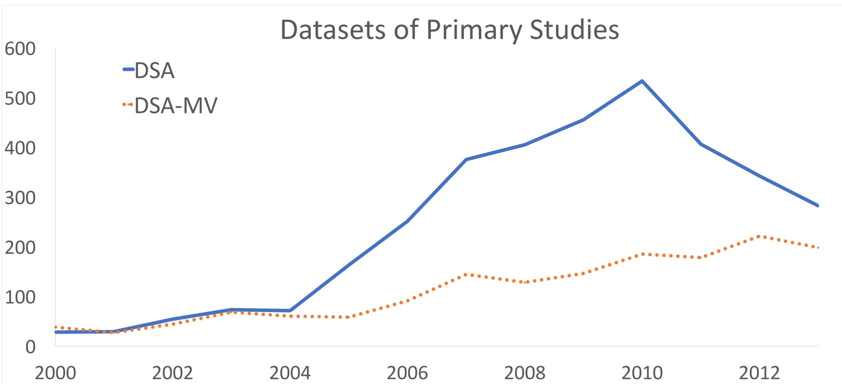

Fig. 4. Number of publications in DSA and DSA-MV over the years.

Figure 5 displays the number of publications and citations associated with the most popular sub-areas of SA. The papers in DSA yield on average 4.8 ± 2.1 in citations versus the 13.6 ± 7.0 citations of those in DSA-MV. Reasonably, this tendency suggests that the papers published in the main SA venues tend to be more recognized by the research community.

[27]https://www.elastic.co/
[28]https://mariadb.org/

Conferences
WICSA - IEEE/IFIP Conference on Software Architecture, ECSA - European Conference on Software Architecture, CBSE - Int. ACM SigSoft Symposium on Component-based Software Engineering, QoSA - Conference on the Quality of Software Architecture , ICSE - ACM/IEEE Int. Conference on Software Engineering, ASE - IEEE/ACM Int. Conference on Automated Software Engineering, ESEC/FSE - European Software Engineering Conference, SEAA - Euromicro Conference on Software Engineering and Advanced Applications, ACM/SAC - ACM Symposium on Applied Computing
Journals
CACM - Communications of the ACM, ACM TOSEM - ACM Transactions on Software Engineering and Methodology, IEEE TSE - IEEE Trans. on Software Engineering, IEEE Software, Elsevier JSS - Journal of Systems and Software, Elsevier IST - Information and Software Technology, Wiley JSME/JSEP - Journal of software: Evolution and Process

Table 1
List of venues used for the DSA-MV dataset.

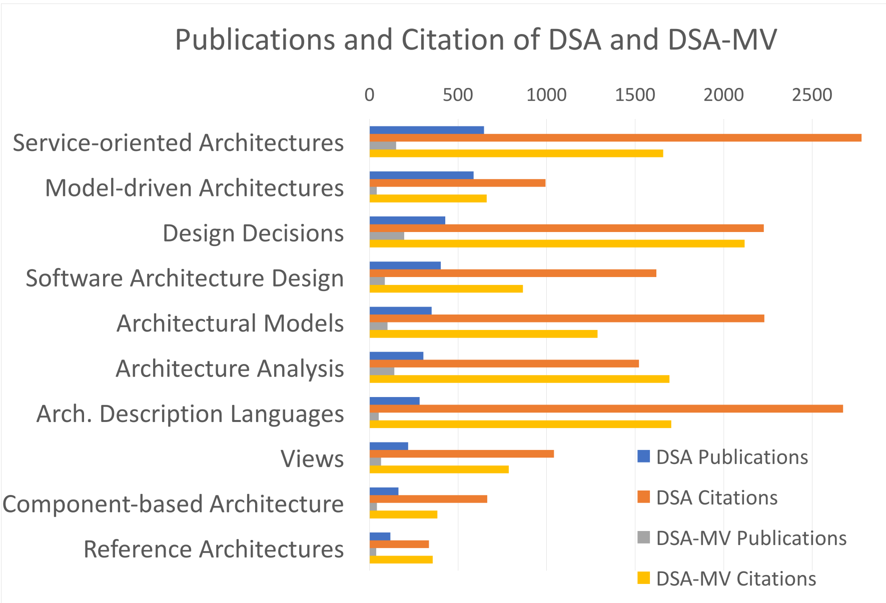

Fig. 5. Number of publications and citations of the main topics in DSA and DSA-MV.

Figure 6 shows the percentage of papers published over time in the main topics within SA. We focus on the 2005-2013 period, since in this interval the number of publications is high enough to highlight the topic trends.

Software-oriented Architectures appears to have been the most prominent topic before 2009, while from 2010, Model-driven Architectures appears to be the most popular topic in this dataset. We can also appreciate the rising of Design Decisions, that seems the most significant positive trend of the last period together with Architecture Description Languages.

Interestingly, the dataset regarding the main venues (DSA-MV) exhibits some different dynamics. Figure 7 highlights the difference between DSA and DSA-MV by showing for each topic the ratio between its number of publications and the total publications in the ten main topics. The research areas of Design Decisions and Views appear much more prominent in the main venues, while Model-Driven Architectures and Architecture Analysis are more popular in DSA. We can further analyze these differences by considering the main trends of the DSA-MV dataset, displayed by Figure 8. The trend of Design Decisions in DSA-MV mirrors the one exhibited in DSA, both growing steadily from 2010. Conversely, Service-oriented Architectures, which has a negative trend in DSA, remains stable in DSA-MV.

5. Discussion

In the following, we reflect on this preliminary application of EDAM. We include an evaluation of the automatic classification of primary studies, an analysis of its limitations, and a discussion about the implications for systematic mappings in software engineering.

5.1. Evaluation of the primary study classification

The most critical step of EDAM is the classification of primary studies. If these are correctly associated to the relevant topics, the subsequent analysis will present a realistic assessment of the landscape of a research field. Thus, even if working on a large number of papers can alleviate the weight of some minor misclassification mistakes, we need to be able to trust the classification process to a good degree.

Unfortunately, it is not easy to produce a gold standard for this kind of task. It is hard to define the set of topics which 'correctly' classify a research paper. Domain experts may disagree for a variety of reasons, including their background and their mental taxonomy of research topics in the field. Therefore, when manually classifying research papers, it is usually good practice to have the same studies analyzed by multiple experts, integrate their annotations, and have a mechanism (e.g., majority vote) to address possible disagreements. On this basis, we assume that the quality of a set of annotations can be measured according to its agreement with the annotations of other domain experts, as also reflected by 'good practices' in empirical software engineering.

We evaluated the ability of EDAM to correctly discriminate between different topics by (1) randomly selecting a set of 25 papers in the DSA dataset, (2) classifying them both with EDAM and with six human experts (researchers in the field of SA), and (3) comparing the results. For simplifying the task and allowing to compare the annotation algorithmically, we first selected five unambiguous categories from the main topics of SA: Design Decisions, Service-oriented Architectures, Model-driven Architectures, Architecture Description Languages, and Views. For each category, we randomly selected from the DSA dataset five primary studies that were classified by EDAM exclusively under that topic, for a total of 25 papers. These papers were described in a spreadsheet by means of their title, author list, abstract, and keywords. The human experts were given this spreadsheet and asked to classify each paper either with one of the five categories or with a "none of the above" tag. We then compared the seven annotation sets produced by the six human experts and by EDAM, considered as an additional annotator[29].

[29]The material and the results of the evaluation are available at http://rexplore.kmi.open.ac.uk/data/edam

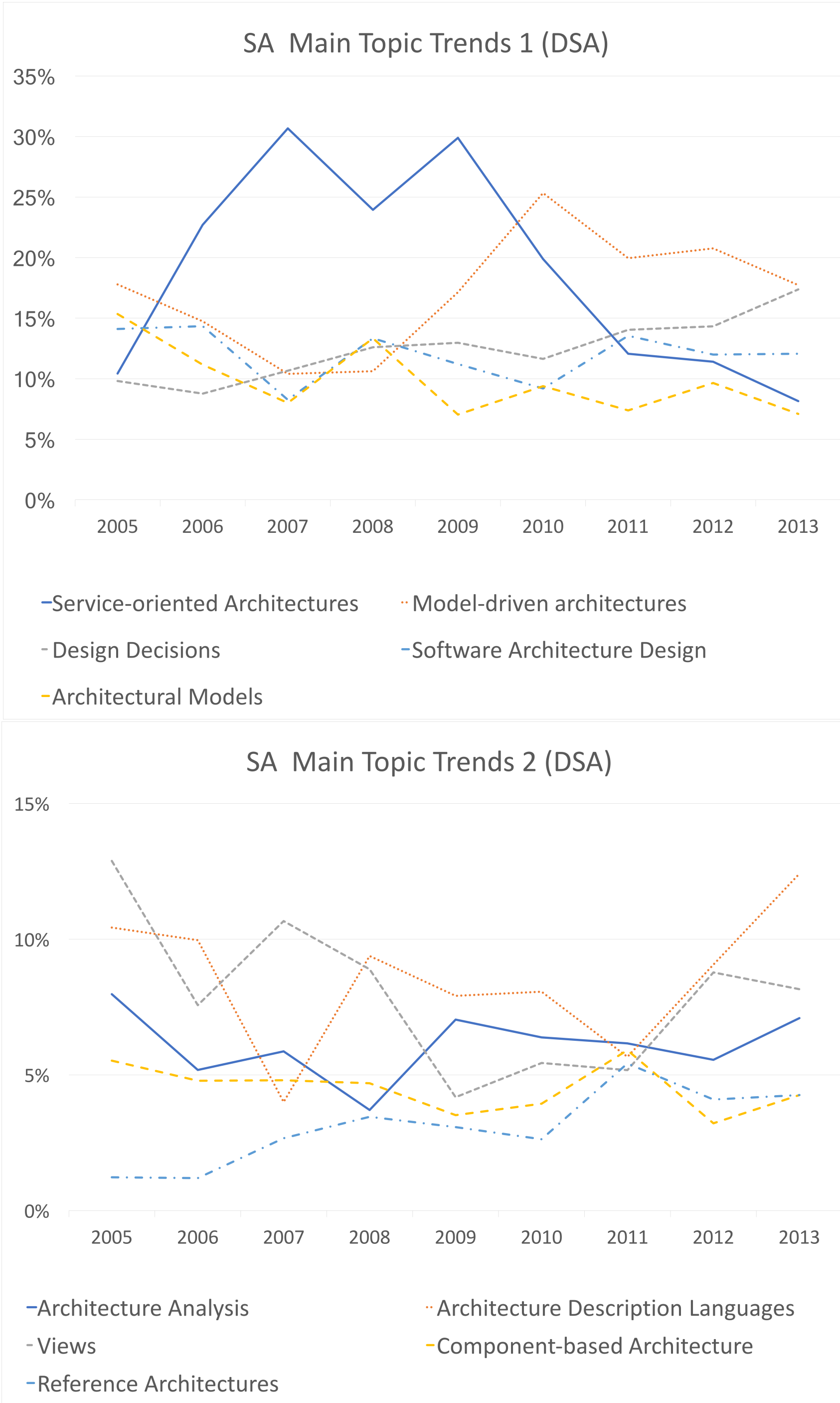

Fig. 6. Number of publications of the top ten main topics in DSA over time.

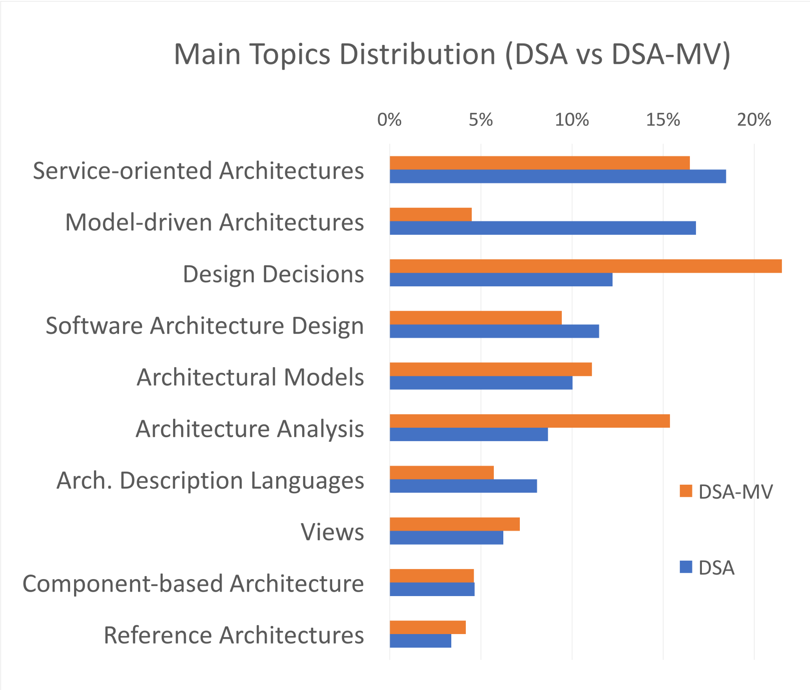

Fig. 7. Comparison DSA and DSA-MV in terms of topic distribution. The percentage value refers to the ratio between the number of publications in a topic and the total publications in the ten main topics.

Table 2 shows the agreement between the annotators. It was computed by calculating the ratio of papers which were tagged with the same category by both annotator. EDAM has the highest average agreement and it also yields the highest agreement with three out of six users. User5 does even better in this regards and has the highest agreement with four annotators.

Running the chi-square test on the human users shows that their behaviors are significantly different ($p = 0.017$). However, if we group together users $\{2, 3, 5, 6\}$ and users $\{1, 4\}$, the intra-group behavior is not significantly different ($p = 0.81$, $p = 0.38$), while the inter-group behavior is very different ($p = 0.0007$). Interestingly, users $\{1, 4\}$ were two students at the beginning of their PhD, hence still relatively new to the domain. This could suggest the importance of considerable domain experience for this task. EDAM exhibits a behavior consistent with the most senior group, from which it is not significantly different ($p = 0.77$).

As anticipated, a good way to measure the performance of annotators is their agreement with the majority of other expert users.

Figure 9 shows the percentage of annotations of each annotator that agree with other *n* annotators. EDAM agrees with four out of six human annotators for 68% of the studies, it agrees with at least three of them for 80% of the studies, and it agrees with at least one of them for all the studies but one. Indeed, the categories generated by EDAM coincide with the ones suggested by the relative majority of users in 84% of the cases. Therefore, EDAM's performance is comparable to the performance of the two annotators (User5 and User3) with the highest agreement with the user majority. In addition, EDAM always agrees with the majority for the studies in which no more than one annotator disagrees. It thus seems to perform well in handling simple not-ambiguous papers, that nonetheless human experts may sometimes get wrong.

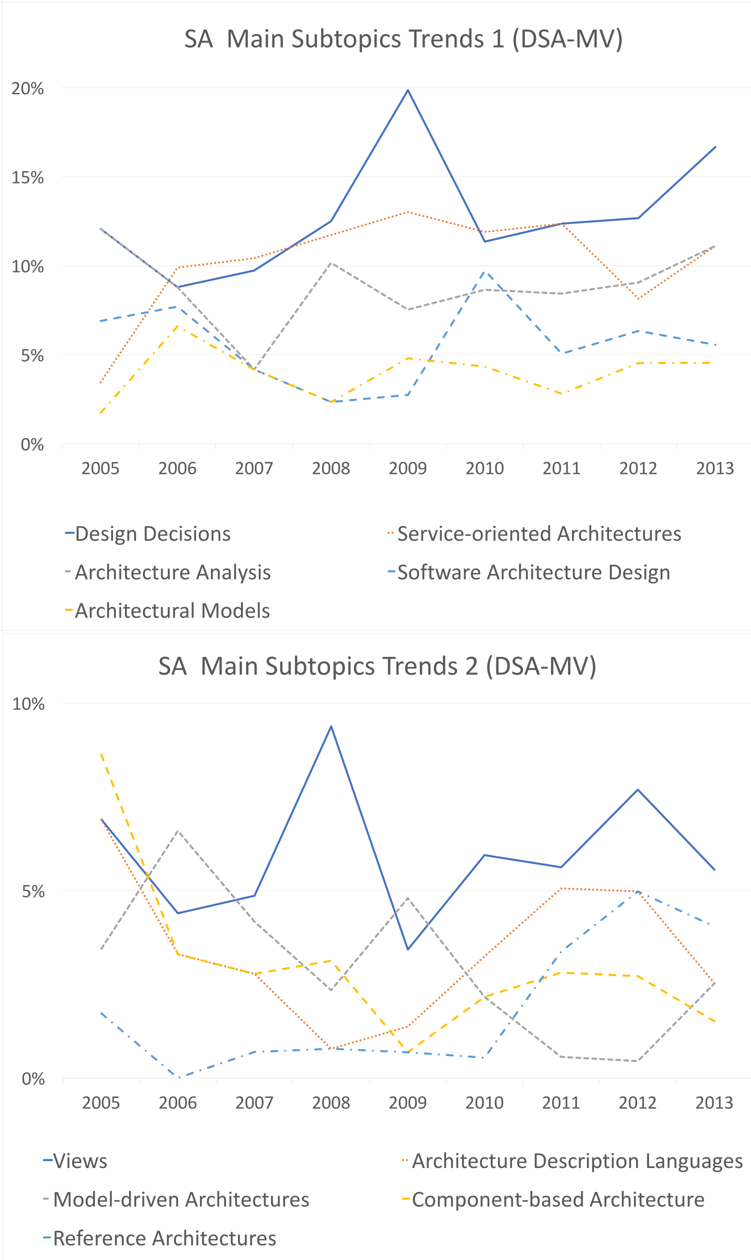

Fig. 8. Number of publications of the top ten main topics in DSA-MV over time.

In conclusion, this study suggests that the EDAM classification step generates annotations that agree with the majority of human experts and are not statistically different from the ones produced by the senior group.

Naturally, EDAM performance may change according to the quality of the ontology and the domain knowledge of the human users that refined it. EDAM is not an alternative to human experts, rather a methodology that allows humans to annotate on a larger scale, by defining a sound domain knowledge

	EDAM	User1	User2	User3	User4	User5	User6
EDAM		56%	68%	64%	64%	**76%**	64%
User1	**56%**		40%	**56%**	36%	48%	44%
User2	68%	40%		64%	52%	**76%**	64%
User3	64%	56%	64%		52%	64%	**68%**
User4	**64%**	36%	52%	52%		**64%**	52%
User5	**76%**	48%	76%	64%	64%		72%
User6	64%	44%	64%	68%	52%	**72%**	
Av. Agreement	**66%**	45%	58%	59%	51%	63%	60%

Table 2

Agreement between annotators (including EDAM) and average agreement of each annotator. In bold the best agreements for each annotator.

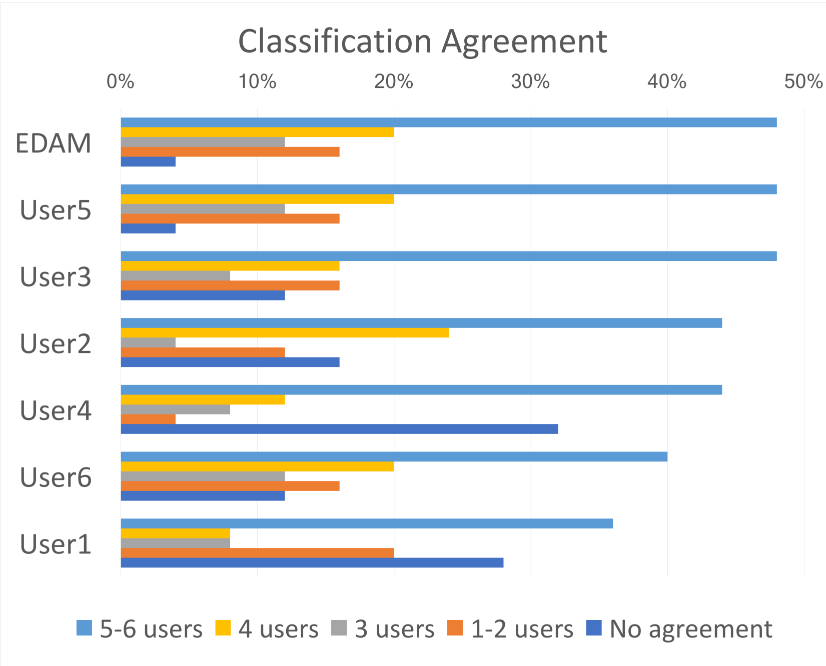

Fig. 9. Percentage of annotations that agree with other *n* annotators.

and a mapping function. However, this preliminary example application already shows very promising results.

5.2. Limitations

In this section we discuss EDAM limitations based on the categorization given in [45].

For *internal validity* we have identified two main threats that regard the generation of a reliable ontology, which is key to select relevant studies that directly fulfill the selection criteria (and hence correspond to the primary studies for the study at hand). In particular:

Ontology learning (step 3): hierarchy is important. The domain ontology, automatically inferred by the ontology learning technique, is structured hierarchically. Therefore, an area marked as *subarea* (e.g., architecture description languages) is subsumed by the previous area at the upper level of the taxonomy (e.g., software architecture). *Deeper hierarchies bring finer-grained topics, and therefore a higher precision in the classification process.*

During the application of ontology learning techniques to various research areas (not reported in this paper for the sake of brevity) we found that current ontology learning methods usually identify only mature (in terms of number of publications) research areas. Emerging topics may be excluded, thus reducing the granularity of recent fields' ontologies.

To alleviate this problem, human experts may be asked to manually identify the most recent areas and to possibly adopt ontology forecasting techniques [5]. Therefore, the role of experts in improving the quality and deepness of the hierarchy is indeed critical. For the sake of this study, aimed at showing the advantages of automation, the relatively small number of experts was acceptable. However, a larger and more diversified pool of experts should be involved when the research area under investigation is broader.

Ontology refinement (step 4): experience matters. As illustrated in Figure 2, EDAM requires human expertize to refine the automatically generated ontology (step 4). This task is not always straightforward, since humans can have different views on the foundational conceptual elements characterizing a certain discipline. Those differences may be related to many factors, such as the researcher's exposure to the research area under investigation, seniority, broad vs. specialized knowledge on specific sub-disciplines. Our preliminary experiments allows us to conclude that senior domain experts, with a mature yet wide view on the research area under investigation, should be selected to minimize this threat.

The main threats for *external validity* regard the practical exploitation of EDAM. In particular:

Scholarly dataset: different research areas require different datasets. This paper reports on our experience with EDAM's application to the software architecture research area. Since the domain of software engineering is well represented in the Scopus Computer Science dataset, we are not facing generalizability issues. However, moving to a totally different domain would require to take into account (assuming to have access to) different scholarly datasets.

Unfortunately, finding up-to-date datasets of scholarly data covering the field under analysis is not always easy and this could be a threat to our approach. Nonetheless, the movement toward open access is helping in mitigating this issue by making available a variety of datasets containing machine-readable data about scientific publications, e.g., CORE[30], OpenCitations[31], DBLP[32], ScolarlyData.org[33], Nanopub.org[34], and others.

Tool support: closed-source tools. EDAM is making use of some closed-source, proprietary tools for running some of the tasks. This may reduce the application of our approach from other research groups. In order to mitigate this threat, we are planning to release a web service accessible by other colleagues interested to carry out an EDAM study.

[30]https://core.ac.uk
[31]http://opencitations.net/
[32]http://dblp.uni-trier.de/
[33]http://www.scholarlydata.org/
[34]http://nanopub.org/

Research Questions: some may not be automatized. Many research questions that are typical of mapping studies can be answered by producing relevant analytics [45], e.g., by counting the number of publications, authors, and venues associated with certain topics in subsequent years. However, some more complex research questions may still require domain experts to manually analyse the relevant studies, e.g., for classifying them in categories that a state of the art classifier would be unable to detect with good accuracy. This is an inherent limitation of the methodology. Nonetheless in many of these cases a preliminary classification by an automatic system may still alleviate the expert work load, e.g., by reducing the set of publications that need to be manually analysed. In addition, the performance of entity extraction and linking tooks is steadily improving [3, 13, 33], allowing to extract increasingly better representations of research knowledge from scientific articles. Therefore, the number of research questions that can be addressed algorithmically may increase over the following years.

5.3. Implications for Systematic Mappings

There are a few implications that can potentially change the way we perform systematic mapping studies in software engineering. As mentioned in Section 4, these implications regard:

Scalability: size does not matter anymore. EDAM can process a potentially endless set of publications. This allows e.g., mapping studies to be based on *all* relevant primary studies, previously scoped down due to the fact that humans could not manually process hundreds or thousands of papers.

Objectivity: the automatic classification is less biased. The automatic classification of primary studies does not suffer from the biases of specific human annotators. Nonetheless, the quality of the classification appears on par with the one produced by the human annotators.

Reproducibility: study duplication and extension is easy. Thanks to EDAM, replicating or extending studies, either by the same researcher or by someone else, requires simple tuning, e.g., to extend the publication period, or to select different views illustrating the publication trends of interest.

Granularity of the study: zooming-in and -out is simpler. Thanks to the fact that the selection and classification of primary studies is based on an domain ontology, and of course to automation, EDAM allows to tune the depth of the classification the researcher desires in a given research area. Such tuning just requires setting the level of categories and sub-categories to be included in the classification, and then re-run the methodology.

5.4. Reusing EDAM for other Systematic Reviews

EDAM can be applied to any domain of interest and for different types of studies. The scenarios that we envisage are discussed below and illustrated in Figure 10. They are: S1) Application of EDAM to a *new* application domain, S2) Mapping study *replication*, S3) Mapping study *refinement*, and S4) *Systematic literature review*.

Application of EDAM to a new application domain (S1). In the basic scenario (**S1**), the ontology for the new application domain is not yet available. In this case, the complete process illustrated in Figure 2 (and emphasized in Figure 10.(S1)) shall be applied. This is the scenario followed in the work presented in this article. It is applicable while investigating a new domain notwithstanding its specific characteristics.

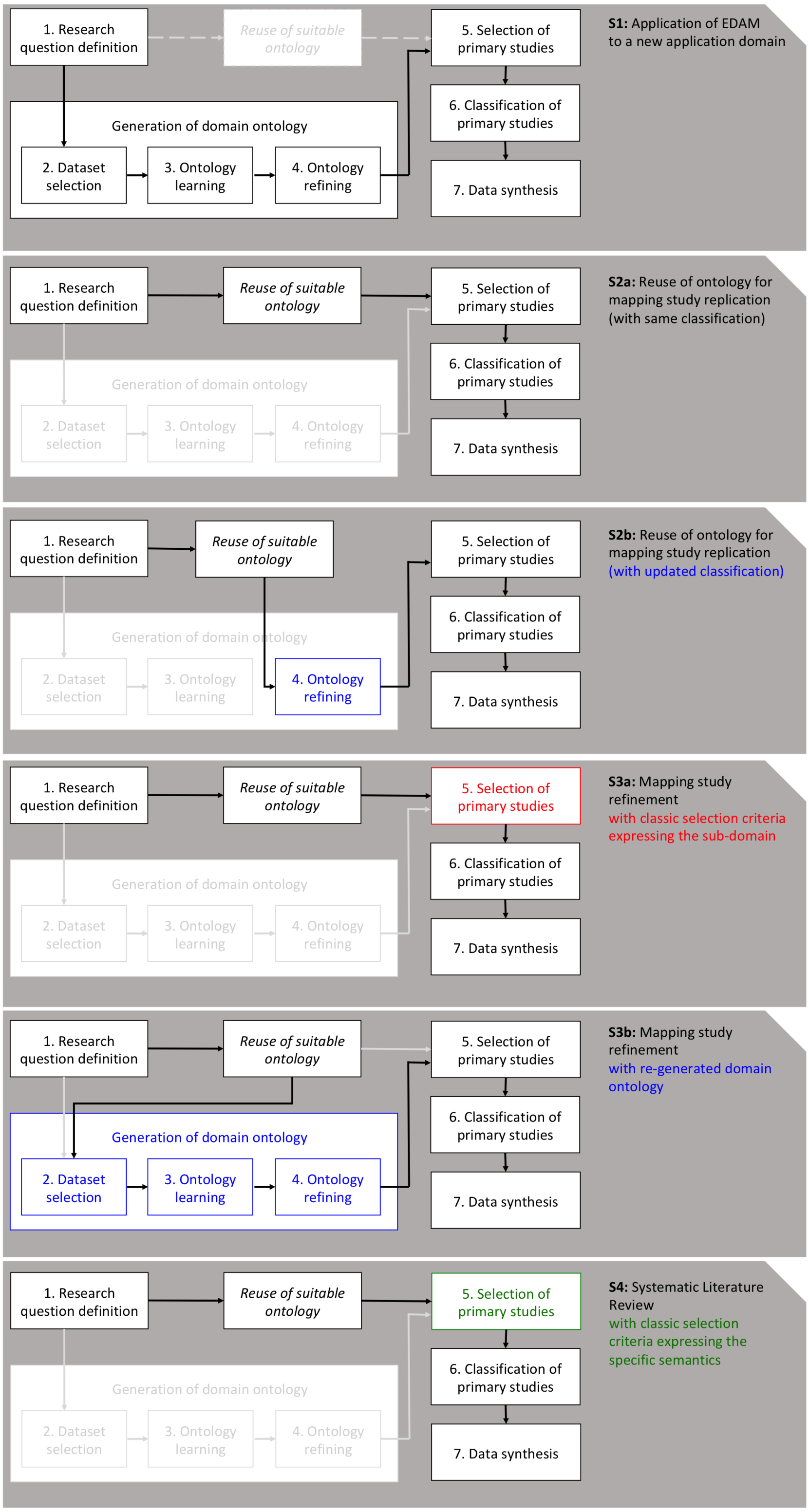

Fig. 10. Possible EDAM applications.

If instead a researcher wants to perform a SR in a domain for which the ontology already exists (scenario S2), such generated domain ontology can be *reused* in the following two ways, depending on the specific study goal:

Mapping Study Replication (same classification, S2a). Suppose we want to replicate a pre-existing EDAM mapping study conducted at time t_0, in order to update the list of primary studies and related analysis at time t_1 (e.g., update in year 2020 the study on Software Architecture presented in this paper). In this case, we can directly reuse the previously generated ontology (cf. Figure 10.(S2a)). The list of (updated) primary studies can be automatically re-calculated (in step 5) and used (in step 6) for classification and analysis purposes. Notice, however, that this scenario does not address the potential need to *update* the list of topics. Such a scenario is covered below.

Mapping Study Replication (updated classification, S2b). Differently from scenario S2a, we may be interested to replicate a pre-existing study *and* also include any new topics that may have emerged in the period between time t_0 and time t_1 (e.g., updating this study in year 2020 while including new topics appeared after this study). This need requires an update of the domain ontology; therefore, the process in Figure 10.(S2b) must be run from step 4 onward.

Another scenario (S3) accommodates the case in which we want to *refine* the classification and analysis conducted as a mapping study. In the current approach, as shown in the Software Architecture domain scenario, step 5 in Figure 2 returns a set of primary studies that can be further classified into sub-domains (e.g., Architectural Styles, being one element of our ontology, can be further refined to discover all the papers that cover selected styles). We identify two sub-scenarios in order to provide a refinement of sub-domains contents:

Mapping Study Refinement with classic selection criteria (S3a). In this scenario, one may classify the articles into sub-domains of interest by applying the inclusion and exclusion criteria [17] to the primary studies selected in step 5 of EDAM. For example, knowing that Publish-Subscribe, Client-Server, and Event-driven are sub-domains of Architectural Styles, we introduce selection criteria to position Architectural Styles articles into those categories. This approach allows us to zoom into a specific sub-domain of interest and extract the articles fitting in the specific target sub-domain.

Mapping Study Refinement with re-generated domain ontology (S3b). The selected sub-domain of interest may contain hundreds of papers (for example, the Design Decisions sub-domain in our study includes 428 papers). Consequently, applying the selection criteria reported in scenario S3a may be cumbersome, requiring the manual analysis of most of those papers. Alternatively, the researcher may execute an additional round of steps 2-4 to refine the domain ontology for the specific sub-domain (cf. Figure 10.(S3b)). This scenario is similar to S1, but applied to a specific sub-domain of interest.

A fourth scenario sees the researcher is interested to run a systematic literature review (SLR) on specific research questions:

Systematic Literature Reviews (S4). In step 5 (cf. Figure 10.(S4)), given the list of primary studies generated based on the existing ontology, we may run the *classic* SLR approach [16] to select those papers that fit with the research questions of interest. Differently from scenario S3a, S4 adds the semantics beyond the definition of the domain, and encapsulated into the research questions and the corresponding selection criteria. E.g., given the list of all studies on software architecture styles, one may want to perform an SLR to analyze those approaches that are adopted in industrial settings.

6. Conclusions and Future Work

In this paper we have presented EDAM, an expert-driven automated methodology to assist systematic reviews. Its application to the software architecture research area shows preliminary and very promising results.

Motivated by the large amount of time and effort needed by classic methodologies to select and classify the primary studies, EDAM offers benefits that can help SE researchers to dedicate most of their time to the most cognitive-intensive tasks like e.g., interpretation of the trends and extraction of lessons and research gaps.

Additional benefits have been emphasized in Section 4.1 (after presenting EDAM) and Section 5.3 (discussing implications for systematic mappings). Among the benefits we mention the great potential for re-using EDAM and in particular domain ontologies and functions to build a shared framework helping the research community at large. Much can be done in this direction.

Our next step is to complement EDAM with automated forward snowballing to further reduce the effort for identifying relevant primary studies. With the same goal, we are planning to investigate other possible data synthesis techniques through machine learning techniques or the (manual) intervention of human experts. Last, but most important for us, we plan to reconstruct the 25 years of the software architecture body of knowledge by fully exploiting EDAM automation and human expertise.

Acknowledgments

The authors would like to thank the colleagues which donated their time and expertise by contributing to this study as domain experts and/or annotators: Paris Avgeriou, Barbora Buhnova, Rafael Capilla, Jan Carlson, Ivica Crnkovic, John Grundy, Rich Hilliard, Heiko Koziolek, Anton Jansen, Ivano Malavolta, Leonardo Mariani, Marina Mongiello, Matthias Naab, Patrizio Pelliccione, Mohammad Sharaf, Damian Andrew Tamburri, Antony Tang, Jan Martijn van der Werf, Smrithi Rekha Venkatasubramanian, Rainer Weinreich, Danny Weyns, Eoin Woods, and Uwe Zdun.

We also thank Davide Falessi for reviewing an earlier version of this manuscript, and Elsevier BV for providing us with access to its large repository of scholarly data.

References

[1] Ahmed Al-Zubidy, Jeffrey C Carver, David P Hale, and Edgar E Hassler. Vision for SLR tooling infrastructure: Prioritizing value-added requirements. *Information and Software Technology*, 91:72–81, November 2017.

[2] Rubayyi Alghamdi and Khalid Alfalqi. A survey of topic modeling in text mining. *I. J. ACSA*, 6(1):147–153, 2015.

[3] Isabelle Augenstein, Sebastian Padó, and Sebastian Rudolph. Lodifier: Generating linked data from unstructured text. In *Extended Semantic Web Conference*, pages 210–224. Springer, 2012.

[4] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3 (Jan):993–1022, 2003.

[5] Amparo Elizabeth Cano-Basave, Francesco Osborne, and Angelo Antonio Salatino. Ontology forecasting in scientific literature: Semantic concepts prediction based on innovation-adoption priors. In *Knowledge Engineering and Knowledge Management: 20th International Conference, EKAW 2016, Bologna, Italy, November 19-23, 2016, Proceedings 20*, pages 51–67. Springer, 2016.

[6] Philipp Cimiano and Johanna Völker. text2onto. In *International Conference on Application of Natural Language to Information Systems*, pages 227–238. Springer, 2005.

[7] Fabio Q B da Silva, Marcos Suassuna, A César C França, Alicia M Grubb, Tatiana B Gouveia, Cleviton V F Monteiro, and Igor Ebrahim dos Santos. Replication of empirical studies in software engineering research: a systematic mapping study. *Empirical Software Engineer*, 19(3):501–557, June 2014.

[8] Jorge Calmon de Almeida Biolchini, Paula Gomes Mian, Ana Candida Cruz Natali, Tayana Uchôa Conte, and Guilherme Horta Travassos. Scientific research ontology to support systematic review in software engineering. *Advanced Engineering Informatics*, 21(2):133–151, 2007.

[9] Rafael Maiani De Mello and Guilherme Horta Travassos. Surveys in software engineering: Identifying representative samples. In *Proceedings of the 10th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, ESEM '16, pages 55:1–55:6, New York, NY, USA, 2016. ACM.

[10] Tommaso Di Noia, Roberto Mirizzi, Vito Claudio Ostuni, Davide Romito, and Markus Zanker. Linked open data to support content-based recommender systems. In *Proceedings of the 8th International Conference on Semantic Systems*, pages 1–8. ACM, 2012.

[11] Katia Romero Felizardo, Emilia Mendes, Marcos Kalinowski, Érica Ferreira Souza, and Nandamudi L Vijaykumar. Using forward snowballing to update systematic reviews in software engineering. In *Proceedings of the 10th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, page 53. ACM, September 2016.

[12] Bojan Filipovic, Boris Van Lindschoten, Giuseppe Procaccianti, and Patricia Lago. *Systematic Literature Study on Sustainable Software*. VU Technical Report, 2 2017.

[13] Aldo Gangemi, Valentina Presutti, Diego Reforgiato Recupero, Andrea Giovanni Nuzzolese, Francesco Draicchio, and Misael Mongiovì. Semantic web machine reading with fred. *Semantic Web*, 8(6):873–893, 2017.

[14] Edgar Hassler, Jeffrey C Carver, David Hale, and Ahmed Al-Zubidy. Identification of SLR tool needs – results of a community workshop. *Information and Software Technology*, 70:122–129, 2016.

[15] Barbara Kitchenham. Procedures for performing systematic reviews. *Keele, UK, Keele University*, 33(2004):1–26, 2004.

[16] Barbara Kitchenham and Pearl Brereton. A systematic review of systematic review process research in software engineering. *Information and software technology*, 55(12):2049–2075, 2013.

[17] Barbara A Kitchenham and Stuart Charters. Guidelines for performing systematic literature reviews in software engineering, 2007.

[18] Petr Knoth and Zdenek Zdrahal. Core: three access levels to underpin open access. *D-Lib Magazine*, 18(11/12), 2012.

[19] Marco Kuhrmann, Daniel Méndez Fernández, and Maya Daneva. On the pragmatic design of literature studies in software engineering: an experience-based guideline. *Empirical Software Engineer*, pages 1–40, 6 January 2017.

[20] Xueqing Liu, Yangqiu Song, Shixia Liu, and Haixun Wang. Automatic taxonomy construction from keywords. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1433–1441. ACM, 2012.

[21] Christopher Marshall, Pearl Brereton, and Barbara Kitchenham. Tools to support systematic reviews in software engineering: a cross-domain survey using semi-structured interviews. In *Proceedings of the 19th International Conference on Evaluation and Assessment in Software Engineering*, page 26. ACM, April 2015.

[22] Pablo N Mendes, Max Jakob, Andrés García-Silva, and Christian Bizer. Dbpedia spotlight: shedding light on the web of documents. In *Proceedings of the 7th international conference on semantic systems*, pages 1–8. ACM, 2011.

[23] Jefferson Seide Molleri, Kai Petersen, and Emilia Mendes. Survey guidelines in software engineering: An annotated review. In *Proceedings of the 10th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement - ESEM '16*, pages 1–6. ACM Press, 2016.

[24] Erica Mourão, Marcos Kalinowski, Leonardo Murta, Emilia Mendes, and Claes Wohlin. Investigating the use of a hybrid search strategy for systematic reviews. In *Proceedings of the 11th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, ESEM '17, pages 193–198. IEEE Press, 2017.

[25] Fábio R Octaviano, Katia R Felizardo, José C Maldonado, and Sandra C P. Semi-automatic selection of primary studies in systematic literature reviews: is it reasonable? *Empirical Software Engineer*, 20(6):1898–1917, 2015.

[26] Francesco Osborne and Enrico Motta. Mining semantic relations between research areas. In *International Semantic Web Conference*, pages 410–426. Springer, 2012.

[27] Francesco Osborne and Enrico Motta. Klink-2: integrating multiple web sources to generate semantic topic networks. In *International Semantic Web Conference*, pages 408–424. Springer, 2015.

[28] Francesco Osborne, Enrico Motta, and Paul Mulholland. Exploring scholarly data with rexplore. In *International semantic web conference*, pages 460–477. Springer, 2013.

[29] Francesco Osborne, Angelo Salatino, Aliaksandr Birukou, and Enrico Motta. Automatic classification of springer nature proceedings with smart topic miner. In *International Semantic Web Conference*, pages 383–399. Springer, 2016.
[30] Kai Petersen, Robert Feldt, Shahid Mujtaba, and Michael Mattsson. Systematic mapping studies in software engineering. In *Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering*, EASE, pages 68–77, Swinton, UK, UK, 2008. British Computer Society.
[31] Kai Petersen, Sairam Vakkalanka, and Ludwik Kuzniarz. Guidelines for conducting systematic mapping studies in software engineering: An update. *Information and Software Technology*, 64:1–18, 2015.
[32] Giulio Petrucci, Chiara Ghidini, and Marco Rospocher. Ontology learning in the deep. In *Knowledge Engineering and Knowledge Management: 20th International Conference, EKAW 2016, Bologna, Italy, November 19-23, 2016, Proceedings 20*, pages 480–495. Springer, 2016.
[33] Giuseppe Rizzo and Raphaël Troncy. Nerd: a framework for unifying named entity recognition and disambiguation extraction tools. In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 73–76. Association for Computational Linguistics, 2012.
[34] Rasmus Ros, Elizabeth Bjarnason, and Per Runeson. A machine learning approach for semi-automated search and selection in literature studies. In *Proceedings of the 21st International Conference on Evaluation and Assessment in Software Engineering*, EASE'17, pages 118–127. ACM, 2017. ISBN 978-1-4503-4804-1.
[35] Hassan Saif, Yulan He, and Harith Alani. Semantic sentiment analysis of twitter. *The Semantic Web–ISWC 2012*, pages 508–524, 2012.
[36] Angelo A Salatino, Francesco Osborne, and Enrico Motta. How are topics born? understanding the research dynamics preceding the emergence of new areas. *PeerJ Computer Science*, 3:e119, 2017.
[37] Angelo A Salatino, , Thanapalasingam Thiviyan, Andrea Mannocci, Francesco Osborne, and Enrico Motta. The computer science ontology: A large-scale taxonomy of research areas. In *Proceedings of the International Semantic Web Conference 2018*, ISWC'18, 2018.
[38] Mark Sanderson and Bruce Croft. Deriving concept hierarchies from text. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 206–213. ACM, 1999.
[39] J Michael Schultz and Mark Liberman. Topic detection and tracking using idf-weighted cosine coefficient. In *Proceedings of the DARPA broadcast news workshop*, pages 189–192. San Francisco: Morgan Kaufmann, 1999.
[40] Yueming Sun, Ye Yang, He Zhang, Wen Zhang, and Qing Wang. Towards evidence-based ontology for supporting systematic literature review. 2012.
[41] Tassio Vale, Ivica Crnkovic, Eduardo Santana de Almeida, Paulo Anselmo da Mota Silveira Neto, Yguarata Cerqueira Cavalcanti, and Silvio Romero de Lemos Meira. Twenty-eight years of component-based software engineering. *Journal of Systems and Software*, 111(1):128 – 148, 2016. ISSN 0164-1212.
[42] Roel J Wieringa. *Design Science Methodology for Information Systems and Software Engineering:*. Springer Berlin Heidelberg, 2014.
[43] Mark D Wilkinson, Michel Dumontier, IJsbrand Jan Aalbersberg, Gabrielle Appleton, Myles Axton, Arie Baak, Niklas Blomberg, Jan-Willem Boiten, Luiz Bonino da Silva Santos, Philip E Bourne, et al. The fair guiding principles for scientific data management and stewardship. *Scientific data*, 3:160018, 2016.
[44] Gerhard Wohlgenannt, Albert Weichselbraun, Arno Scharl, and Marta Sabou. Dynamic integration of multiple evidence sources for ontology learning. *Journal of Information and Data Management*, 3(3):243, 2012.
[45] C. Wohlin, P. Runeson, M. Höst, M.C. Ohlsson, B. Regnell, and A. Wesslén. *Experimentation in Software Engineering*. Computer Science. Springer, 2012.
[46] Claes Wohlin and Rafael Prikladnicki. Systematic literature reviews in software engineering. *Information and Software Technology*, 55(6):919–920, 2013.
[47] Claes Wohlin, Per Runeson, Paulo Anselmo da Mota Silveira Neto, Emelie Engström, Ivan do Carmo Machado, and Eduardo Santana de Almeida. On the reliability of mapping studies in software engineering. *The Journal of systems and software*, 86(10):2594–2610, October 2013.
[48] Nina J.E. Wolfram, Patricia Lago, and Francesco Osborne. Sustainability in software engineering. In *IFIP Conference on Sustainable Internet and ICT for Sustainability (SustainIT)*, December 2017.
[49] He Zhang and Muhammad Ali Babar. Systematic reviews in software engineering: An empirical investigation. *Information and Software Technology*, 55(7), 2013. ISSN 0164-1212.
[50] He Zhang, Muhammad Ali Babar, and Paolo Tell. Identifying relevant studies in software engineering. *Information and Software Technology*, 53(6):625–637, 2011.