On Different Search Methods for Systematic Literature Reviews and Maps

Experiences from a Literature Search on Validation and Verification of Emergent Behavior

Jennifer Brings, Marian Daun, Markus Kempe, Thorsten Weyer
University of Duisburg-Essen,
paluno – The Ruhr Institute for Software Technology
45127 Essen, Germany
{jennifer.brings, marian.daun, thorsten.weyer}@paluno.uni-due.de

ABSTRACT

[Background] Systematic literature reviews and maps have become well-established research methods in software engineering research. Of the three commonly suggested and used search methods: manual search, database search, or snowball search; systematic literature reviews and maps typically employ one or a combination of two or three of those as their search strategy. As systematic literature reviews and maps raise a claim to result in a representative set of relevant papers for a certain area of investigation, it is of importance to understand the impact the search strategy has on achieving this goal. [Aim] This paper contributes a study to compare all three search methods. This study aims at providing evidence as to what advantages and disadvantages of these three search methods are. [Method] We conducted three systematic literature reviews on the same topic, which affects multiple software engineering related disciplines, using different search methods, while keeping other parameters like inclusion and exclusion criteria consistent among all three reviews. [Results] Our results show a similar effectiveness for snowball and database search and the highest efficiency for database searches. However, our literature reviews led to three barely overlapping sets of papers, which in turn led to distinct impressions of the same field. [Conclusion] Our results show that the use of a single search method can lead to a set of included papers, which misrepresents the research field under investigation. Hence, particularly when conducting literature reviews that affect different software engineering sub-disciplines and related disciplines, researchers should not just rely on the single most effective and/or efficient search method.

EASE'18, June 28–29, 2018, Christchurch, New Zealand

ACM ISBN 978-1-4503-6403-4/18/06…$15.00
https://doi.org/10.1145/3210459.3210463

CCS CONCEPTS

• **General and reference → Surveys and overviews**
• **General and reference → Empirical studies**

KEYWORDS

Systematic literature review, search method, experience report

ACM Reference format:

J. Brings, M. Daun, M. Kempe, T. Weyer. 2018. On Different Search Methods for Systematic Literature Reviews and Maps: Experiences from a Literature Search on Validation and Verification of Emergent Behavior. In *22nd International Conference on Evaluation and Assessment in Software Engineering 2018 Proceedings*, 11 pages.
DOI: 10.1145/3210459.3210463

1 INTRODUCTION

In recent years systematic literature reviews and maps have become popular research methods in software engineering research. Systematic literature reviews answer specific research questions and synthesize the results of the primary studies, while mapping studies identify gaps and clusters in the literature by classifying existing research. [23]. Systematic literature reviews and maps are seen as more thorough and less biased than serendipitous searches. Furthermore, systematic literature reviews and maps facilitate the repeatability of the search and, thereby, the verifiability of the results [23].

A key characteristic of systematic literature reviews and maps is the a priori definition of a search strategy [23]. Ideally, a search strategy should be chosen that finds all relevant and no irrelevant publications. Search strategies can comprise different search methods. To conduct a manual search, selected journals and conference proceedings are scanned for relevant publications. While this ensures discovering all relevant publications therein, findings published elsewhere go overlooked. Database searches are conducted by defining one or more search strings and applying these to one or more database. The main challenge in conducting database searches is the definition of one or more appropriate search strings. Too narrow search strings exclude discovering relevant publications, while too broad search strings lead to many irrelevant search results. Backward and forward searches, also known as snowball

searches, are conducted by investigating reference sections and citations of relevant publications.

All three search methods are commonly suggested for different purposes [20]. Additionally, combinations of the different search methods are proposed as well [16, 18]. As literature searches are time consuming, it is of interest to investigate which search method is most efficient to use while still arriving at a representative set of papers. Furthermore, there is a risk that, depending on the search method, the field under investigation is misrepresented. This risk materializes in literature reviews and maps when conclusions are drawn and claims are made about a field that are not true but appear substantiated due to the systematic approach towards the literature study. While there have been other related studies on this question (see Section 2), there is still a need for more insights into which technique is advantageous under which circumstances (cf. [3]). Therefore, this paper complements existing works by investigating advantages and disadvantages of different search methods for topics relevant in multiple software engineering related disciplines with partly different terminology.

Many software engineering problems are investigated in different software engineering sub-disciplines, and techniques developed in one sub-discipline can be adapted to be used in another. Furthermore, software engineering research is often closely related to other computer science sub-disciplines. Hence, systematic literature reviews and maps must often cope with different terminology used in the different software engineering related disciplines. So is the case for the literature reviews conducted for this paper. To investigate the usefulness of the three search approaches, we conducted three systematic literature reviews on the topic of validation and verification of emergent behavior using the three different search approaches, while keeping all other factors consistent. Validation and verification of emergent behavior is, among others, relevant in the research field of formal methods, model-based development, requirements engineering, artificial intelligence, and systems engineering.

This paper contributes not only a comparison of all three search methods in terms of effectiveness and efficiency, but also gives new insights into whether literature reviews employing only one search method can lead to a misrepresentation of the field.

The paper is outlined as follows. Section 2 discusses related work. In Section 3 we provide a detailed description of our research method, i.e., the three systematic literature reviews. Results of the systematic literature reviews are presented in Section 4, which are discussed in Section 5. Finally, Section 6 concludes this paper.

2 RELATED WORK

Effectiveness and efficiency, and particularly reliability, and repeatability of literature studies have been invested in the past (e.g., [3, 17, 19, 46]). For instance, MacDonell et al. [30] compared two systematic literature reviews conducted by different researchers on the same research questions. They concluded that in this case systematic literature reviews were robust against differences in the reviewing process and different reviewing researchers. However, other studies have shown that the search and selection process has an impact on the papers found by a literature study (e.g., [19]).

Resulting from such research, guidelines have been defined for systematic literature studies (e.g., [18, 23, 27, 33]). In these guidelines, it is commonly acknowledged that the search strategy is an important part of thorough systematic literature reviews (cf. [22]) and systematic mapping studies (cf. [33]). For instance, Greenhalgh and Peacock [14] found that systematic reviews of complex evidences cannot entirely be based on database searches alone. Skoglund and Runeson [39] came to a similar result; they found database searches to be satisfactory for narrow research questions, while literature reviews in large search areas that solely rely on database searches typically lead to a large number of missed paper.

Jalali and Wohlin [17] compared snowballing with database searches and concluded that both methods seem adequate and that the results of their literature review are not dependent on the search method. Yet snowballing might be preferable if only general terms can be defined for database searches. A replication by Wohlin [45] supported these results and concluded that snowballing is a good alternative to database searches. Also in favor of snowballing Wohlin et al. [46] conclude that snowballing is advantageous for broad literature searches, as in their case snowballing found the papers from the database search as well as additional papers not found by database search. Badampudi et al. [3] also compare database search and snowballing, coming to similar results.

In contrast to the aforementioned related works, our results indicate that for broad search areas (i.e. when multiple software engineering sub-disciplines need to be considered relevant) database search and snowballing (as well as manual search) did not lead to a similar set of papers as shown by [17] and [45]. Thus, there exists a risk of misrepresenting the field depending on the search method. Furthermore, we will show that, unlike stated in [14] and [39], snowballing is not universally advantageous for broad literature searches, as we will show that also database search was able to find a considerable amount of paper not found by snowballing.

While there exist additional related work dealing with improving and applying snowballing (e.g., [10]) or data base search (e.g., [48]); or even with combining different search methods (e.g., [16, 18]), we will not place emphasis on the optimization of search strategies in this paper.

3 RESEARCH METHOD

Three systematic literature reviews, each employing a different search method, were undertaken. The goal of each literature review was to identify literature relevant for the validation and verification of emergent behavior. The goal of conducting three literature reviews on the same topic was to investigate different outcomes of the used search methods. This section provides details on the used research method.

3.1 Research Questions

We defined three major research questions to investigate the effects of different search methods. Effectiveness of a search strategy is an important factor, as it describes how complete the results of a search are. However, determining the true effectiveness of a search is impossible, as we cannot measure the number of papers that could potentially to be found, but only the papers that we actually found. Thus, we define effectiveness as the ratio between the number of papers found by this particular search method and the number of unique papers found by all search methods combined. Hence, we define as research question:

> **RQ1:** Are there differences in the effectiveness of the different search methods w.r.t. the amount of papers found among all search methods?

We determine effectiveness[1] using the following formula:

$$\frac{number\ of\ relevant\ papers\ found\ by\ individual\ search\ method}{number\ of\ relevant\ papers\ found\ by\ all\ search\ methods}$$

Thus, effectiveness is calculated by dividing the number of relevant papers found by an individual search method (i.e. manual, database, or snowball search) by the number of unique papers found by all three search methods combined.

Furthermore, efficiency of a search method is important to ensure that a review can be completed in a reasonable time frame. Hence, we define:

> **RQ2:** Are there differences in the efficiency of the different search methods?

Efficiency[2] was determined using the following formula:

$$\frac{number\ of\ relevant\ papers\ found\ by\ individual\ search\ method}{number\ of\ papers\ investigated\ by\ individual\ search\ method}$$

Thus, efficiency is calculated by dividing the number of relevant papers found by an individual search method (i.e. manual, database, or snowball search) by the number papers investigated by the same search method.

Lastly, we want to determine whether different search methods result in a different understanding of the field. If the selected papers from different literature searches show a field in different lights, i.e. misrepresent the actual state of the art in a field under investigation, the use of a single search method must be avoided. Thus, we define as research question:

> **RQ3:** Do the different search method lead to different conclusions about the studied research field?

To investigate RQ3 we use commonly proposed criteria for mapping studies [33], i.e. publication volumes, most prolific authors, top venues, most cited, most common keywords, and research methods used. This allows us to gain an impression whether the search results differ to a considerable amount.

3.2 Design and Procedure

We conducted three systematic literature reviews on the topic of validating and verifying emergent behavior of collaborative systems using three different search methods. This topic has recently gained interest in the engineering of cyber-physical systems as automation (e.g., autonomous driving, industry automation) makes use of collaborating individual systems to fulfill some greater good [5]. However, in other domains emergent behavior and its validation and verification have also been investigated, for instance, in the domain of multi-agent-systems (e.g., [36]) or in the area of swarm behavior (e.g., [26]). Hence, the topic under investigation is not exclusive to a certain discipline and approaches developed for other application domains might be transferable to one another. Thus, a thorough systematic literature review will need to take these related areas into account.

Figure 1 summarizes the study design. We conducted three systematic literature reviews on the same topic, each employing a different search method. To achieve comparability of the search methods, we kept other factors consistent. For instance, the same inclusion and exclusion criteria have been defined by two senior researchers for all three literature reviews. In this paper, we compare the sets of included papers for each literature review with the final set of included papers consisting of all found papers in all three searches. To create the latter, we combined the single sets and checked that there were no inconsistencies in including or excluding the same paper in two different searches.

For each literature review, two different researchers, all of similar background and equally experienced, evaluated papers on their own. To determine inclusion or exclusion, first each paper's title and abstract were read and if necessary the remainder was perused. Papers were included in the set of relevant papers of the respective literature review if both researchers found the paper relevant and excluded if both found the paper irrelevant. In cases of inconsistent perceptions of the paper's relevance, the paper was discussed among four researchers, the two who conducted the respective search and the two additional senior researchers who defined inclusion and exclusion criteria, time span and venues for the manual search, the search string for the database search, and the start set for the snowball search.

[1] In similar studies sometimes also referred to as reliability or recall.
[2] In similar studies sometimes also referred to as precision.

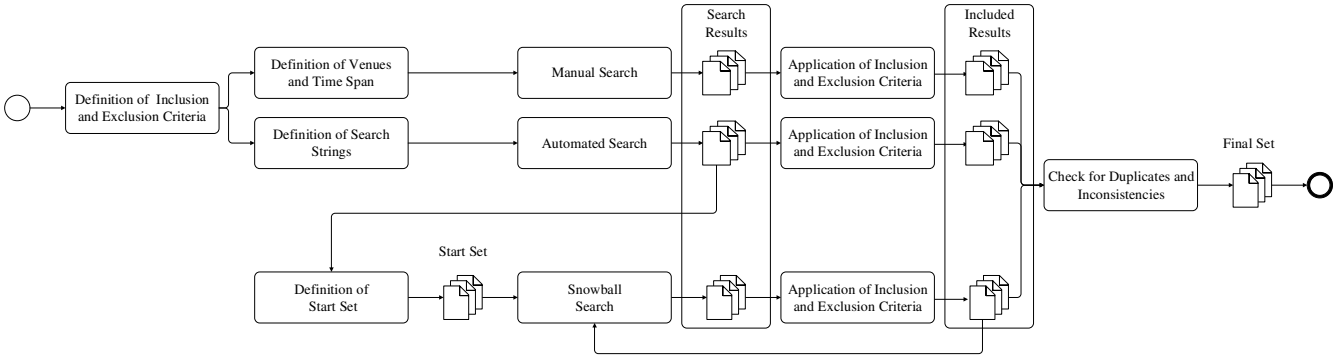

Figure 1: Study design

3.3 Manual Search

Manual search involves investigating all papers published in a given set of venues, for a given set of years. Our manual search covered the years of 2007 to 2017 for the following venues:

- International Conference on Software Engineering (ICSE)
- Foundations of Software Engineering (SIGSOFT FSE)
- IEEE Transactions on Software Engineering (TSE)
- ACM Transactions on Software Engineering and Methodology (TOSEM)

As the topic of the literature search is relevant for multiple software engineering disciplines, we focused our venue selection on general software engineering venues. The chosen venues are often considered the most prestigious for software engineering research and do not focus on a certain sub-discipline of software engineering. During manual search, we investigated 2665 papers.

3.4 Database Search

A database search involves the review of papers returned by a given set of databases using a given set of search parameters. For the database search we used Scopus, as it covers many publishers among them also some of the most common publishers for computer science research, such as the ACM, IEEE, Elsevier, Springer, etc. and unlike Google Scholar allows for filtering non-peer reviewed publications. We did not search any other database as the effort needed to identify and remove duplicates would skew the results for the database search. The search string defined is provided in Listing 1.

The search string was developed based on the literature review's topic and research questions, as is commonly done in systematic reviews [33]. The database search was conducted in October 2017 and generated 487 publications to be considered.

```
TITLE-ABS-KEY ( ( "Verification" OR "Verify" OR "Verifying" ) OR ( "Validation" OR "validate" OR "validating" ) OR "Checking" OR "Proof" OR "Check" OR "Prove" OR "Proving" ) AND TITLE-ABS-KEY ( ( "Emergent behavior" OR "Collaborative behavior" OR "Cooperative behavior" OR "Co-operative behavior" ) OR ( "Emergent behaviour" OR "Collaborative behaviour" OR "Co-operative behaviour" OR "Cooperative behaviour" OR "feature interaction" ) ) AND ( LIMIT-TO ( SUBJAREA , "COMP " ) )
```

Listing 1: Database search string

3.5 Snowball Search

The snowball search as seen in Figure 2 consists of the initial definition of a set of relevant papers and repeated iterations of two subprocesses, forward search and backward search. The backward search yields the set of all papers cited in the relevant papers of the last iteration, the forward search yields the set of citing papers. After both sets have been compiled, results are deemed relevant or irrelevant according to the defined inclusion and exclusion criteria. Both processes and the subsequent evaluation are then repeated for the newly identified relevant papers. A snowball search terminates once an iteration is completed without yielding new relevant papers.

The start set was generated by scanning the 20 most relevant[1] results from the database search. This yielded the following 11 papers: Fard and Far [8], Fard and Far [9], Gore and Reynolds [12], Neto [13], Fard [15], Kobayashi et al. [24], Kobayashi et al. [25], Moshirpour et al. [32], Ren et al. [37], Ren et al. [38] and Szabo and Teo [41].

[1] As defined by the Scopus relevancy ranking

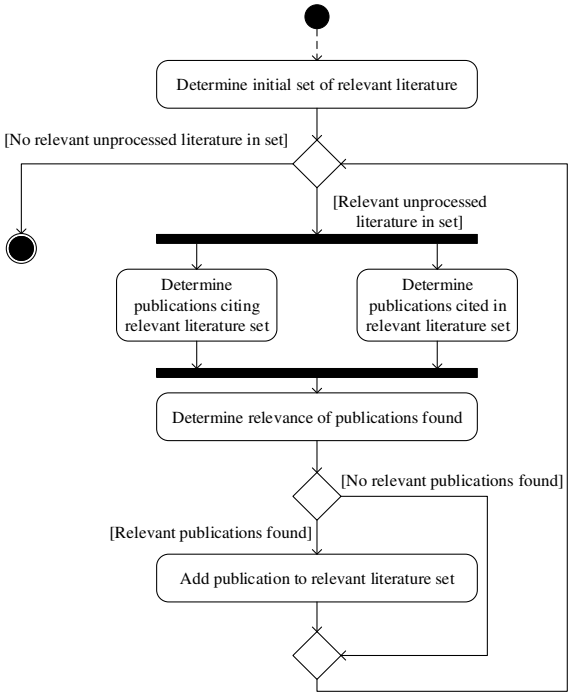

Figure 2: Snowball search visualization

Figure 3 shows the number of papers investigated and included in each round. Note that the number of papers investigate also includes duplicates that have already been evaluated in previous round to reflect the effort needed to identify them as duplicates. The snowball search was conducted based on the citation data from Scopus as of October 2017. The snowball search terminated after round six as no new relevant papers were discovered in this round.

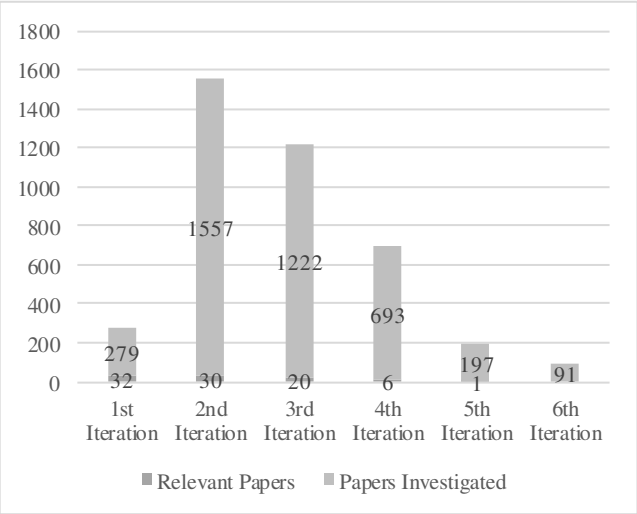

Figure 3: Snowball search iterations

3.6 Inclusion and Exclusion Criteria

For all three searches, we applied the following inclusion and exclusion criteria.

Inclusion:

- Published in a peer-reviewed journal or conference/workshop proceedings
- Focus on detection, verification or validation of behavior that cannot be attributed to a single component, system, etc. but emerges from interplay

Exclusion:

- Focus on detection, verification or validation of behavior that can be attributed to a single component, system, etc.
- Focus on documentation of behavior that cannot be attributed to a single component, system, etc. but emerges from interplay
- Focus on summarizing existing research (e.g., secondary studies)
- Introductory papers for special issues, conferences, or workshops
- Publications shorter than three pages
- Publications not written in English
- Full text not available online

3.7 Data Collection and Quality Assessment

Each search was conducted by two researchers who applied the inclusion and exclusion independent of each other. Where there was disagreement about the inclusion of a publication two more researchers were consulted who decided for or against inclusion. Each literature search was conducted by different researchers. However, inclusion and exclusion criteria were the same for all three searches. To ensure sufficient quality, we limited the included results to peer reviewed publications and excluded very short publications. While grey literature can provide valuable insights [21], their inclusion would have impeded the comparability of the three search methods as they are often not included in databases.

4 RESULTS

Our literature reviews yielded 168 unique results. As can be seen in Figure 4, the snowball search yielded 100 results, the database search 78 and the manual search 10. Only 20 papers, including the 11 papers from the start set, were found by the database search as well as by the snowball search. There is no overlap between the results of the manual search with either the snowball search nor the database search.

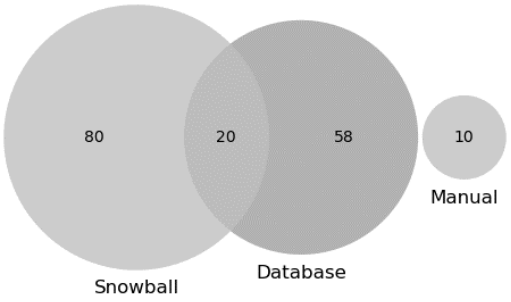

Figure 4: Illustration of the search result sets

4.1 Effectiveness

The manual search found only 10 of the 168 relevant papers, resulting in an effectiveness of 5.95%, which is by far the least effective of the three searches. Conducting the database search, we found 78 of the 168 relevant papers, resulting in an effectiveness of 46.43%. The snowball search found 100 relevant papers, resulting in an effectiveness of 59.52%. Figure 5 illustrates the effectiveness of the different search methods.

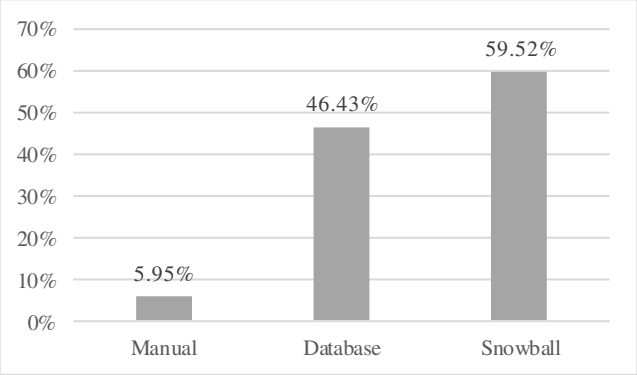

Figure 5: Effectiveness of the different search methods

4.2 Efficiency

The manual search investigated 2665 publications, 10 of which were relevant to the literature review's topic, thus resulting in an efficiency of 0.38%. Again, by far the least efficient of the three searches. The database search found 487 publications to be considered. In the end 78 relevant publications were found, resulting in an efficiency of 16.02%. Finally, the snowball search found 4039 publications for consideration over six iterations, 89 of which were found to be relevant. Adding the eleven papers from the start set to the 89 from the six iterations of snowballing and the 20 papers investigated to form the start set to the 4039 investigated during snowballing, results in 100 papers found by investigating 4059 papers, leading to an efficiency of 2.46%. Figure 6 illustrates the efficiency of the different approaches.

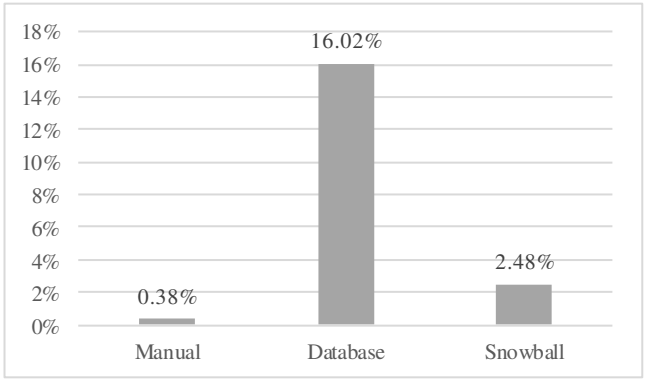

Figure 6: Efficiency of the different search methods

4.3 Publication Volumes

Figure 7 shows a comparison of the annual publication volumes per year for each search method. As can be seen until 2014 database and snowball search indicate similar trends. However, the snowball search indicates a much sharper increase for 2015 than the database search. Note that results for 2017 are based on papers published by October 2017 and thus incomplete.

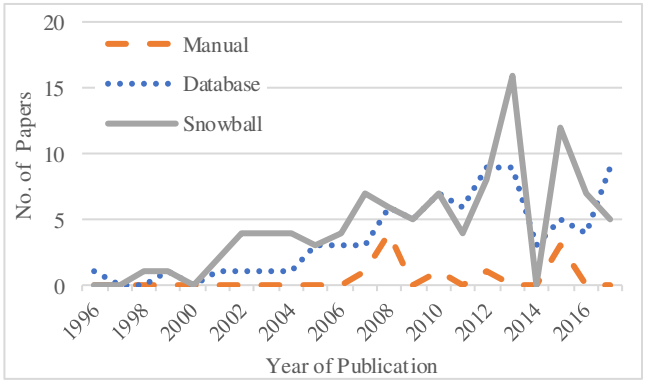

Figure 7: Publication volume per year by search method

4.4 Most Prolific Authors

Table 1 lists the most prolific authors in the field according to the different search methods. Note that the publications identified by manual search did not include more than one by the same author. Therefore, no meaningful author ranking exists for the manual search method. While there are some similarities between the author rankings for the database and snowball search, two notable differences stand out. The second most prolific author according to the database search (Moshirpour – 5 publications) is only ranked eleven according to the ranking for the snowball search (3 publications). Even starker is the contrast for the most prolific author according to the snowball search (Lomuscio – 12 publications). None of his publications were identified by the database search.

Table 1: Most prolific authors by search method

Rank	Manual		Database		Snowball	
	Name	Publications	Name	Publications	Name	Publications
1	No authors with more than one publication		Far, B.H.	8	Lomuscio, A.	12
2			Moshirpour, M.	5	Far, B.H.	10
3			Fard, F.H.	4	Szabo, C.	8
4			Dixon, C.	3	Teo, Y.M.	8
5			Fisher, M. Gore, R. Mousavi, A. Reynolds Jr., P.F. Szabo, C. Teo, Y.M. Winfield, A.F.T.		Fard, F.H.	6

4.5 Top Venues

Table 2 shows the most popular venues where research about the validation and verification of emergent behavior is being published according to the three different search methods.

Table 2: Top venues by search method

Rank	Manual		Database		Snowball	
	Venue	Publications	Venue	Publications	Venue	Publications
1	ICSE	7	PADS	4	AAMAS	6
2	ESEC/FSE	2	INES	3	PADS	
3	TSE	1	TACAS		WSC	5
4			ADMI,	2	RV	3
5			ASE, ICCS, SEKE, TAROS, WSC, Sci Comput Program, Simulation, Stud Comp Intell		ASONAM CCECE ESOA ICFEM IEA/AIE IRI IAS ICONIP, IWSOS SEKE, ICTAI WASA, SoSE, Artif Intell Eng Appl Artif Intel J Log Comp.	2

As our manual search in TOSEM did not result in any relevant publications, we only consider the remaining three. As can be seen, we identified seven relevant publications in the ICSE proceedings of the past ten years, two in the ESEC/FSE proceedings and one TSE paper. No publications from these venues (including all years) were identified by database nor by snowball search.

The publications identified by database search were mainly published in venues where no or only one other relevant paper was published, except for the four papers from the Workshop on Parallel and Distributed Simulation/Workshop on Principles of Advanced and Distributed Simulation/Conference on Principles of Advanced Discrete Simulation (PADS) proceedings and the three papers each from the Conference on Intelligent Engineering Systems (INES) and the Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS) proceedings. Of the INES and TACAS papers only one each was also identified by the snowball search.

The most popular venues for publications about validation and verification of emergent behavior according to the snowball search are International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS), Workshop on Parallel and Distributed Simulation/Workshop on Principles of Advanced and Distributed Simulation/Conference on Principles of Advanced Discrete Simulation (PADS), and Winter Simulation Conference (WSC). However, none of the AAMAS papers and only some of the WSC papers were identified by the database search.

4.6 Most Cited

Table 3 lists the most frequently cited publications found by each search method. As can be seen, each search method discovered a completely different set of most frequently cited papers. Furthermore, it is noteworthy that the citation count of the top five papers identified by database search is considerably lower than those of the manual and the snowball search. The citation data is based on the citations listed in Scopus as of October 2017.

Table 3: Most cited publications per method

Rank	Manual		Database		Snowball	
	Paper	Citations	Paper	Citations	Paper	Citations
1	[29]	209	[34]	61	[40]	252
2	[7]	171	[2]	57	[28]	105
3	[42]	46	[6]	47	[47]	97
4	[11]	13	[44]	36	[4]	89
5	[31]	12	[1]	29	[35]	78

4.7 Most Common Keywords

Table 4 shows the most commonly used author keywords from the publications included in each set. In the set from the manual search only two keywords occurred more than once. The most common keywords from the database search and the snowball search are very similar. Some notable differences are the keywords *feature interaction* and *formal method(s),* which were used by seven and six publications from the database search respectively, but not by any from the snowball search. Note that we merged different spellings of the same keyword.

Table 4: Most common author keywords

Rank	Manual		Database		Snowball	
	Keyword	Count	Keyword	Count	Keyword	Count
1	model synthesis, requirements analysis	2	emergent behavio(u)r(s)	22	multi(-)agent system(s)	22
2			multi(-)agent system(s)	10	emergent behavio(u)r(s)	12
3			feature interaction	7	model(-)checking	7
4			verification	7	formal verification, simulation	6
5			formal method(s), model(-)checking, system(s)(-)of(-)systems	6		

4.8 Research Methods Used

After reading each included paper, we classified each paper according to the research methods classification proposed by Wieringa et al. [43]. Figure 8 shows the results of this classification for each search method. Only the snowball search

discovered *philosophical papers*. Furthermore, the database search discovered considerably more *evaluation research*. Note that due to the very low number of publications discovered by the manual search, it is difficult to draw meaningful conclusions about the distribution of research methods for this set.

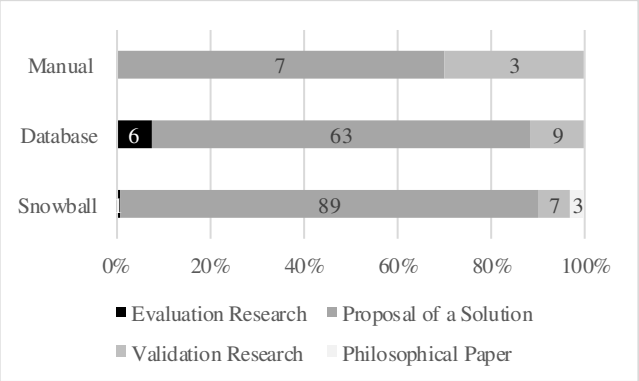

Figure 8: Research methods used according to the different search methods

5 DISCUSSION

In this section, we discuss the findings from Section 4. Section 5.1 summarizes the principle findings with respect to the research questions defined in Section 3.1. Subsequently, Section 5.2 discusses possible interpretations of these findings considering our experiences from conducting the three literature reviews. Section 5.3 gives an overview of the most relevant remaining threats to validity of this study. Considering these threats, Section 5.4 draws inferences from our findings.

5.1 Principal Findings

With respect to our research questions, we can briefly summarize the following findings:

- There are differences in the effectiveness of the different search methods w.r.t. the amount of papers found among all search methods (RQ1). Particularly, snowball search was most effective, database search the second most effective, and manual search by far the most ineffective.

- There are differences in the efficiency of the different search methods (RQ2). Specifically, database search was most efficient, snowball search the second most efficient but already lagging considerably behind database search, and manual search was by far the most inefficient.
- The different search methods led to different conclusions about the studied research field (RQ3). In the mapping study categories investigated (see Section 4.3-4.7), considerable differences between the outcomes of the three different searches were noticed. For instance, when it comes to the author keywords, it becomes noticeable that entire research streams were missed by some search methods.

In particular, we showed that at least for the topic of this systematic literature review (i.e., validation and verification of emergent behavior), which affects multiple sub-disciplines of software engineering research, there is indeed a difference in the outcomes, depending on the search method used. The sets of included papers from the different search methods are nearly disjoint. Additionally, the set from the manual search is disjoint from both other sets. The overlap between the sets found by snowballing and database search consists mostly of the papers used as the start set for the snowball search.

5.2 Interpretation of Findings

Our findings show that different search methods have different effectiveness and efficiency. This is in accordance with findings from other related studies (see Section 2). Additionally, we showed that this finding also holds for studies that affect multiple software engineering related disciplines. Thus, we confirm other researchers' findings claiming that snowball search is a good search method in terms of effectiveness compared to the others. However, an effectiveness of less than 60% can barely considered sufficient nor can an efficiency of less than 3%.

More importantly, we found that in case of literature reviews affecting multiple software engineering related disciplines there is a risk of misrepresenting the field, when choosing only a single search method. In contrast to other studies (see Section 2), the resulting sets of papers included in the different literature reviews are nearly disjoint and each is missing research streams. Particularly noteworthy is our discovery that the snowball search did not result in a set of papers covering all relevant topics and research streams, unlike findings from other related studies. Hence, as different search methods complement each other and lead to partly different results, in our case it seem necessary to use a combination of search methods to reduce the number of papers missed in a systematic literature review and, furthermore, to not arrive at a set of papers misrepresenting the field.

We found manual search to be ineffective and inefficient. However, the manual search resulted in papers not found by the other search methods. Hence, manual search can contribute to literature searches as such papers found seem to be not easily detected by snowballing and database search, most probably due to the use of more general terminology used to attract interest from researchers from multiple disciplines.

5.3 Threats to Validity

First of all, it must be stated that there is a risk that the results presented in this paper might not be generalizable to the overall field of software engineering as we report on experiences made from one case. Particularly, it must be considered that other

related studies came in part to different conclusions (see Section 2). However, the findings at least show, that, the use of snowball search is not sufficient for all literature reviews and that there are topics that need to consider multiple search methods as the use of a single search method can lead to a misrepresentation of the field. Future work, will, hence, have to deal with the question whether findings are generalizable for all literature reviews that affect multiple software engineering related disciplines and if not, under which circumstances such a risk exists.

Second, it must be mentioned that we compared one execution of each search method to get an impression of the differences. Existing approaches for search string optimization [48] have not been applied. Application of such approaches might lead to more overlapping sets of papers found.

Furthermore, the definition of the search string and the selected database, the definition of the start set for the snowball search, and the selection of venues and time span searched in the manual search, might have had an impact and other criteria might have led to different results. However, we based our search string, database selection, the start set, and the selection criteria for the manual search on common suggestions (e.g., [33, 45]) and are, thus, confident that the results reflect common literature review practices.

Nevertheless, regarding our reporting of effectiveness, it must be considered that we do not know all relevant papers and, hence, base our measurement on the papers found across all three literature reviews. However, this problem exists for any kind of literature study.

Lastly, a major threat remains as papers might have been misclassified (i.e. found to be relevant while irrelevant and vice versa). Therefore, we conducted an investigation of the final sets by all researchers. In this we found that papers in the manual search had been included, although not exactly matching the inclusion criteria. Those papers were then removed. This might result from the ineffectiveness and inefficiency of the manual review, leading researchers to be more inclined to include papers as otherwise very few paper were found. We assume that this threat must be considered for all literature reviews using manual search.

5.4 Inferences

The results from our study show that the use of a single search method can lead to a misrepresentation of the research field under investigation. Researchers should not just rely on the single most effective and/or efficient search method, in particular when conducting literature reviews that affect different software engineering related disciplines.

Our findings complement existing research, partly supporting findings from other researchers, partly giving new insights into the impact of the used search method when it comes to literature searches affecting multiple software engineering related disciplines. Like similar studies [3, 17], we found the used search strategy to have a major impact on the outcome of systematic literature studies. However, interestingly, in contrast to other research, we found the three different search methods (i.e. snowballing, manual search, and database search) result in nearly disjoint sets of included papers, leading to different representations of the field to be investigated. As other researchers also found different search methods to come to different results for rather broad literature searches, we are confident in the validity of our results, even when taking the threats to validity into account.

We want to stress again the difference to previous studies (see Section 2): In this case, we found no search method advantageous in the sense that one search method leads to finding all or most publications found by other search methods. Even the often highly valued, and particularly good results producing snowballing missed all papers found by the manual search and almost all papers found by database search.

Commonly researchers conclude that snowballing is better than database search for broad literature reviews, which investigate a wide field or which can only rely on general terms. As cause, it is often assumed, that changing terminology or different terminology in different fields hinders detection of related papers. In our case, snowballing was not able to identify related papers from different author groups and different disciplines due to the fact that authors either do not know what happens in other software engineering disciplines or prefer citing within their own community. Our database search could find papers and entire areas of interest snowballing did not detect.

While these findings need substantiation by further complementary investigations on literature searches for topics relevant in multiple software engineering disciplines, for now, we conclude that there seems to be a need to use different search methods when reviewing a topic potentially concerning different communities. However, we cannot rule out the possibility that there remains a potentially large amount of papers undiscovered, even when using all three search methods. While we cannot conclude that a combination of all three search methods is generally best, we can state that in our case it would lead to better results. Future work will be needed on defining effective and efficient search strategies for such literature studies affecting multiple disciplines. In particular, the time consumption of combining different search methods for conducting one single systematic literature review or map may easily become disproportionate. Hence, there is a need to define criteria for identifying cases in which a more intensive search approach is needed and cases where, e.g., a single database search is sufficient.

6 CONCLUSION

There is a need to get a deeper understanding of the effects search strategies have on the outcome of systematic literature reviews and maps and which search method is preferable in which case (cf. [3]). This paper contributes a comparison of different search methods for a concrete research topic, thereby complementing existing research on the advantages and disadvantages of different search methods. As the topic of the literature search (i.e. validation and verification of emergent behavior) is relevant for multiple software engineering related

disciplines, this paper contributes to the question, which search methods is preferable for broad literature studies, where multiple application areas need to be considered relevant.

In this paper, we report our findings that show that in these situations the commonly suggested search methods (i.e. snowballing, database search, manual search by venues) result in almost disjoint sets of found relevant papers. This means, in contrast to other studies, we showed that each single search method bears the risk of misrepresenting the field, even snowballing which has commonly been suggested for broader literature searches. Consequently, for literature studies where the topic affects multiple disciplines, we assume, a more careful definition of the search strategy is needed. We assume, that an appropriate search strategy for these cases must use a combination of the different search methods. However, there is still a need for future work to investigate the effects of search methods and strategies in more detail and to identify optimized search strategies for literature studies affecting multiple software engineering related disciplines.

ACKNOWLEDGMENTS

We thank Patricia Aluko Obe, Julian Bellendorf, Kevin Keller, Nils Schlüter, Florian Uphoff for their support in conducting the literature searches. This work is funded in part by the *German Ministry for Education and Research* under grant number 01IS16043V.

REFERENCES

[1] Amyot, D., Logrippo, L., Buhr, R.J.A. and Gray, T. 1999. Use case maps for the capture and validation of distributed systems requirements. *4th IEEE International Symposium on Requirements Engineering, RE'99* (Limerick, Ireland, 1999), 44–53.

[2] Apel, S., Speidel, H., Wendler, P., Von Rhein, A. and Beyer, D. 2011. Detection of feature interactions using feature-aware verification. *26th IEEE/ACM International Conference on Automated Software Engineering, ASE 2011* (Lawrence, KS, 2011), 372–375.

[3] Badampudi, D., Wohlin, C. and Petersen, K. 2015. Experiences from Using Snowballing and Database Searches in Systematic Literature Studies. *Proceedings of the 19th International Conference on Evaluation and Assessment in Software Engineering* (New York, NY, USA, 2015), 17:1–17:10.

[4] Bar-Yam, Y. 2004. A mathematical theory of strong emergence using multiscale variety. *Complexity.* 9, 6 (2004), 15–24. DOI:https://doi.org/10.1002/cplx.20029.

[5] Broy, M. and Schmidt, A. 2014. Challenges in Engineering Cyber-Physical Systems. *Computer.* 47, 2 (Feb. 2014), 70–72. DOI:https://doi.org/10.1109/MC.2014.30.

[6] Chen, T., Forejt, V., Kwiatkowska, M., Parker, D. and Simaitis, A. 2013. PRISM-games: A Model Checker for Stochastic Multi-Player Games. *Tools and Algorithms for the Construction and Analysis of Systems* (Mar. 2013), 185–191.

[7] Classen, A., Heymans, P., Schobbens, P.-Y., Legay, A. and Raskin, J.-F. 2010. Model Checking Lots of Systems: Efficient Verification of Temporal Properties in Software Product Lines. *Proceedings of the 32Nd ACM/IEEE International Conference on Software Engineering - Volume 1* (New York, NY, USA, 2010), 335–344.

[8] Fard, F.H. and Far, B.H. 2013. Detection and verification of a new type of emergent behavior in multiagent systems. *INES 2013 - IEEE 17th International Conference on Intelligent Engineering Systems, Proceedings* (2013), 125–130.

[9] Fard, F.H. and Far, B.H. 2013. Visualizing the Network of Software Agents for Verification of Multiagent Systems. *Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining* (New York, NY, USA, 2013), 1280–1281.

[10] Felizardo, K.R., Mendes, E., Kalinowski, M., Souza, É.F. and Vijaykumar, N.L. 2016. Using Forward Snowballing to Update Systematic Reviews in Software Engineering. *Proceedings of the 10th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement* (New York, NY, USA, 2016), 53:1–53:6.

[11] Filieri, A., Hoffmann, H. and Maggio, M. 2015. Automated multi-objective control for self-adaptive software design. *10th Joint Meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering, ESEC/FSE 2015* (2015), 13–24.

[12] Gore, R. and Reynolds, P.F. 2008. Applying causal inference to understand emergent behavior. *Proceedings - Winter Simulation Conference* (2008), 712–721.

[13] Graciano Neto, V.V. 2016. Validating emergent behaviors in systems-of-systems through model transformations. *2016 ACM Student Research Competition at MODELS 2016, ACM SRC at MODELS 2016* (2016).

[14] Greenhalgh, T. and Peacock, R. 2005. Effectiveness and efficiency of search methods in systematic reviews of complex evidence: audit of primary sources. *BMJ.* 331, 7524 (Nov. 2005), 1064–1065. DOI:https://doi.org/10.1136/bmj.38636.593461.68.

[15] Hendijani Fard, F. 2013. Detecting and fixing emergent behaviors in Distributed Software Systems using a message content independent method. *2013 28th IEEE/ACM International Conference on Automated Software Engineering, ASE 2013 - Proceedings* (2013), 746–749.

[16] Horsley, T., Dingwall, O., Tetzlaff, J.M. and Sampson, M. 2009. Checking reference lists to find additional studies for systematic reviews. *Cochrane Database of Systematic Reviews.* The Cochrane Collaboration, ed. John Wiley & Sons, Ltd.

[17] Jalali, S. and Wohlin, C. 2012. Systematic Literature Studies: Database Searches vs. Backward Snowballing. *Proceedings of the ACM-IEEE International Symposium on Empirical Software Engineering and Measurement* (New York, NY, USA, 2012), 29–38.

[18] Kitchenham, B. and Brereton, P. 2013. A systematic review of systematic review process research in software engineering. *Information and Software Technology.* 55, 12 (Dec. 2013), 2049–2075. DOI:https://doi.org/10.1016/j.infsof.2013.07.010.

[19] Kitchenham, B., Brereton, P., Li, Z., Budgen, D. and Burn, A. 2011. Repeatability of systematic literature reviews. *15th Annual Conference on Evaluation Assessment in Software Engineering (EASE 2011)* (Apr. 2011), 46–55.

[20] Kitchenham, B., Brereton, P., Turner, M., Niazi, M., Linkman, S., Pretorius, R. and Budgen, D. 2009. The impact of limited search procedures for systematic literature reviews; A participant-observer case study. *2009 3rd International Symposium on Empirical Software Engineering and Measurement* (Oct. 2009), 336–345.

[21] Kitchenham, B.A., Brereton, P., Turner, M., Niazi, M.K., Linkman, S., Pretorius, R. and Budgen, D. 2010. Refining the systematic literature review process—two participant-observer case studies. *Empirical Software Engineering.* 15, 6 (Dec. 2010), 618–653. DOI:https://doi.org/10.1007/s10664-010-9134-8.

[22] Kitchenham, B.A., Budgen, D. and Brereton, P. 2016. *Evidence-based software engineering and systematic reviews.* Chapman & Hall.

[23] Kitchenham, B.A. and Charters, S. 2007. *Guidelines for performing systematic literature reviews in software engineering.* School of Computer Science and Mathematics, Keele University.

[24] Kobayashi, K., Kurano, T., Kuremoto, T. and Obayashi, M. 2012. Cooperative Behavior Acquisition in Multi-agent Reinforcement Learning System Using Attention Degree. *Neural Information Processing* (Nov. 2012), 537–544.

[25] Kobayashi, K., Ueda, R. and Arai, T. 2008. Cooperative behavior of multiple robots by chain of monolithic policies for two robots. *10th International Conference on Intelligent Autonomous Systems, IAS 2008* (2008), 202–210.

[26] Kouvaros, P. and Lomuscio, A. 2015. Verifying emergent properties of swarms. *24th International Joint Conference on Artificial Intelligence, IJCAI 2015* (2015), 1083–1089.

[27] Kuhrmann, M., Fernández, D.M. and Daneva, M. 2017. On the Pragmatic Design of Literature Studies in Software Engineering: An Experience-based Guideline. *Empirical Softw. Engg.* 22, 6 (Dec. 2017), 2852–2891. DOI:https://doi.org/10.1007/s10664-016-9492-y.

[28] Lomuscio, A., Qu, H. and Raimondi, F. 2009. MCMAS: A Model Checker for the Verification of Multi-Agent Systems. *Computer Aided Verification* (Jun. 2009), 682–688.

[29] Lorenzoli, D., Mariani, L. and Pezzè, M. 2008. Automatic Generation of Software Behavioral Models. *Proceedings of the 30th International Conference on Software Engineering* (New York, NY, USA, 2008), 501–510.

[30] MacDonell, S., Shepperd, M., Kitchenham, B. and Mendes, E. 2010. How Reliable Are Systematic Reviews in Empirical Software Engineering? *IEEE Transactions on Software Engineering.* 36, 5 (Sep. 2010), 676–687. DOI:https://doi.org/10.1109/TSE.2010.28.

[31] Mitchell, B. 2008. Characterizing communication channel deadlocks in sequence diagrams. *IEEE Transactions on Software Engineering*. 34, 3 (2008), 305–320. DOI:https://doi.org/10.1109/TSE.2008.28.

[32] Moshirpour, M., Mousavi, A. and Far, B.H. 2010. A technique and a tool to detect emergent behavior of distributed systems using scenario-based specifications. *Proceedings - International Conference on Tools with Artificial Intelligence, ICTAI* (2010), 153–159.

[33] Petersen, K., Vakkalanka, S. and Kuzniarz, L. 2015. Guidelines for conducting systematic mapping studies in software engineering: An update. *Information and Software Technology*. 64, (Aug. 2015), 1–18. DOI:https://doi.org/10.1016/j.infsof.2015.03.007.

[34] Plath, M. and Ryan, M. 2001. Feature integration using a feature construct. *Science of Computer Programming*. 41, 1 (Sep. 2001), 53–84. DOI:https://doi.org/10.1016/S0167-6423(00)00018-6.

[35] Raimondi, F. and Lomuscio, A. 2007. Automatic verification of multi-agent systems by model checking via ordered binary decision diagrams. *Journal of Applied Logic*. 5, 2 (2007), 235–251. DOI:https://doi.org/10.1016/j.jal.2005.12.010.

[36] Raimondi, F. and Lomuscio, A. 2004. Verification of multiagent systems via ordered binary decision diagrams: An algorithm and its implementation. *3rd International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS 2004* (New York, NY, 2004), 630–637.

[37] Ren, G., Deng, P. and Yang, C. 2017. A 3-layer method for analysis of cooperative behaviors of physical devices in cyber-physical systems. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (2017), 741–754.

[38] Ren, G., Hua, Q., Deng, P., Yang, C. and Zhang, J. 2017. A Multi-Perspective Method for Analysis of Cooperative Behaviors among Industrial Devices of Smart Factory. *IEEE Access*. 5, (2017), 10882–10891. DOI:https://doi.org/10.1109/ACCESS.2017.2708127.

[39] Skoglund, M. and Runeson, P. 2009. Reference-based Search Strategies in Systematic Reviews. *Proceedings of the 13th International Conference on Evaluation and Assessment in Software Engineering* (Swindon, UK, 2009), 31–40.

[40] Stone, P. and Veloso, M. 1999. Task decomposition, dynamic role assignment, and low-bandwidth communication for real-time strategic teamwork. *Artificial Intelligence*. 110, 2 (1999), 241–273. DOI:https://doi.org/10.1016/S0004-3702(99)00025-9.

[41] Szabo, C. and Teo, Y.M. 2013. Post-mortem Analysis of Emergent Behavior in Complex Simulation Models. *Proceedings of the 1st ACM SIGSIM Conference on Principles of Advanced Discrete Simulation* (New York, NY, USA, 2013), 241–252.

[42] Uchitel, S., Brunet, G. and Chechik, M. 2007. Behaviour model synthesis from properties and scenarios. *29th International Conference on Software Engineering, ICSE 2007* (Minneapolis, MN, 2007), 34–43.

[43] Wieringa, R., Maiden, N., Mead, N. and Rolland, C. 2005. Requirements engineering paper classification and evaluation criteria: A proposal and a discussion. *Requirements Engineering*. 11, 1 (Nov. 2005), 102–107. DOI:https://doi.org/10.1007/s00766-005-0021-6.

[44] Winfield, A.F.T., Sa, J., Gago, M.-C.F., Dixon, C. and Fisher, M. 2005. On formal specification of emergent behaviours in swarm robotic systems. *International Journal of Advanced Robotic Systems*. 2, 4 (2005), 363–370.

[45] Wohlin, C. 2014. Guidelines for Snowballing in Systematic Literature Studies and a Replication in Software Engineering. *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering* (New York, NY, USA, 2014), 38:1–38:10.

[46] Wohlin, C., Runeson, P., da Mota Silveira Neto, P.A., Engström, E., do Carmo Machado, I. and de Almeida, E.S. 2013. On the reliability of mapping studies in software engineering. *Journal of Systems and Software*. 86, 10 (Oct. 2013), 2594–2610. DOI:https://doi.org/10.1016/j.jss.2013.04.076.

[47] Wooldridge, M., Fisher, M., Huget, M.-P. and Parsons, S. 2002. Model checking multi-agent systems with MABLE. *Proceedings of the International Conference on Autonomous Agents* (2002), 952–959.

[48] Zhang, H., Babar, M.A. and Tell, P. 2011. Identifying relevant studies in software engineering. *Information and Software Technology*. 53, 6 (Jun. 2011), 625–637. DOI:https://doi.org/10.1016/j.infsof.2010.12.010.