International Journal of Software Engineering
and Knowledge Engineering
Vol. 32, No. 2 (2022) 163–191

DOI: 10.1142/S0218194022500140

World Scientific
www.worldscientific.com

A Systematic Mapping Study of Search-Based Software Engineering for Enterprise Application Integration

Angela Mazzonetto*,‡, Rafael Z. Frantz*,§, Fabricia Roos-Frantz*,¶, Carlos Molina-Jimenez†,‖ and Sandro Sawicki*,**

*Department of Exact Sciences and Engineering
Unijuí University, Ijuí - RS - Brazil

†Department of Computer Science and Technology
University of Cambridge, UK
‡angela.mazzonetto@unijui.edu.br
§rzfrantz@unijui.edu.br
¶frfrantz@unijui.edu.br
‖carlos.molina@cl.cam.ac.uk
**sawicki@unijui.edu.br

Received 24 August 2021
Revised 23 September 2021
Accepted 29 January 2022

Search-Based Software Engineering (SBSE) is widely used in different fields of Software Engineering, notoriously, in Enterprise Application Integrations (EAIs). EAI encompasses methodologies, techniques and tools that a software engineer can use to create integration solutions. SBSE is currently an active research topic of increasing interest. The number and diversity of publications produced yearly are large to the extent that it is hard to identify the active research groups, their locations, techniques used and research topics that have not received enough attention. To answer these questions categorically, we have conducted systematic mapping study of the literature. In this paper, we report our methodology and findings. In our study, we used systematic search strategies that resulted in the retrieval of 560 articles, of which we first selected 25. Second, on the basis of the authors' experience, we included eight additional articles. Finally, we used a snowballing sample technique to include another 12 articles. The results demonstrate that during the last two decades (1999–2020) EAI has benefited from the use of Search-Based Software Engineering techniques.

Keywords: Search-based software engineering; optimization; enterprise application integration; runtime system; systematic mapping study.

1. Introduction

Enterprise Application Integration (EAI) is still a research topic and as such has received significant attention from the research community over the last 21 years.

§Corresponding author.

The main motivation is the growing need for data exchange and reuse of functionality between applications in the context of enterprise software. These applications support business processes and are commonly not designed and developed to collaborate. Enterprise software ecosystems [1] are often heterogeneous, including applications developed on different technologies, with different data formats and models, that run on different operating systems. These applications run on-premises or in the cloud, frequently consuming third-party software services provided over the Internet. EAI aims to provide methodologies, techniques and tools to create integration solutions [2]. An integration solution deals with a set of applications within the software ecosystem to facilitate data sharing and functionality reuse, to support its business processes [3, 4]. In recent years, EAI has been a subject of interest due to the pressing need to design and implement solutions that integrate web applications, sensor data, and social media data with traditional applications that support companies' business processes [5, 6].

The need to find optimal or near-optimal solutions appears frequently is several areas of engineering, such as civil, electrical, mechanical, chemical, medical, biomedical engineering and software engineering [7]. Engineers address the challenge with the help of optimization techniques. Incidentally, in software engineering, engineers often deal with problems associated with balancing competitive constraints, such as trade-offs and imprecision in the requirements. These issues are in essence optimization problems that demand optimal or near-optimal solutions that fit within pre-established limits. In these situations, the engineer can abstract the problem as a search problem and take advantage of well-known optimization techniques. A search problem is a type of computational problem, where optimal or near-optimal solutions are looked for in a space of candidate solutions, managed by a fitness function that identifies better and worse solutions [8].

The term Search-Based Software Engineering (SBSE) was coined by Harman and Jones [9], and since then, it has been used to refer to the use of optimization techniques in software engineering. The first records of search-based techniques in software engineering date back to the mid-1990s and were used in software testing. Since then, SBSE has experienced a continuous increase in research activity, as new results and areas of application continue to grow rapidly. Additionally, SBSE has been used to solve problems in different phases of the software development life cycle, from requirements engineering, design, code development, testing, project planning, maintenance and re-engineering [8]. The interest in SBSE in EAI has resulted in a large list of publications that is difficult to quantify and categorized unless a systematic approach is taken. The aim and main contribution of this paper are to explore the situation in a manner that the SBSE in EAI community is able to appreciate which topics have been covered and which ones have been neglected; and what topics are emerging and trendy.

There are two types of methodologies to conduct a literature review: vertical and horizontal [10]. Vertical reviews aim to conduct an in-depth research on a specific topic, while horizontal reviews contemplate wider and more comprehensive research. An example of a horizontal review is the systematic mapping study [11]. Based on

clear objectives defined by the research questions, this type of study provides an overview of a research topic through the classification and counting of contributions regarding the categories of that classification.

In our mapping study, based on the counting and classification of publications, concerning the pre-established categories, we identified studies that use optimization techniques to solve problems in EAI, mainly addressing problems of heterogeneous systems integration, followed by workflow, service composition and quality of web services. For instance, we have found that amongst the most used techniques are algorithms based on more than one heuristic technique, and the use of genetic algorithms in 16% of the studies. Another finding was that most of the studies were published at conferences between 2007 and 2013. The university that published the largest number of studies was Dresden University of Applied Sciences in Germany.

2. Related Work

In the scientific literature, there are systematic mappings aimed at structuring and understanding fields of study and topics in the area of software engineering, such as requirements, design, metrics, testing, standards, patterns, design, optimization and EAI. Amongst these studies, some discuss the use and application of SBSE to solve problems in software engineering.

Several studies of systematic mapping in software engineering have been published, however, as revealed by our literature review, mapping of SBSE in EAI has been largely neglected. In this section, we focus our attention to publications on systematic mapping studies on EAI and systematic mapping studies on SBSE that we consider related to our work.

Table 1 summarizes the systematic mapping studies found, which focus on requirements, software design, software metrics, software testing, patterns, project management, software architecture, software refactoring, optimization, and EAI.

A systematic mapping to verify the evolution of research in SBSE in Brazil was conducted by Assunção *et al.* [14]. The authors surveyed the main research groups working on SBSE to reveal: what software engineering problems they mostly address and the research techniques that they use more frequently to solve their problems. The authors also analyzed the main publications and collaborations on their research topics. Their study revealed that Brazilian researchers published research-based articles in several areas of Software Engineering from 2000 to 2012. In that work, the authors identified 98 articles published between 2000 and 2012 by 90 different authors which resulted in more than 700 citations on Google Scholar. The number of publications and collaboration amongst authors and researchers involved in the area of SBSE has increased since 2010, to the extent that more than 70% of the studies were published from that year on. The authors also discussed the community, introducing Brazilian researchers and research groups focused on SBSE, providing an overview of their research projects. They paid attention to the venues where SBSE studies are published. Their analysis showed that about 75% of publications

Table 1. Topics covered by systematic mappings in software engineering.

	Requirement	Software design	Software metrics	Software testing	Patterns	Project management	Optimization	Software refactoring	Software architecture	EAI
Aleti *et al.* [12]									✓	
Pitangueira *et al.* [13]	✓									
Assunção *et al.* [14]							✓			
Meignan *et al.* [15]							✓			
Pitangueira *et al.* [16]	✓									
Ambreen *et al.* [17]	✓									
Marinani *et al.* [18]								✓		
Silva *et al.* [19]				✓						
Vegendla *et al.* [20]	✓									
Ameller *et al.* [21]	✓									
Ros and Runeson [22]				✓						
Gurbuz and Tekinerdogan [23]				✓						
Barros-Justo and Cravero-Leal [24]					✓					
Villalobos-Arias *et al.* [25]				✓						
Hirama and Carvalho [26]						✓				
Rasool and Noman [27]			✓							
Souza *et al.* [28]							✓			
Ramírez *et al.* [29]						✓				
Real *et al.* [30]										✓
Morales-Trujillo *et al.* [31]		✓								
Ramírez *et al.* [32]							✓			

produced in Brazil were published in conference proceedings, 20.04% in journals, 2.04% were published as posters, 1.02% in book chapters and 1.94% as technical reports. The authors state that the following evolutionary techniques are preferred: Multi-objective Evolutionary Algorithms, Genetic Algorithms, and Genetic Programming. The Non-dominated Sorting Genetic Algorithm II and Multi-objective Evolutionary Algorithm-Based techniques are the most used, but Improving the Strength Pareto technique and the Multi-Objective Cellular genetic algorithm are also frequently used. In addition to evolutionary techniques, 22% of studies used other metaheuristics, including Simulated Annealing, GRASP, and Hill Climbing. Four articles used Ant Colony optimization. Classical techniques are applied by 5% of the studies: in this category, Greedy and Branch-and-Bound are the most used. The authors also claim that Brazilian researchers published articles related to the application of SBSE in several areas of software engineering between 2000 and 2012, including software testing (54 articles), project management (14 articles), requirements (14 articles) and software design (8 articles). According to the conclusions, the SBSE community in Brazil has been expanding, both geographically and in terms of number of publications. Furthermore, the creation of a national *workshop* focused on SBSE research was essential to this growth.

The study by Harman *et al.* [8] reports that articles on SBSE appeared in 201 different publishing venues. The authors also point out that the dissemination of publications indicates that SBSE has been gaining acceptance in leading software engineering journals and conferences, as well as in the Search-Based optimization communities. Harman *et al.* [8] cite as an example the venues where studies on regression testing and software testing tend to appear. Work on Regression testing can appear at conferences such as the ACM International Symposium on Software Testing and Analysis and the IEEE International Conference on Software Testing. Harman *et al.* [8] comment on numerous authors who have carried out studies in the area of SBSE. The authors also argue that a wide range of different search and optimization techniques have been used in software engineering and that the most common are Tabu Search, Simulated Annealing, Genetic Algorithms, Genetic Programming, and Hill Climbing.

Aleti *et al.* [12] presented a systematic literature review that explores optimization approaches applied in software architecture. The authors reviewed 188 articles that provided an overview of the state-of-the-art in the domain of software architecture optimization. This work also presented a taxonomy to classify research involving software architecture optimization. Pitangueira *et al.* [13] used a systematic literature review to find which SBSE approaches explored software requirements selection and prioritization problems. The authors identified traditional techniques in the area, such as mono-objective modeling, as well as gaps in the execution of experiments. In 2015, Meignan *et al.* [15] presented a literature review on interactive optimization methods. According to the authors, the current literature does not yet include an optimization survey that covers all interactive approaches to solve optimization problems. In this sense, they proposed a classification of interactive

optimization methods from two perspectives: user and system. Silva *et al.* [19] presented a systematic literature review on the use of search metaheuristics for mutation testing. The aim of the authors is to find evidence of how the Search-Based Software Testing (SBST) area has explored the field of mutation testing. This work found 263 articles published between 1996 and 2014 that addressed the use of search techniques based on metaheuristics that involved the optimization of mutation testing. The authors reported that the use of research-based techniques can improve the field of mutation testing. In 2015, Pitangueira *et al.* [16] presented a new systematic literature review and mapping aiming to identify the state-of-the-art of optimization applications in software requirements selection problems. The mapping helped to identify publishing forums and authors in the area. This work also analyzed some quality criteria that must be considered in the selection and prioritization of requirements. In their study, Souza *et al.* [28] address aspects of human competitiveness in SBSE. However, neither of these studies addresses the use and application of optimization techniques in EAI. This is regrettable since it is well known that solutions in EAI can benefit from SBSE techniques.

Mariane and Vergilio [18] presented a systematic literature review that investigated research-based approaches to software refactoring over a sixteen-year period. According to the authors, the most used search technique was based on evolutionary algorithms, where Fowler's catalog refactoring were the most applied. Souza *et al.* [28] present a systematic mapping study using snowballing to provide an overview of the current state of human competitiveness in SBSE. The study confirms the growing interest in discussing human competitiveness in SBSE, especially after 2010. The authors find out that the Symposium on Search-Based Software Engineering was the conference with most articles. There are some similarities between the number of studies published in conferences (48.3%) and journals (41.4%). Furthermore, the authors affirm that Requirements/Specification and Software/Product Verification were identified as the most common software engineering areas addressed by primary sources. They also state the need to increase studies on human competitiveness of SBSE. In 2019, Ramíres *et al.* [29] presented results from a systematic literature review of interactive SBSE (iSBSE). This interaction occurs when the software engineer provides his feedback during the research process. The authors highlighted that, for some problems, human intervention is still important to indicate information to the algorithm about how a situation should be resolved. In this work, they formulated research questions about the use of interactivity within the SBSE. They also investigated the most used search techniques and how interactive approaches can help in the software development life cycle. From the 669 articles found, 26 primary studies were selected. The analysis was formulated from a classification scheme based on the problem formulation, interactive approach, research technique and empirical structure. These results allowed to discuss many open issues and future trends involving the SBSE community.

In the systematic mapping performed by Real *et al.* [30], EAI initiatives that address semantic aspects and cover the service integration layer are investigated.

The main focus of the study was to investigate approaches to integrate services and the use of ontology's in the EAI context. Considering the results of the study, the authors identified some gaps related to the research topic, such as the lack of systematic approaches to guide integration in the service layer; for instance, task ontology's are not used to support the integration of services; another problem is the lack of a general conceptualization of enterprise application services. Real *et al.* [30] claim that only a few EAI initiatives followed a systematic approach, in spite of the fact that it is necessary to provide viable methods to perform semantic EAI. Regarding the second gap mentioned, the authors noted that the behavioral aspects of services have been neglected.

Most studies focused on structural aspects of the service (input and output parameters) to designate semantics, though the authors affirm that behavioral aspects must also be considered. They also state that tasks based on ontology's can be useful for assigning semantics to services, features, activities and other related information. According to the authors, none of the analyzed studies had a conceptualization of the enterprise application service. To conclude, Real *et al.* [30] analyzed how integration initiatives have addressed semantic integration in the service layer and how semantics has been attributed to services.

Ramírez *et al.* [32] presented a systematic literature review aiming to provide an overview of the SBSE field in Spain (2001 and 2019). This study was conducted and adapted for this community considering the guidelines developed for systematic reviews in the area of Software Engineering [10]. In this work, research questions were formulated in order to show the contribution of the Spanish SBSE community in quantitative data. The results revealed an active SBSE community in Spain, with 145 authors and 19 institutions. From more than 3700 papers retrieved, 232 primary studies published in conferences and international journals were selected, including articles published in collaborations with researchers from other international universities. The most used search techniques were population-based metaheuristics, in addition to the use of alternative proposals, such as single-solution metaheuristics and exact techniques. Global search methods were the most used (83.2%), while 6.9% of articles used local search methods and 15.5% did not clearly describe the search algorithm. According to the authors, challenges remain for the Spanish SBSE community to improve the empirical validation of their experiments, as well as to seek new cooperation with industrial partners.

Cerqueira *et al.* [33] carried out a systematic mapping study on the use of ontology in EAI to capture semantics of the integration in the process layer. The results of the mapping provided an overview of research related to the integration of processes in semantic EAI initiatives. The authors claim that the semantic EAI initiatives have been used in ontology's to assign semantics mainly to data and services. Service-oriented solutions have been applied to provide communication between applications associated to process managers (such as workflow engine) that orchestrate services to support the execution of processes. The authors further claim that models have been used to support integration at the conceptual level and also to create integration

solutions based on model transformation. Cerqueira *et al.* [33] also mention some research gaps, namely: the lack of systematic approaches to guide integration in the process layer; the absence of task ontology's to support process integration; and the lack of a general concept of business processes.

It is worth mentioning that both Assunção *et al.* [14] and Harman *et al.* [8] present results on the application of optimization techniques to solve problems in several areas of software engineering such as: network protocols, specification of requirements, tools and techniques design, tools and techniques for coding, verification, testing, maintenance, artificial intelligence. Harman *et al.* [8] also state that 54% of the publications are on the application of optimization in software testing. Therefore, to the best of our knowledge, no systematic mapping studies of EAI optimization techniques have been published. This is regrettable since it is well known that solutions in EAI can benefit from SBSE techniques. Our study is aimed at covering the gap. It focuses specifically on the use of techniques of Search-Based Software Engineering and therefore, complements the articles mentioned above.

3. Research Method

This section details the research method that we have used in our study; it is based on the protocol proposed by Petersen *et al.* [11] and Wohlin *et al.* [34]. It consists of five activities: definition of research questions, search, analysis and classification, results and discussion, and threats to validity. Figure 1 shows a diagram with an overview of the activities performed. The first activity is focused on defining the research questions that must be answered by the systematic mapping. In the second activity, the search and selection of articles are performed. This activity is divided

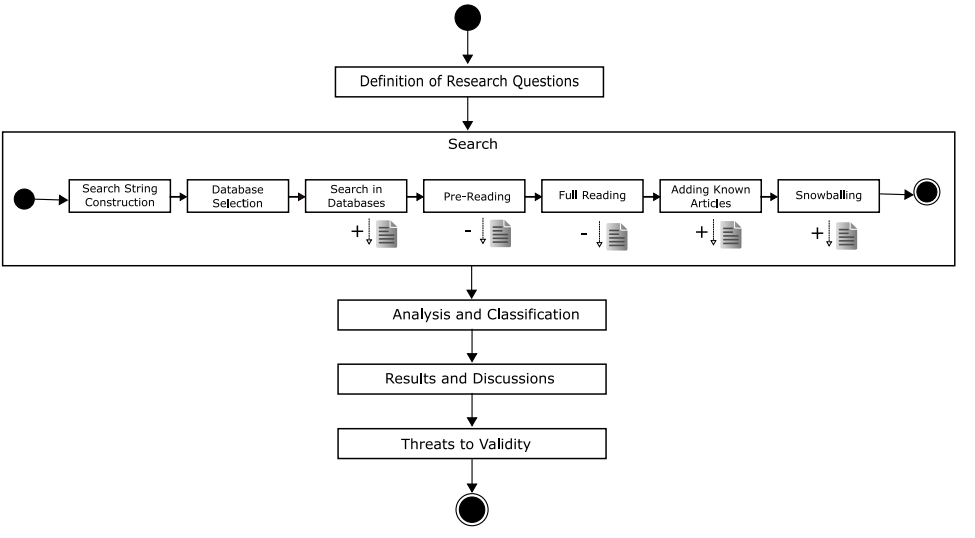

Fig. 1. Activities of the systematic mapping process.

into seven steps: building the search string, selecting the research databases, searching the databases, screening, and pre-reading of articles, full-text reading, adding known articles, and snowballing [35]. The third activity classifies the results according to the research questions, defined and discussed in the first activity of the study. The results extracted from the classification are analyzed in the fourth activity. Finally, in the last activity, the threats to validity that may affect the result of the study are assessed.

3.1. *Definition of research questions*

In this activity, the research questions are elaborated according to the general purpose of the systematic mapping study. The identification of keywords is a fundamental activity in the search for articles. As a methodology to guide the activity of defining research questions, we used a simplified version of (Population, Intervention, Comparison, Outcome (PICO)) introduced by Kitchenham and Charters [10]. This process consists of identifying keywords, formulating sequences of words in a search string from the search questions.

Population: Frames the people or companies that the study is expected to cover. For example, Solidity programmers, software testers with more than five years of experience and companies with expertise in software development for automotive applications.

Intervention: Defines the software methodology, procedure or technology under study. For example, software cost estimates, testing of automotive IoT devices and development of secure software for mobile phone payment applications.

Comparison: Concerns any comparison with data listed in the intervention (comparison with some methodology, tool, technology or software process), for example, comparison between development frameworks.

Outcome: They are the expected results, for example availability, reduced production costs, quality.

3.2. *Search*

The search activity consists in formulating a search string, defining the research databases, inserting the string in the search tool of the selected databases, reading the search result which includes titles and summaries of articles and filtering the results, adding any articles already known by the researchers and also articles identified by the snowballing technique. According to Wohlin *et al.* [34], the search string often works differently with different databases. Search engines have different functionalities; for example, some allow searching only in the title of the article, and others allow searching in both titles and abstracts. Other technical limitations are related to the construction of the search string, to exacerbate the problem, there are differences in

how searches are executed internally by the database search engines. Therefore, the search expression must consider the specifications of each search engine, while maintaining its objective as much as possible. Below we describe the search activity

- Search string construction: The search string is defined based on keywords that guide the research topic. The keywords are derived from known articles and also by the PICO criteria.
- Database selection: This stage defines the databases that are likely to store articles that fit the purpose of the study and as such are considered relevant. These databases are usually large national and international repositories of scientific publications. They are public or private.
- Search: The elaborated search string is inserted by researchers in the search field of the selected database and the search engine will return a set of articles related to the searched string.
- Pre-reading: At this stage, the titles and abstracts of articles from the database search are reviewed manually by the researchers, to include or exclude them, forming a list, according to the pre-established criteria for the inclusion or exclusion of articles.
- Full reading: After listing the articles in the pre-reading stage, with an appraisal against the eligibility criteria, the selected articles are evaluated by the researchers through a full reading, with a second application of inclusion and exclusion criteria.
- Addition of other articles: Other articles known by the researchers and related to the study are included in the list.
- Snowballing: A sampling technique that consists of manual examination of the list of references of selected articles. Some of the articles listed in references are directly related to the search topic and therefore are manually included.

3.3. *Analysis and classification*

In this activity, an analysis is performed to extract and classify information from the set of articles collected in the search activity. The purpose here is to extract data from each article to answer the research questions formulated in the first activity. The extracted information is classified according to each research question. After classification, graphs are created with the help of statistic software to analyze the articles against the research questions. To ensure greater reliability, a quality assessment of the final set of articles is also conducted at this stage. In addition, to get an overview of how the collected articles relate to the search topic, a percentage of them is analyzed.

3.4. *Results and discussion*

In this activity, the graphs created in the previous stage are studied and discussed with focus on the research questions and the purpose of the study.

Conclusions regarding each answer to the research questions can be reached. This stage is essential for the study since, according to the results achieved, the researchers can comprehensively acknowledge the research topics investigated in the systematic mapping.

3.5. *Threats to validity*

According to Wohlin *et al.* [36], the validity of a study denotes the reliability of the results, and the extent to which the results are true and not biased by the researchers. There are different ways to classify aspects and threats to validity in the literature. In this study, we used the classification described in the book by Wohlin *et al.* [36]

- Construct Validity: Reflects the extent to which the operational measures studied truly represent the researcher's objective and the main topic investigated according to the research questions. This validation is determined by the researcher's ability to capture what is intended to be captured, that is, if the topics discussed in the interview questions are not equally interpreted by the researcher and the interviewed individuals, there is a threat to the construct validity.
- Internal Validity: This aspect of validity is a matter of consideration when causal relationships are examined. When the researcher is investigating whether a factor affects another investigated factor, there is a risk that the investigated factor might be affected by a third factor. If the researcher is not aware of the third factor and/or does not know to what extent it affects the investigated factor, there is a threat to internal validity.
- External Validity: Concerns the extent to which the results of the study can be generalized across other situations, the degree to which the conclusions would be of interest to others outside the investigated case. The external validity represents the application of the topic in general terms, that is, its scope, or utility and/or applicability. During the analysis of external validity, the researcher assesses the extent to which the results are relevant to other cases.
- Reliability: Refers to the extent to which data and analysis are biased towards the personal view of the researcher responsible for the study. Hypothetically, different researchers should produce similar results. Threats to reliability include lack of clarity in the encoding of collected data and in questionnaires and interview questions.

4. Study Implementation

This section presents the application of the methodology that we use to search for SBSE techniques in EAI. We use the Parsifal software[a] to help us classify the results in the following manner: First, we used Parsifal to update the list of articles returned by the search string. Second, we refined the list of included and excluded articles with

[a] https://parsif.al.

the help of Parsifal. Third, we used Parsifal to catalogue the articles according to the research questions entered by the researcher. Finally, at the end of the process, we used Parsifal to display the list of articles included in the mapping.

4.1. *Research questions*

The research questions for this systematic mapping study were initially guided by the definition of the research purpose:

> *Provide an overview of SBSE in EAI by answering different research questions, identifying trends in this scenario from the frequency/quantity of publications by category within a pre-established goal.*

The research questions formulated for this study are as follows:

- RQ1: In which journals, conference and workshop proceedings are articles published?
- RQ2: What are the trends of SBSE publications in EAI in recent years?
- RQ3: Which research groups and universities study SBSE in EAI?
- RQ4: Which optimization techniques are used?
- RQ5: What are the types of problems solved?

To apply the PICO strategy, the items related to the study's research topic were defined as follows:

Population: Does not apply, because the population will not be affected by our research.

Intervention: Techniques used by SBSE.

Comparison: No comparisons are made, we focus on understanding the area of application of SBSE in EAI and its applications.

Outcome: Have an overview of the studies that include Search-based Software Engineering and its application in enterprise integration platforms.

4.2. *Search*

The keywords identified to formulate the search string are shown in Fig. 2. They take into account PICO and the research questions listed in Sec. 4.1. In our systematic study, we used the following keywords: enterprise application integration, meta-heuristic, and search-based software engineering. Synonyms were also considered. The `AND` operator was used to separate term groups, while the `OR` operator was used to capture synonyms.

We evaluated and corrected the results obtained from the search engines (title, abstract and keywords) to improve the search string. For example, we left out articles returned by the search engine that matched keywords but were outside of the scope.

```
({search-based software engineering}
 OR optimization
 OR heuristic
 OR heuristics
 OR meta-heuristics
 OR {search-based techniques}
 OR {operational research}
 OR {decision-making})
AND ({application integration}
     OR {integration solution}
     OR {integration platform}
     OR {system integration}
     OR {integration process}
     OR {orchestration engine}
     OR {integration middleware})
```

Fig. 2. The search string used.

4.3. *Analysis and classification*

The main scientific databases selected for this systematic mapping were: ACM, Elsevier, IEEE and Springer and other databases of little relevance. Figure 3 shows the

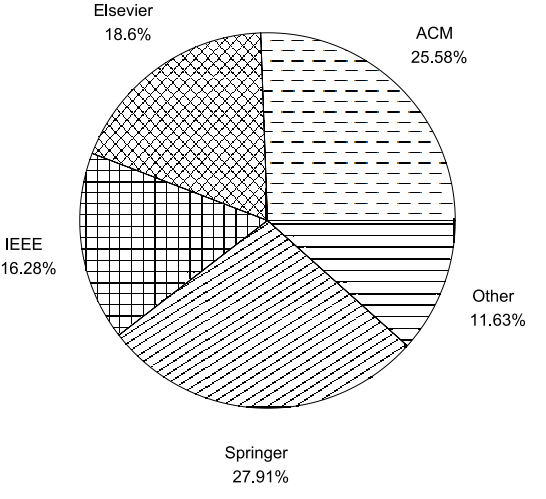

Fig. 3. Indexed databases by Scopus.

percentage of studies found per database. In the initial search, 560 articles were found. The title and abstract of the 560 articles were read as part of the screening process, resulting in the exclusion of 446 articles. The remaining 55 articles were evaluated with a full reading, resulting in 25 articles selected for final analysis. The inclusion criteria shown in Table 2 were applied to titles and abstracts; exclusion criteria were applied in the pre-reading stage and later during the full reading of the articles.

Table 2. Exclusion and inclusion criteria.

Criteria	Description
Exclusion	Articles representing conference summaries.
	Secondary or tertiary studies.
	Duplicate articles.
	Articles without full access permission.
	Articles not peer-reviewed.
	Books.
	Gray literature
Inclusion	The article addresses SBSE.
	Focuses on enterprise application integration.
	The article uses optimization techniques in Software Engineering.
	The article presents some connection between SBSE and enterprise application integration platforms.
	The article was published between 1999 and 2020.
	The article whether they should be indexed in JCR or emerging journal ranking.
	The conferences should be categorized with Scopus Score.

After applying inclusion and exclusion criteria to the set of articles, we included — on the basis of the authors' criteria — eight additional articles. In the snowballing, 12 additional articles were included, resulting in a total of 45 articles. The quality of the 45 articles was assessed using a simple scoring technique for two assessment questions:

- Is the article related to enterprise application integration?
- Does the article use any optimization techniques?

To select the articles used in the analysis and classification, accurately, we kept only the articles that received a positive response in the two questions.

4.4. *Results of mapping study*

Information was extracted from the articles to answer the research questions listed in Sec. 4.1. Data were extracted from the set of articles listed in the inclusion/exclusion and snowballing processes.

RQ1: Publishing forums

The primary studies yielded articles published in conferences, journals and workshops (see Fig. 4). Most of the publications (55%) appeared in conferences, 33% in journals and only 11% in workshops. Over the past five years, more articles have been published in conferences than in journals and workshops.

We found 2 in Information Systems, 2 in Proceedings of the International Conference on Enterprise Information Systems (ICEIS), 2 in CEUR Workshop Proceedings, 1 in Proceedings of the International Conference on Very Large Data Bases (VLDB), 1 in Advanced Engineering Informatics, 1 in Proceedings of the ACM

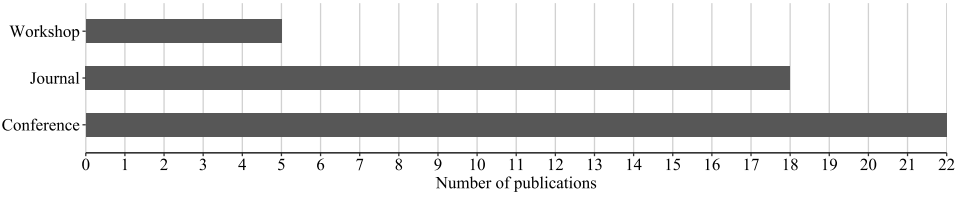

Fig. 4. Distribution of studies by venue types.

International Conference on Distributed and Event-based Systems (DEBS), 1 in Systems Engineering with, 1 in Proceedings of the ACM SIGMOD international conference on Management of data (SIGMOD/PODS), 1 in Computer Systems Science and Engineering, 1 in Workshop on Algorithms and Systems for MapReduce and Beyond (ACM SIGMOD), 1 in Computers in Industry, 1 in Wuhan University Journal of Natural Sciences, 1 in IEEE International Conference on Web Services (ICWS), 1 in International Conference on Intelligent Computer Communication and Processing (ICCP), 1 in International Journal of Advanced Computer Science and Applications, 1 in Proceedings of the Fourteenth East-European Conference on Advances in Databases and Information Systems (ADBIS), 1 in International Conference on Software Engineering (ICSE), 1 in Procedia Computer Science, 1 in Proceedings of the Nineteenth Annual ACM Symposium on Parallel Algorithms and Architectures (SPAA), 1 in Design Automation for Embedded Systems, Proceedings of the 2012 IEEE International Conference on Computer Supported Cooperative Work in Design (CSCWD), 1 in IEEE Congress on Evolutionary Computation (CEC), 1 in Journal of Computational Information Systems, 1 in INCOSE International Symposium, 1 in Proceedings of the ACM Conference on Information and Knowledge Management (CIKM), 1 in IEEE Ninth International Symposium on Parallel and Distributed Processing with Applications Workshops, 1 in International Conference Algorithms and Architectures for Parallel Processing (ICA3PP), 1 in Software Engineering, 1 in Computing Research Repository, 1 in Actual Problems of Systems and Software Engineering, Cluster Computing, 1 in Business Information Systems Workshops, 1 in Enterprise Information Systems, 1 in Journal of Systems and Software, 1 in Springer International Publishing, 1 in Data and Applications Security, 1 in International Conference on Logic Programming and Non-monotonic Reasoning, 1 in International Conference on Business Information Systems. Possibly, these are the events and journals where research results involving SBSE in EAI can be published or in which other articles related to this research topic can be found.

RQ2: Frequency of publications by years

The number of publications in each year was taken into account to determine the annual trend of SBSE in EAI. As shown in Fig. 5, although our search returned an article published in 1999, the next article retrieved was only published in 2005.

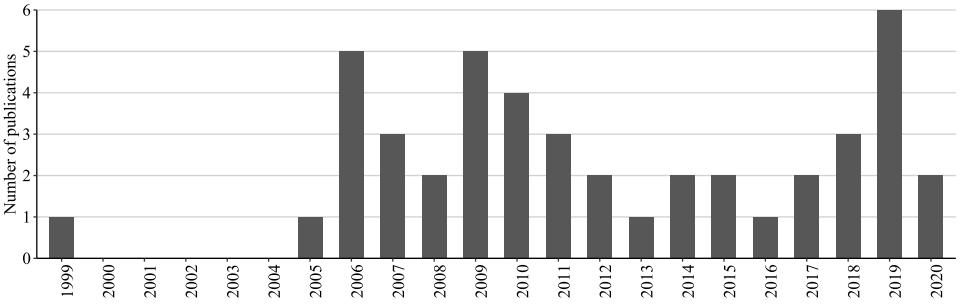

Fig. 5. Frequency of annual publications.

Therefore, we concluded that the rise of SBSE techniques in EAI began in 2005. Besides, there was a greater number of publications related to this topic between the years 2006 and 2010, followed by a decrease in the activities of the scientific community in the following years. Despite the decrease, there is still interest in the area, according to the last four years, mainly in 2019.

RQ3: Universities, companies, and authors

The universities/companies that have been conducting research related to SBSE in EAI were identified, along with authors that had more than one publication, as shown in Fig. 6. The author with the highest number of publications is Matthias

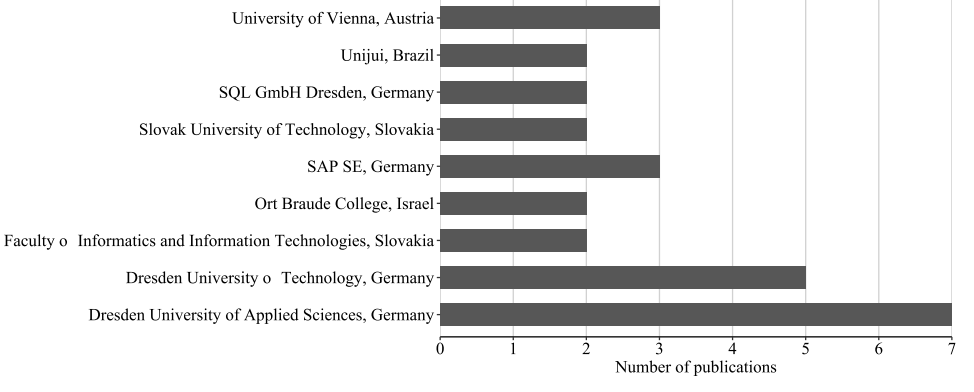

Fig. 6. Distribution of publications by university.

Boehm (7 publications), linked to Dresden University of Technology, followed by Dirk Habich and Wolfgang Lehner (5 publications) and Steffen Preissler (2 publications) both linked to Dresden University of Technology. The author Uwe Wloka (5 publications) is linked to Dresden University of Applied Sciences; Jürgen Bittner (2 publications) is linked to the company SQL GmbH Dresden; Pavol Mederly

(2 publications) is linked to the Faculty of Informatics and Information Technologies; Pavol Návrat is linked to the Slovak University of Technology and Daniel Ritter (3 publications) is linked to the company SAP SE. The author Stefanie Rinderle-Ma, who had 2 publications, is linked to the University of Vienna, Meir Tahan (2 publications) is linked to Ort Braude College and Daniela L. Freire (2 publications) is linked to the University of Unijui.

RQ4: Technique frequency

Regarding techniques used to solve the problems, as shown in Fig. 7, 48.83% of the articles use heuristic techniques of which 7 authors used the metaheuristic genetic

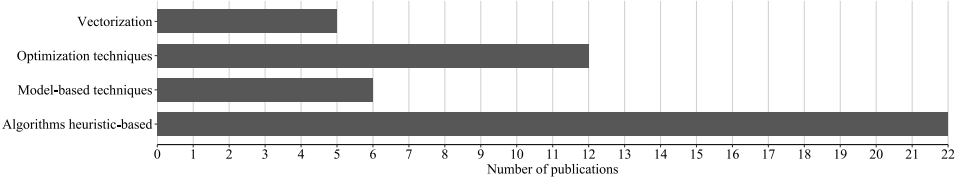

Fig. 7. Distribution of techniques used.

algorithm, 3 authors used the ant colony optimization and 2 others used bee swarm optimization. Another 10 authors addressed different heuristics. From the results, we found a considerable use of heuristic techniques, mainly the metaheuristic technique of the genetic algorithm applied in solving enterprise application integration problems. The use of vectoring and scheduling algorithms is also considered in the studies. The other techniques applied were specific to each problem.

RQ5: Frequency of problems

The analysis of the extracted data showed that a significant number of articles were focused on solving problems related to Runtime System performance (10 articles), see Fig. 8. Three other problems were frequently addressed: selection service composition (8 articles), design of integration (5 articles), business process optimization (4 articles) and integration on cloud computing (4 articles). Next we explain what each of these problems are as follows:

- **Runtime system performance:** Assesses the amount of computational resources consumed by the runtime system that performs the integration process in the platform, and its output. Solutions in this class of problems explore optimization techniques to improve efficiency of scheduling of computational resources. The aim is to increase the capacity of an integration process to process more input data per unit of time.

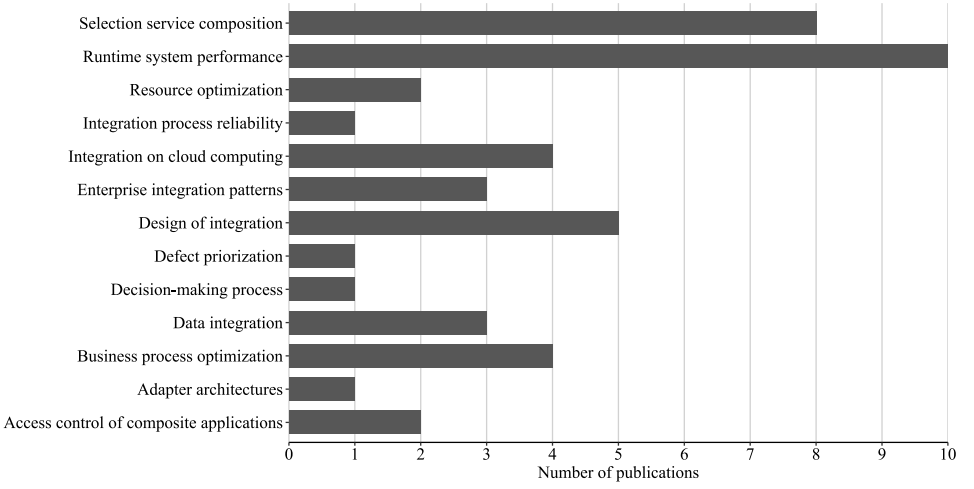

Fig. 8. Distribution of addressed problems.

- **Business process optimization:** A business processes is a set of tasks that collaborate (organized as a workflow) to produce a service or product. Business process optimization studies the use of optimization techniques in business processes. Solutions in this class of problems explore optimization techniques to find the shortest or cheapest (for example, in terms of time or money) path and to synchronize the interaction between the tasks to minimize idle times.
- **Enterprise integration patterns:** Is a catalogue of patterns to guide the design to develop integration application solutions/processors. The standardization of integration processes opens up possibilities that facilitate optimization. Enterprise integration patterns allow better planning of system optimization.
- **Adaptor architecture:** Adaptors are components used for communicating the integration process with applications. Due to their monolithic architecture and the need for data independence, adaptors can harm performance. Solutions in this class of problems explore approaches that improve the flexibility of adaptors.
- **Data integration:** Involves the combination of data from different sources in a transparent manner that provides the user with a unified view of the data. Sometimes processing of data takes place on different systems before reaching the application but it is done fast enough that becomes transparent to the application and the user. Solutions in this class of problems explore optimization techniques to improve data integration efficiency and system response time.
- **Service selection composition:** Concerns the selection of services to compose an integration process. It is all about choosing which services (out of several candidates) to use in the composition. The solutions in this class of problems explore optimization techniques for an optimal or near-optimal selection of the composing services which normally offer different properties. For example, they

offer different prices, service level agreements (say response time, throughput and availability) and data quality.

- **Integration on cloud computing:** Deals with the deployment and use of cloud computing resources for enterprise applications. Solutions in this class of problems explore optimization techniques for cloud application suitability. They are based on the estimation of resource demand to achieve some desirable properties such as best response time, service monitoring, and cost minimization.
- **Design of integration solutions:** Deals with the search of the best logical management of components during the integration process. Solutions in this class of problems explore optimization techniques to find the best service composition and optimize the integration process.
- **Access control of composite applications:** Involves verifying access rights of requests issued by applications. The aim is to verify what resources each request is trying to access and its permissions. Solutions in this class of problems explore techniques for Optimizing the control access of web service compositions. They are based on the estimation of the maximum number of authorized subjects.
- **Integration process reliability:** Refers to the need to monitor failures in integration process. Especially designed for heterogeneous systems, for countermeasures that can be taken as soon as failures are detected. Solutions in this class of problems explore optimization techniques to improve efficiency of the scheduling of computational resources aiming to maximize reliability and minimize makespan.
- **Optimization defect prioritization:** Defect prioritization is focused on finding the most potentially dangerous defects in large projects. The proposals in this class of problems explore optimization techniques to find priority defects. Thus, the development team focuses only on solving the defects prioritized in the optimization.
- **Decision-making process:** Consists in helping to choose the integration platform based on some requirements. The decision-making process is supported by criteria with objective functions aimed at the optimization of the process.
- **Optimizing use of resources:** Competition takes when several applications use resources from the same source. In this situations, delays become an issue unless careful optimization techniques are deployed. The class of optimization here concerns the application of techniques for maximizing the use of available resources without saturating them. The challenge is to deploy the minimum amount of resources to meet the expected quality of service.

4.5. *Threats to validity*

In this section, we identify threats that could have jeopardized our results and explain the countermeasures that we took to minimize them. The latter are based on the recommendations described by Wohlin *et al.* [36].

- To nullify the threat to construct validity, we used an online form that was revisited as the data were extracted. Accordingly, we carefully constructed the search string to perform the searches. Initially, only one search string was formulated. We refined it as needed after each application to the search engine. When needed, we changed operators to suit the search engine of each database.
- From the analysis of internal validity, we found two important aspects that would influence this threat. The first one is the fact that only one researcher performed the process of searching and selecting articles. To reduce this threat, we applied the snowballing technique. The results obtained were also validated by a second researcher and other articles were added based on the researcher's experience and knowledge. The second aspect is the search in multiple databases with a high probability of duplicate data. This threat was controlled through the use of a single search engine that indexes studies from multiples databases.
- The external validity was also examined. We ensured that the returned articles presented sufficient inputs and information for the composition of approaches that can be used in a systematic review. Additionally, this threat was reduced and reliability was guaranteed by the choice of the data source.
- The limited experience in systematic mapping studies of the first researcher could have jeopardized the results. We countermeasured this threat with the experience of the other researchers who have conducted several systematic mapping studies before.

5. Discussions

In this paper, we have presented a systematic mapping study to investigate the use of optimization techniques in the field of Enterprise Application Integration. We believe that our finding can help to comprehensively appreciate the state-of-the-art in this topic. Our results provide information about what types of problems EAI is helping to solve and what the most popular optimization techniques and research trends are. Our results also show what scholars, researchers and research groups are working on SBSE in EAI and what countries have published articles. Initially, our automatic search retrieved 560 articles in total. After applying inclusion and exclusion criteria we selected 45 articles for our analysis which yielded several facts.

We were able to verify that the veracity of the study was approximately 70%, that is, 25% of the articles included by the authors were left out of the search because the search did not find them. We also estimated that there is a probability that 5% of the articles were left out of the study because the search did not find them or because it was mistakenly discarded by the first filter applied to the 560 articles. This statistic was calculated based on the amount of already known articles that were left out of the search. We hope that our methodology and results provide a basis for conducting other mapping studies and systematic literature reviews. Furthermore, the extracted data can be reused as a catalogue for future research focused on the investigation of topics for similar purposes.

The results of our effort showed that optimization techniques have been explored by the EAI community over the last 21 years. The systematic mapping reported in Assunção *et al.* [14] covered the Brazilian SBSE community only. Our study is more comprehensive and is not restricted to Brazil. Furthermore, it also focused on SBSE applications for EAI. Our results showed that nearly half (48.83%) of the articles on optimization discuss the use of heuristics and metaheuristics to solve problems. This result demonstrates the interest in this topic.

Since the rise of SBSE several application areas in software engineering have benefited from optimization techniques. In EAI, optimization is used to optimize solutions to problems that concern the integration of heterogeneous applications, including workflow, service composition and quality of web services. The results have revealed that the most studied problem was related to runtime system performance and selection service composition. Furthermore, they show that the most used technique to solve such problems are heuristic techniques. The results reveal that industry is also producing research articles. This suggests that SBSE has already reached a level of maturity that makes it suitable for industrial applications.

We appreciate the effort of the research and industry communities, however, in our opinion some topics have not received enough attention or have been largely neglected. In the remaining of this section, we will briefly discuss some topics that, in our opinion, need attention to progress SBSE in EAI beyond its current state-of-the-art. To start with, we believe that it is time to progress research beyond performance and explore heuristics with optimal solutions or close to optimum in order to efficiently use system resources. In fact, this gap is in the research agenda of our research group. In the near future will focus on the development of a new heuristic based on the existing ones so that in addition to performance, maximum efficiency is also achieved in system of integration.

Data confidentiality is another topic that is closely related to EAI and has not received enough attention. The issue is that EAIs are likely to involve resources (for example data) from several independent sources; for example from different departments within a single company, that do not necessarily share information; likewise, an EAI solution might consume data from independent companies, including business competitors. Thus, data consumed by EAIs are not necessarily public, for example, intellectual properties and personal sensitive data might be involved, therefore, we feel that data security is central to EAI. It is sensible to expect that data providers might be reluctant to participate in EAI solutions unless the confidentiality of their data is guaranteed both in transit and execution time. Technologies for addressing these issues are currently under exploration under the name of trusted computing; they should, in our opinion, be also explored by researchers in SBSE in EAI. In this regard, specific technologies are trusted hardware (for example, Intel SGX platform [37] and ARM TrustZone [38]) and homomorphic encryption [39]. Trusted hardware can also be used for building Information Flow Control (IFC) mechanisms to prevent EAIs from illegally transferring data received from their sources to other parties [40].

Another emerging technology of which EAI solutions can take advantage of is smart contracts. In the existing work, in EAI participants provide their resources to EAI solutions under no commitments or expectations, this is a reasonable assumption when the EAI solution involves only applications from a single company; however, there is a large class of EAI solutions that involve several independent companies under strict commitments (for example, service level agreements). Such commitments can be implemented as smart contracts (also called executable contracts) deployed on trusted third parties or blockchains and enforced at runtime [41]. This is a topic that in our opinion deserves further exploration. In fact, the implementation of smart contracts can benefit from optimization. In practice, business contracts are not converted fully into smart contracts: only some of their clauses are, others are not coded but left for manual execution for efficiency reasons or because they cannot be expressed in computer code. We feel that research in SBSE can help in the implementation of efficient contracts. Researchers from the SBSE community can help to develop optimization methodologies to determine informatively what clauses of a given contract are likely to be executed more efficiently with human support [42] than automatically.

6. Conclusions

We have presented a systematic mapping study where we investigate the literature on the use of optimization techniques in enterprise application integration (SBSE in EAI). Our study can help gain a global and comprehensive view of research activities conducted between 1999 and 2020 on this topic. Our results identified the problems that have been subject of research, most used optimization techniques and trendy topics. Also, it identifies scholars, research groups and countries with active researchers in these topics. Our study yielded two important observations. First, we discovered that one of the main optimization techniques used to solve EAI problems in recent years have been algorithms based on heuristic techniques. Second, the study revealed that in the field of EAI there are still problems that need research attention such as runtime system performance and selection service composition. We will take advantage of our results in our future work and pay attention to runtime system performance problems and the application of heuristic techniques. We believe that the results of this study may help a wide range of researchers in the field of Software Engineering interested in finding optimization problems and techniques applied to EAI. Most importantly, we hope that our study will help them to highlight research gaps.

Acknowledgments

This work was supported by the Brazilian Coordination for the Improvement of Higher Education Personnel (CAPES); the National Council for Scientific and Technological Development (CNPq) under grant 309315/2020-4; and the Research Support Foundation of Rio Grande do Sul (FAPERGS) under grant 17/2551-0001206-2.

Appendix A. Articles

Table A.1 shows the articles that were included in our mapping study. The articles that were categorized in Sec. 4 of our systematic mapping study are also listed in

Table A.1. Articles included during the search activity.

Action	Articles
Articles included from search	[43–54] [55–67]
Articles included through snowballing	[68–79]
Articles included ad-hoc by us	[80–87]

Table A.2. Articles by indexed database searched.

Database	Articles
ACM	[68, 72, 73, 47, 54, 74, 59, 62–64, 84, 87]
Elsevier	[43, 44, 56, 82, 61, 76, 66, 86]
IEEE	[71, 52, 57, 58, 60, 77, 83]
Springer	[45, 69, 48, 50, 55, 80, 75, 81, 78, 65, 79, 85, 67]
Other	[70, 46, 49, 53, 51]

Table A.3. Articles by publication venue type.

Venues	Articles
Workshops	[57, 62, 79, 46, 49]
Journals	[43, 44, 69, 70, 48, 53, 54, 56, 82, 61, 76, 66, 85, 81] [78, 84, 86, 87]
Conferences	[68, 71, 45, 72, 73, 47, 50–52, 55] [74, 80, 75, 58–60, 77, 63–65, 83, 67]

Table A.4. Articles by year of publication.

Years	Articles
1999	[68]
2005	[43]
2006	[44, 45, 69–71]
2007	[46, 72, 73]
2008	[47, 48]
2009	[50–53, 49]
2010	[54, 55, 74, 80]
2011	[56, 57, 75]
2012	[58, 81]
2013	[59]
2014	[60, 82]
2015	[61, 76]
2016	[77]
2017	[62, 63]
2018	[64, 78, 65]
2019	[66, 83, 79, 84–86]
2020	[87, 67]

Table A.5. Articles by used techniques.

Techniques	Articles
Vectorisation	[46, 50, 74, 56, 82]
Model-based techniques	[68, 43, 71, 44, 58, 62]
Optimization techniques	[72, 52, 51, 75, 60, 76, 61, 63, 64, 78, 84, 79]
Algorithms heuristic-based	[69, 70, 45, 73, 48, 49, 47, 53, 80, 54, 55, 57, 81] [59, 77, 65, 85, 86, 66, 83, 87, 67]

Table A.6. Articles by research problem addressed.

Problems	Articles
Runtime system performance	[72, 46, 47, 50, 52, 74, 56, 82, 62, 87]
Business process optimization	[43, 44, 75, 58]
Enterprise integration patterns	[63, 64, 84]
Adapter architectures	[49]
Data integration	[68, 51, 79]
Selection service composition	[69–71, 54, 55, 57, 77, 78]
Integration on cloud computing	[59, 76, 61, 83]
Design of integration	[48, 80, 81, 60, 67]
Access control of composite applications	[45, 53]
Integration process reliability	[73]
Defect prioritization	[85]
Decision-making process	[86]
Resource optimization	[65, 66]

this appendix. Table A.2 corresponds to Fig. 3 and lists the articles grouped by indexed databases. Table A.3 corresponds to Fig. 4 and lists the articles grouped by venue types of publications. Table A.4 corresponds to Fig. 5 and lists the corresponding articles grouped by year of publications. Table A.5 corresponds to Fig. 7 and lists the articles grouped by techniques used. Table A.6 corresponds to Fig. 8 and lists the corresponding articles grouped by problems addressed. Table A.7 corresponds to Fig. 6 and lists the corresponding articles grouped by universities.

Table A.7. Articles by universities and companies.

Universities/Companies	Articles
Dresden University of Technology, Germany	[46, 49, 47, 50, 74, 56, 82]
Dresden University of Applied Sciences, Germany	[46, 49, 47, 50, 56]
Faculty of Informatics and Information Technologies, Slovakia	[80, 81]
Slovak University of Technology, Slovakia	[80, 81]
University of Vienna, Austria	[63, 64, 84]
SQL GmbH Dresden, Germany	[46, 49]
SAP SE, Germany	[63, 64, 84]
Ort Braude College, Israel	[48, 58]
Unijui, Brazil	[86, 87]

References

1. K. Manikas, Revisiting software ecosystems research: A longitudinal literature study, *J. Syst. Softw.* **117** (2016) 84–103.
2. G. Hohpe and B. Woolf, *Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions* (Addison Wesley, Boston, MA, USA, 2011).
3. R. Z. Frantz, M. H. Rehbein, R. Berlezi and F. Roos-Frantz, Ranking open source application integration frameworks based on maintainability metrics: A review of five-year evolution, *Softw. - Pract. Exp.* **49**(10) (2019) 1531–1549.
4. R. Z. Frantz, R. Corchuelo, V. Basto-Fernandes, F. Rosa-Sequeira, F. Roos-Frantz and J. L. Âarjona, A cloud-based integration platform for enterprise application integration: A model-driven engineering approach, *Softw. - Pract. Exp.* **51**(4) (2021) 824–847.
5. D. Ritter, N. May, F. N. Forsberg and S. Rinderle-Ma, Optimization strategies for integration pattern compositions, in *ACM Int. Conf. Distributed and Event-Based Systems*, 2018, pp. 88–99.
6. D. Romero and F. Vernadat, Enterprise information systems state of the art: Past, present and future trends, *Comput. Ind.* **79** (2016) 3–13.
7. G. R. Santhanam, Qualitative optimization in software engineering: A short survey, *J. Syst. Softw.* **111** (2016) 149–156.
8. M. Harman, S. A. Mansouri and Y. Zhang, Search-based software engineering: Trends, techniques and applications, *ACM Comput. Surv.* **45**(1) (2012) 11–61.
9. M. Harman and B. F. Jones, Search-based software engineering, *Inf. Softw. Technol.* **43**(14) (2001) 833–839.
10. B. Kitchenham and S. Charters, Guidelines for performing systematic literature reviews in software engineering, in *Proc. Int. Workshop on Evidential Assessment of Software Technologies*, 2007, p. 1–2.
11. K. Petersen, S. Vakkalanka and L. Kuzniarzk, Guidelines for conducting systematic mapping studies in software engineering: An update, *Inf. Softw. Technol.* **64**(100) (2015) 1–18.
12. A. Aleti, B. Buhnova, L. Grunske, A. Koziolek and I. Meedeniya, Software architecture optimization methods: A systematic literature review, *IEEE Trans. Softw. Eng.* **39**(5) (2013) 658–683.
13. A. Pitangueira, R. S. Maciel, M. Barros and A. Andrade, A systematic review of software requirements selection and prioritization using SBSE approaches, in *5th Int. Symp. Search Based Software Engineering*, 2013, pp. 188–208.
14. W. K. Assunção, M. de O Barros, T. E. Colanzi, A. C. Dias-Neto, M. H. Paixão, J. T. de Souza and S. R. Vergilio, A mapping study of the brazilian SBSE community, *J. Softw. Eng. Res. Devel.* **2**(1) (2014) 3–19.
15. D. Meignan, S. Knust, J.-M. Frayret, G. Pesant and N. Gaud, A review and taxonomy of interactive optimization methods in operations research, *Trans. Interact. Intell. Syst.* **5**(10) (2015) 1–43.
16. A. M. Pitangueira, R. S. P. Maciel and M. Barros, Software requirements selection and prioritization using SBSE approaches: A systematic review and mapping of the literature, *J. Syst. Softw.* **103** (2015) 267–280.
17. T. Ambreen, N. Ikram, M. Usman and M. Niazi, Empirical research in requirements engineering: Trends and opportunities, *Requir. Eng.* **23**(1) (2016) 63–95.
18. T. Mariani and S. R. Vergilio, A systematic review on search-based refactoring, *Inf. Softw. Technol.* **83** (2017) 14–34.
19. R. A. Silva, S. do Rocio Senger de Souza and P. S. L. de Souza, A systematic review on search based mutation testing, *Inf. Softw. Technol.* **81** (2017) 19–35.

20. A. Vegendla, A. N. Duc, S. Gao and G. Sindre, A systematic mapping study on requirements engineering in software ecosystems, *J. Inform. Technol. Res.* **11**(1) (2018) 49–69.
21. D. Ameller, X. Burgués, D. Costal, C. Farré and X. Franch, Non-functional requirements in model-driven development of service-oriented architectures, *Sci. Comput. Program.* **168** (2018) 18–37.
22. R. Ros and P. Runeson, Continuous experimentation and A/B testing: A mapping study, in *Int. Conf. Software Engineering*, 2018, pp. 35–41.
23. H. G. Gurbuz and B. Tekinerdogan, Model-based testing for software safety: A systematic mapping study, *Softw. Q. J.* **26**(4) (2018) 1327–1372.
24. J. L. Barros-Justo, F. B. Benitti and A. L. Cravero-Leal, Software patterns and requirements engineering activities in real-world settings: A systematic mapping study, *Comput. Stand. Interf.* **58** (2018) 23–42.
25. M. Villalobos-Arias, C. Quesada-López, A. Martínez-Abraín and M. A. Jenkins, A tertiary study on model-based testing areas, tools and challenges: Preliminary results, in *Ibero-American Conf. Software Engineering* (CIbSE, Bogotá, Colombia, April 2018), pp. 15–28.
26. E. F. Franco, K. Hirama and M. M. Carvalho, Applying system dynamics approach in software and information system projects: A mapping study, *Inf. Softw. Technol.* **93** (2018) 58–73.
27. T. Tahir, G. Rasool and M. Noman, A systematic mapping study on software measurement programs in SMES, *E-Inform. Softw. Eng. J.* **12**(1) (2018) 133–165.
28. J. Souza, A. A. Araújo, R. Saraiva, P. Soares and C. Maia, A preliminary systematic mapping study of human competitiveness of SBSE, *Int. Symp. Search Based Software Engineering* **11036** (2018) 131–146.
29. A. Ramírez, J. R. Romero and C. L. Simons, A systematic review of interaction in search-based software engineering, *IEEE Trans. Softw. Eng.* **45**(8) (2019) 760–781.
30. G. A. C. V. Real, J. C. Nardi, M. P. Barcellos and R. de Almeida Falbo, Semantic enterprise application integration at service layer a systematic mapping, in *Ibero-American Conf. Software Engineering*, 2019, pp. 17–31.
31. M. E. Morales-Trujillo, E. O. Matla-Cruz, G. A. GarcÃa-Mireles and M. Piattini, Privacy by design in software engineering: A systematic mapping study, *CLEI Electron. J.* **22** (2019) 1–29.
32. A. Ramírez, P. Delgado-Pérez, J. Ferrer, J. Romero, I. Medina-Bulo and F. Chicano, A systematic literature review of the SBSE research community in spain, *Prog. Artif. Intell.* **9** (2020) 113–128.
33. L. D. Cerqueira, J. C. Nardi, M. P. Barcellos and R. A. Falbo, Process integration in semantic enterprise application integration: A systematic mapping, in *Int. Conf. Product-Focused Software Process Improvement*, 2016, p. 13.
34. C. Wohlin, P. Runeson, P. A. da Mota Silveira Neto, I. D. C. M. Emelie Engström and E. S. de Almeidac, On the reliability of mapping studies in software engineering, *J. Syst. Softw.* **86**(10) (2013) 2594–2610.
35. B. A. Kitchenham and D. Budgen, Evidence-based software engineering and systematic literature reviews, *Inf. Softw. Technol.* **51** (2006) 3.
36. C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell and A. Wesslén, *Experimentation in Software Engineering: An Introduction* (Kluwer Academic Publishers, Berlin, Heidelberg, 2000).
37. V. Costan and S. Devadas, Intel SGX explained, 2016, https://eprint.iacr.org/2016/086.pdf.

38. S. Pinto and N. Santos, Demystifying arm trustzone: A comprehensive survey, *ACM Comput. Surv.* **51** (2019) 1–36.
39. C. Gentry, Computing arbitrary functions of encrypted data, *Commun. ACM* **53** (2010) 9.
40. A. H. Bell-Thomas, Trusted reference monitors for linux using intel SGX enclaves, Master's thesis, Computer Laboratory University of Cambridge (2020).
41. C. Molina-Jimenez, I. Sfyrakis, L. Song, I. Ng, M. W. Wong, A. Chun and J. Crowcroft, Implementation of smart contracts using hybrid architectures with on and off-blockchain components, 2019, https://www.researchgate.net/publication/329561864 (January, 2019), Visited on 12 April 2020.
42. C. D. Clack, V. A. Bakshi and L. Braine, Smart contract templates: Foundations, design landscape and research directions, 2020, https://arxiv.org/abs/1608.00771.
43. Y. Caseau, Self-adaptive middleware: Supporting business process priorities and service level agreements, *Adv. Eng. Inf.* **19**(3) (2005) 199–211.
44. Q. Ni, W. F. Lu, P. K. Yarlagadda and X. Ming, A collaborative engine for enterprise application integration, *Comput. Ind.* **57**(7) (2006) 640–652.
45. M. Wimmer, A. Kemper, M. Rits and V. Lotz, Consolidating the access control of composite applications and workflows, in *Data and Applications Security XX*, 2006, pp. 44–59.
46. M. Boehm, D. Habich, U. Wloka, J. Bittner and W. Lehner, Towards self-optimization of message transformation processes, *CEUR Workshop Proceedings* **325**(1) (2007) 116–125.
47. M. Boehm, U. Wloka, D. Habich and W. Lehner, Workload-based optimization of integration processes, in *Proc. ACM Conf. Information and Knowledge Management*, 2008, pp. 1479–1480.
48. M. Tahan and J. Z. Ben-Asher, Modeling and optimization of integration processes using dynamic programming, *Syst. Eng.* **11**(2) (2008) 165–185.
49. M. Boehm, J. Bittner, D. Habich, W. Lehner and U. Wloka, Model-driven generation of dynamic adapters for integration platforms, *CEUR Workshop Proc.* **340**(5) (2009) 105–119.
50. M. Boehm, D. Habich, S. Preissler, W. Lehner and U. Wloka, Vectorizing instance-based integration processes, in *Int. Conf. Enterprise Information Systems*, 2009, pp. 40–52.
51. N. Leone, F. Ricca and G. Terracina, An ASP-based data integration system, in *Int. Conf. Logic Programming and Nonmonotonic Reasoning*, 2009, pp. 528–534.
52. F. Pop, C. Dobre and V. Cristea, Genetic algorithm for DAG scheduling in grid environments, in *Int. Conf. Intelligent Computer Communication and Processing*, 2009, pp. 299–305.
53. S. Islam, N. Suri, A. Balogh, G. Csertán and A. Pataricza, An optimization based design for integrated dependable real-time embedded systems, *Des. Autom. Embed. Syst.* **13**(4) (2009) 45–285.
54. Z. Yang, C. Shang, Q. Liu and C. Zhao, A dynamic web services composition algorithm based on the combination of ant colony algorithm and genetic algorithm, *J. Comput. Inf. Syst.* **6**(8) (2010) 2617–2622.
55. W. Li and H. Y. Xiang Computer, A web service composition algorithm based on global qos optimizing with mocaco, in *Int. Conf. Algorithms and Architectures for Parallel Processing*, 2010, pp. 218–224.
56. M. Boehm, D. Habich, S. Preissler, W. Lehner and U. Wloka, Cost-based vectorization of instance-based integration processes, *Inf. Syst.* **36**(1) (2011) 3–29.
57. G. Kousalya, D. Palanikkumar and P. R. Piriyankaa, Optimal web service selection and composition using multi-objective bees algorithm, in *IEEE Ninth Int. Symp. Parallel and Distributed Processing with Applications Workshops*, 2011, pp. 193–196.

58. M. Tahan and R. Benish, Model based tool for planning optimal integration processes under severe uncertainty, *INCOSE Int. Symp.* **22**(1) (2012) 1387–1398.
59. S. Frey, F. Fittkau and W. Hasselbringn, Conformance checking and evolutionary optimization for deployment and reconfiguration of software in the cloud, in *Int. Conf. Software Engineering*, 2013, pp. 512–521.
60. Y. Yu, H. Ma and M. Zhang, A genetic programming approach to distributed QoS-aware web service composition, in *IEEE Congress Evolutionary Computation*, 2014, pp. 1840–1846.
61. I. Hernández, S. Sawicki, F. Roos-Frantz and R. Z. Frantz, Cloud configuration modelling: A literature review from an application integration deployment perspective, *Procedia Comput. Sci.* **64**(4) (2015) 977–983.
62. Y. Kaniovskyi, M. Koehler and S. Benkner, A containerized analytics framework for data and compute-intensive pipeline applications, in *Workshop on Algorithms and Systems for MapReduce and Beyond*, 2017, pp. 6–10.
63. D. Ritter, J. Dann, N. May and S. Rinderle-Ma, Hardware accelerated application integration processing: Industry paper, in *Proc. ACM Int. Conf. Distributed and Event-Based Systems*, 2017, pp. 215–226.
64. D. Ritter, N. May, F. N. Forsberg and S. Rinderle-Ma, Optimization strategies for integration pattern compositions, in *Proc. ACM Int. Conf. Distributed and Event-Based Systems*, 2018, pp. 88–99.
65. D. Nasonov, N. Butakov, M. Melnik, A. Visheratin, A. Linev, P. Shvets, S. Sobolev and K. Mukhina, The multi-level adaptive approach for efficient execution of multi-scale distributed applications with dynamic workload, in *Supercomputing*, eds. V. Voevodin and S. Sobolev (Springer International Publishing, Berlin, 2018), pp. 675–686.
66. D. E. Singh and J. Carretero, Combining malleability and I/O control mechanisms to enhance the execution of multiple applications, *J. Syst. Softw.* **148** (2019) 21–36.
67. A. Drewes, J. M. Joseph, B. Gurumurthy, D. Broneske, G. Saake and T. Pionteck, Optimising operator sets for analytical database processing on FPGAs, in *Applied Reconfigurable Computing. Architectures, Tools, and Applications*, 2020, pp. 30–44.
68. Z. G. Ives, D. Florescu, M. Friedman and A. Levy, An adaptive query execution system for data integration, in *Proc. 1999 ACM SIGMOD Int. Conf. Management of Data*, 1999, pp. 299–310.
69. Y. Gao, J. Na, B. Zhang, L. Yang, Q. Gong and Y. Dai, Immune algorithm for selecting optimum, *Wuhan Univ. J. Nat. Sci.* **1202**(1) (2006) 221–225.
70. L. Aversano, M. di Penta and K. Taneja, A genetic programming approach to support the design of service compositions, *Comput. Syst. Sci. Eng.* **21**(4) (2006) 247–254.
71. R. Berbner, M. Spahn, N. Repp, O. Heckmann and R. Steinmetz, Heuristics for QoS-aware web service composition, in *IEEE Int. Conf. Web Services*, 2006, pp. 72–82.
72. M. Vrhovnik, H. Schwarz, O. Suhre, B. Mitschang, V. Markl, A. Maier and T. Kraft, An approach to optimize data processing in business processes, in *Proc. Int. Conf. Very Large Data Bases*, 2007, pp. 615–626.
73. J. J. Dongarra, E. Saule, E. Jeannot and Z. Shi, Bi-objective scheduling algorithms for optimizing makespan and reliability on heterogeneous systems, in *Proc. Nineteenth Annual ACM Symp. Parallel Algorithms and Architectures*, 2007, pp. 280–288.
74. M. Boehm, D. Habich and W. Lehner, Multi-flow optimization via horizontal message queue partitioning, in *Proc. Int. Conf. Enterprise Information Systems*, 2010, pp. 5–14.
75. F. Niedermann, S. Radeschütz and B. Mitschang, Business process optimization using formalized optimization pattern, in *Int. Conf. Business Information Systems* (ACM, Berlin, Heidelberg, May 2011), pp. 123–135.

76. E. Weintraub and Y. Cohen, Cost optimization of cloud computing services in a networked environment, *Int. J. Adv. Comput. Sci. Appl.* **6**(4) (2015) 148–157.
77. L. Wang, J. Shen and J. Yong, A survey on bio-inspired algorithms for web service composition, in *Proc. 2012 IEEE Int. Conf. Computer Supported Cooperative Work in Design*, 2016, pp. 1–6.
78. U. Arul and S. Prakash, A unified algorithm to automatic semantic composition using multilevel workflow orchestration, *Cluster Comput.* **22** (2018) 15387–15408.
79. F. Kruse, Towards a record linkage layer to support big data integration, in *Business Information Systems Workshops*, eds. W. Abramowicz and R. Corchuelo, 2019, pp. 625–636.
80. P. Mederly and P. Návrat, Construction of messaging-based integration, in *Proc. Fourteenth East-European Conf. Advances in Databases and Information Systems* (ACM, Berlin, Heidelberg, May 2010), pp. 579–582.
81. P. Mederly, M. Lekavý, M. Závodský and P. Návrat, Construction of messaging-based enterprise integration solutions using AI planning, in *Central and East European Conf. Software Engineering Techniques* (ACM, Berlin, Heidelberg, September 2012), pp. 16–29.
82. M. Boehm, D. Habich and W. Lehner, On-demand re-optimization of integration flows, *Inf. Syst.* **45** (2014) 1–17.
83. S. Frenkel, V. Zakharov and B. Basok, Optimization of the integration process of cloud and COTS based computing systems, in *Actual Problems of Systems and Software Engineering*, 2019, pp. 90–96.
84. D. Ritter, F. N. Forsberg, S. Rinderle-Ma and N. May, Catalog of optimization strategies and realizations for composed integration patterns, *Comput. Res. Repos.* arXiv:1901.01005.
85. V. Gupta, D. Kumar and P. K. Kapur, Optimizing the defect prioritization in enterprise application integration, in *Software Engineering* (Springer, 2019), pp. 585–597.
86. D. Freire, R. Z. Frantz, F. Roos-Frantz and S. Sawicki, A methodology to rank enterprise application integration platforms from a performance perspective: An analytic hierarchy process-based approach, *Enterp. Inf. Syst.* **13** (2019) 1292–1322.
87. D. L. Freire, R. Z. Frantz and F. Roos-Frantz, Towards optimal thread pool configuration for run-time systems of integration platforms, *Int. J. Comput. Appl. Technol.* **62** (2020) 129–147.