A systematic literature review of the SBSE research community in Spain
— *Review Protocol* —

Aurora Ramírez[1], Pedro Delgado-Pérez[2], Javier Ferrer[3],

José Raúl Romero[1], Inmaculada Medina-Bulo[2] and Francisco Chicano[3]

June 28, 2019

Table of contents

[1] Dpto. Informática y Análisis Numérico, University of Córdoba, {aramirez, jrromero}@uco.es
[2] Dpto. Ingeniería Informática, University of Cádiz, {pedro.delgado, inmaculada.medina}@uca.es
[3] Dpto. Lenguajes y Ciencias de la Computación, University of Málaga, {ferrer, chicano}@lcc.uma.es

1. Review procedure

The review process is organised in three main phases: (1) planning the review, (2) conducting the review and (3) reporting the review. After each phase, a meeting discussion among all nodes[2] is carried out. During each phase, tasks are distributed among participants, enabling effective mechanisms for communication between nodes when necessary.

1. Planning the review
    a. Definition of the review methodology and classification scheme
    b. Pilot automatic search of papers
    c. Pilot manual search of authors
2. Conducting the review
    a. Selection of primary studies
    b. Data extraction process
    c. Resolution of disagreements
3. Reporting the review
    a. Statistical analysis
    b. Narrative description

2. Search process

2.1. Search strategy

The search strategy combines automatic and manual search with the aim of discovering the researchers working on SBSE within the Spanish community, as well as their contribution to the field. Firstly, a number of potential papers will be extracted. They will be considered candidates to be included in the review. Secondly, these papers should be filtered according to well-defined inclusion and exclusion criteria. The resulting set of papers, also known as primary studies, will constitute the basis for the data extraction process.

The following tasks must be carried out, as depicted in Figure 1:

1. Definition of search strings.
    a. *Input*: list of terms related to SBSE, instructions of search engines[3].
    b. *Output*: search strings to extract SBSE publications in Spain.
2. Execution of automatic search filtered by country.
    a. *Input*: search strings.
    b. *Output*: list of SBSE papers co-authored by Spanish researchers.
3. Manual filtering of authors by institution.
    a. *Input*: list of SBSE papers co-authored by Spanish researchers.
    b. *Output*: automatic list of authors.
4. Manual search of authors.
    a. *Input*: list of institutional and personal web pages.
    b. *Output*: manual list of authors.
5. Generation of the Spanish SBSE community.
    a. *Inputs*: manual and automatic lists of authors.
    b. *Output*: definitive list of authors.

[2] A node is composed by the members of the same institution, , namely University of Cádiz (UCA), University of Córdoba (UCO) and University of Málaga (UMA).

[3] Only those databases that allow filtering by country will be considered.

6. Manual search of papers in Spanish language.
    a. *Input*: electronic proceedings of Spanish conferences.
    b. *Output*: list of candidate papers published in Spanish events.
7. Automatic search of candidate papers by author.
    a. *Inputs*: definitive list of authors, researcher's online profiles.
    b. *Output*: list of candidate papers, separated into Spanish and international.
8. Filtering of primary studies from candidate papers.
    a. *Inputs*: list of candidate papers (both Spanish and international research works), inclusion and exclusion criteria.
    b. *Output*: list of primary studies, separated into Spanish and international.

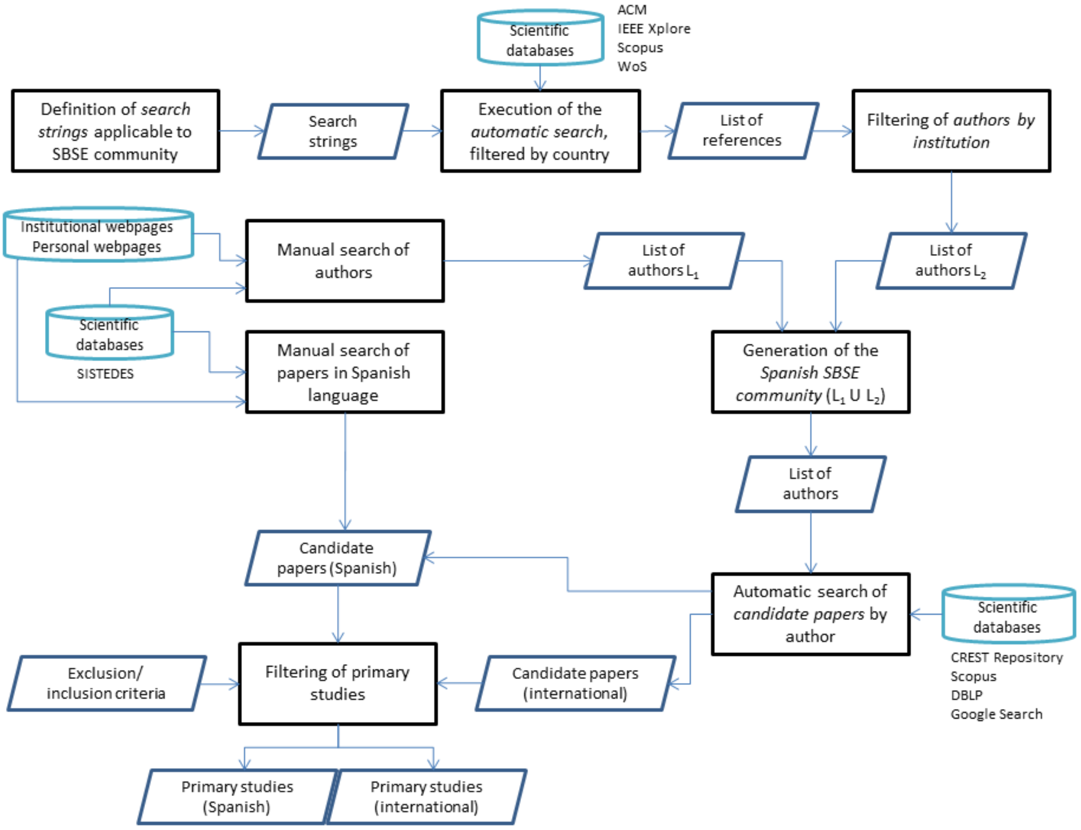
Scientific databases
ACM
IEEE Xplore
Scopus
WoS
Definition of search strings applicable to SBSE community
Search strings
Execution of the automatic search, filtered by country
List of references
Filtering of authors by institution
Institutional webpages
Personal webpages
Manual search of authors
List of authors L1
List of authors L2
Scientific databases
SISTEDES
Manual search of papers in Spanish language
Generation of the Spanish SBSE community (L1 U L2)
List of authors
Candidate papers (Spanish)
Automatic search of candidate papers by author
Scientific databases
CREST Repository
Scopus
DBLP
Google Search
Exclusion/ inclusion criteria
Filtering of primary studies
Candidate papers (international)
Primary studies (Spanish)
Primary studies (international)

Figure 1. Workflow of the overall search process.

2.2. Search strings

A number of search strings are defined according to the existing guidelines within the SBSE field (Harman et al., 2012) and considering any restriction imposed by search engines. Initially, pilot searches have been conducted for all databases to observe how they behave when the strings proposed by Harman et al. are queried. Then, these strings are significantly adapted to properly represent the generality required by current trends of SBSE, so that long strings and too specific terms (e.g. very specific search techniques) are avoided. In this way, a great and relevant variety of studies are not excluded. Besides, some complex strings are difficult to execute by search engines.

1. A common set of generic terms is considered to represent the adoption of SBSE, as a discipline based on the application of search or optimisation approaches: *search*, *search-based*, *optimisation*[4].
2. A significant number of topics are queried in order to cover the different phases of the software engineering process. In general, apart from the descriptive term for general software engineering phase, as well as variations, a number of terms usually associated to these phases in SBSE papers have been also queried:
    a. Software planning: *cost estimation, effort estimation, project planning, project management, project scheduling*.
    b. Requirement engineering: *next release problem, requirements selection, requirements prioritisation, software requirements, software specifications*.
    c. Software design: *object-oriented design, service-oriented design, software architecture, software design, software product lines*.
    d. Implementation: *code, genetic improvement, program slices*.
    e. Testing and validation: *model checking, model transformation, test case, test data, testing*.
    f. Deployment and integration: *component allocation, service composition, software deployment, software migration, component integratio*n.
    g. Maintenance: *maintenance, modularisation, refactoring*.
3. Search is filtered by the author's affiliation country (Spain).
4. Search is filtered by publication date (2001-2019).

The resulting *search strings* have the following general structure:

```
("search"  OR  "search-based"  OR  "optimization"  OR  "optimisation") AND(<list of terms of SE phase>)

AND(affiliation=SPAIN)

AND(time=[2001,2019])
```

For manual search, the Google's search engine is executed with some of the following commands:

```
site: <web page> "sbse"AND(affiliation=SPAIN)

site: <web page> "sbse" software

site: <web page> "search based software engineering"
```

For manual search, Google Scholar is also considered with the terms `SBSE` and `Search Based Software Engineering`.

2.3 Data sources

Search has been performed by gathering information from the following data sources:

- ACM Library [1,2][5]: http://dl.acm.org/
- DBLP [7]: http://dblp.uni-trier.de/
- CREST Repository [7]: http://crestweb.cs.ucl.ac.uk/resources/sbse_repository/
- Google Scholar [4,7]: https://scholar.google.com

[4] When required, both British and American styles are worded.
[5] Numbers in brackets stand for the step of the search procedure where this search string has been applied.

- IEEE Xplore [1,2]: http://ieeexplore.ieee.org/
- Scopus [1,2,7]: https://www.scopus.com/
- SISTEDES library (in Spanish)[6] [6]: https://biblioteca.sistedes.es/
- Web of Science [1,2]: http://apps.webofknowledge.com

In addition, institutional and personal web pages are considered in Step 4:

- Webpages from every Spanish university (50 in total)
- SEBASENet website: http://www.uco.es/SEBASENet

Other sources like conference proceedings will be consulted when available online.

3. Selection of primary studies

Primary studies are selected from the set of candidate papers. Firstly, those papers that do satisfy the exclusion criteria are removed from the set. Secondly, only those papers satisfying the inclusion criteria are considered as primary studies.

3.1. Exclusion criteria

1. No Spanish institution was involved in the paper.
2. The study consists in a bachelor, master or PhD thesis. Similarly, contributions to doctoral symposiums and invited talks are excluded too.
3. The study is not properly indexed by scientific databases or its full content is not available online.
4. The study has to be written in Spanish or English.

3.2. Inclusion criteria

Papers should satisfy the following inclusion criteria in order to be considered as primary studies:

1. Only papers describing a SBSE approach providing detailed information about the search technique and problem formulation can be included, independently of the specific type of research article (survey, theoretical proposal, experimental study, etc.)
2. Books, book chapters and papers presenting tools can be included.
3. The study should be published between 2001 (date when the term SBSE was coined) and 28th February 2019 (date when the search was concluded) to be included.

4. Data extraction

On the set of primary studies, which have been properly saved and sorted in a dedicated, shared repository, the process of data extraction is applied. This is a key step for the correct development and analysis of the literature review under study, and will be performed by *data extractors* according to the *extraction strategy* defined below.

[6] SISTEDES (*Spanish Society of Software Engineer and Development Technologies*) provides open access to the electronic proceedings of the Spanish Conference on Software Engineering and Databases since 2012.

4.1. Data extractors

Each node acts as a *data extractor*, meaning that its members are equally responsible for the data extraction process. Each node can define its own internal procedure to gather the information and produce a final extraction. Common policies will rule the procedure to ensure correctness and consistency.

Any conflict of interest and disagreement is reported to the rest of data extractors.

4.2. Extraction strategy

Primary studies will be equally distributed among data extractors, checking that no paper will be assigned for review to a data extractor with any conflict of interest:

- A paper is not assigned to a data extractor who has (co-)authored the study.
- A paper is not assigned to a data extractor who is working for the same institution as some author of the study.

Online spreadsheets are developed and used in order to collect the outcomes of the extraction process. The information included by the “Data Extraction Form” is conformant to the classification scheme described in Section 4.3.

Each primary study is double-checked by the participants of a node. If disagreements appear or a doubt arises within the node regarding the extracted information from a primary study, the conflict will be resolved by a different data extractor, and formally noted in the extraction documents, namely the “Data Extraction Form”.

4.3. Classification scheme

This section details the classification defined for the data extraction process. This classification determines a number of categories that will be used to organise the "Data Extraction Form" to be filled in with information on every primary study.

The classification scheme defines four categories to provide information about:

1. Manuscript
2. Problem formulation (Software Engineering perspective)
3. Search perspective
4. Empirical validation and reproducibility

As detailed below, each of these items is properly sectioned into subcategories. Unless explicitly mentioned, only one value can be associated to a given subcategory. The abbreviation N/A is used under two circumstances: *(1)* the information is *Not Available*, or *(2)* the specific item is *Not Applicable* to the type of primary study under analysis, e.g. a survey.

After a first filtering of candidate papers, this classification scheme is also revisited.

4.3.1. Manuscript

For every primary study, the following information is gathered about the publication itself:

- **ID**. Every study will be identified by a unique reference that will be used to link the manuscript file, its data extraction form and any additional related information.
- **List of authors**, including their name and affiliation.
- **Title** of the manuscript.

- **Type of publication**: *Journal*[7], *International conference*, *Spanish conference*, *International workshop*, *Spanish workshop*, *Technical report*, *Book*, *Chapter*.
- **Name** of the publication/event.
- **Publisher** (volume, issue, pages).
- **Year** of publication.
- **Language**: *English*, *Spanish*.
- **Citation count** by Scopus.
- **Impact factor**.
    - When applicable, journal papers will be indexed following the Journal Citation Report (JCR) by Web of Knowledge in the form IF(year):value.
    - When applicable, conference papers will be indexed following the CORE[8] index in the form CORE(year):value.
    - The rest of papers will be reported as *N/A*.
- **Inter-organisational paper** (the paper is co-authored by researchers from different institutions): *Yes*, *No*.
- (If Inter-organisational is *Yes*) **Cooperation scope**: *International*, *Spanish*.
- **Threats to the validity**: *Yes* (threats are explicitly mentioned in the paper), *No*.
- **Methodology**: *Yes* (the method followed along the study is properly explained), *No*.

4.3.2. Problem formulation

The problem formulation analyses the paper from the software engineering perspective, putting the focus on the project phase and type of problem being addressed, as well as the importance of providing new tools in the area.

An initial taxonomy of SBSE problems is provided by Harman et al. (2012b), which has been extended for this review. Plus, categories have been completed with information extracted from (Harman et al., 2012) and the ACM Computing classification[9], apart from the experience of the authors in the SE field.

- **Software development stage**, which determines the specific project phase where the solution can be applied to.
    - Values: *Software planning*, *Requirements*, *Analysis and design*, *Code*, *Testing and validation*, *Integration and deployment*, *Maintenance*.
- **Type of problem**, which classifies the specific sort of software engineering problem being addressed in the study. Depending on the specific development stage, the following types of problems could be found:
    - Software planning:
        - *Cost estimation*
        - *Effort estimation*
        - *Project scheduling*
        - *Software quality measures*
    - Requirements:
        - *Elicitation*
        - *Prioritisation*
        - *Selection* (e.g. Next Release Problem)
    - Analysis and design:
        - *Object-oriented A&D*
        - *Architectural design*

[7] Terms in italics refer to literal string values for the field.
[8] http://portal.core.edu.au/conf-ranks/
[9] http://www.acm.org/about/class/ccs98-html#D.2

        - *Service-oriented architecture*
        - *Software product lines*
        - *User interfaces*
        - *Model-based design*
        - *Structured design*
    - Code:
        - *Genetic improvement*
        - *Design patterns*
    - Testing and validation:
        - *Model checking*
        - *Test case generation*
        - *Test case selection*
        - *Test case prioritisation*
        - *Test data generation*
    - Integration and deployment:
        - *Component allocation*
        - *Service selection*
        - *Web service composition*
        - *Software migration and portability*
    - Maintenance:
        - *Modularisation*
        - *Refactoring*
        - *Reverse engineering*
        - *Documentation*
        - *Software improvement*

- **Tool support**, which indicates if the authors are providing some tool to support the optimisation method: *Yes*, *No*.
    - URL: if it is available online, indicate its URL; otherwise, *N/A*.
    - If tool is available, **Type of tool**: *Prototype*, *Demo* (e.g. only a video or limited demo is available), **Release**.
    - If tool is available, **License** (e.g. GPL3, MIT, etc.).
    - If tool is available, **Documentation** like reports, user manuals or FAQs: *Yes*, *No*.
    - If tool is available, **Last update**: the date of the last release.
    - If tool is available, **Online user support**, e.g. forums.

  This information should be easily available from the tool website. Otherwise, the field would be marked as *N/A*.

4.3.3. Search perspective

The search perspective analyses the paper subject to the proposed search technique. From the primary study information about the type and nature of the conducted search, number of objectives and algorithm will be extracted (Boussäid et al., 2013; Blum et al., 2003). For a given category, if a value to be assigned to the primary study has subcategories, then a compound term will be used in the form "*value* > *subvalue_1* > .. > *subvalue_n*".

- **Type of search goal**, which determines if it is oriented towards finding local solutions or global optimum solutions:
    - *Local search* (e.g. Hill climbing, etc.)
    - *Global search* (e.g. –but not limited to– GRASP, Tabu search, Simulated annealing, Evolutionary computation, Particle Swarm Optimisation, etc.)

- **Type of search problem**, which refers to the number of objectives used to evaluate the quality of solutions.
    - Values: *single-objective*, *multi-objective*.
- **Type of search algorithm**, which categorises the approach being applied and its nature, when required. The allowed types of search approaches are:
    - *Exact search*[10].
    - *Metaheuristic algorithm*, which can be categorised depending on the numbers of solutions processed in one iteration:
        - *Single-solution*-based metaheuristics.
        - *Population-solution*-based metaheuristics.
            - *Evolutionary computation*:
                - *Genetic algorithm*
                - *Evolution strategy*
                - *Evolutionary programming*
                - *Genetic programming*
                - *Other* (e.g., estimation of distribution algorithms, differential evolution, path relinking, etc.)
            - *Swarm intelligence:*
                - *Ant colony optimisation*
                - *Particle swarm optimisation*
                - *Bee colony optimisation*
                - *Other* (e.g. bacterial foraging, artificial immune systems, cuckoo search, etc.)
- **Type of search model**, which categorises the way the search process is conducted (multiple values are allowed for this category):
    - *Standalone*, i.e., one single algorithm is applied to resolve the problem.
    - *Hybrid algorithm*, i.e., an approach combining different search techniques like memetic algorithms.
    - *Parallel algorithm*, i.e., an approach simultaneously running multiple search procedures in parallel or under a distributed schema.
    - *Interactive algorithm*, i.e., an approach integrating somehow the user's opinion.
    - *Other* approaches like coevolution, hyperheuristics, etc.
- **Algorithm(s)**, when applicable, it indicates the algorithm(s) used for the search process (e.g. *SPEA2*, *NSGA-II*, *ACS*, etc.). This is a multi-value that intends to explain the most recurrent techniques followed in the field. If any algorithm was proposed by the authors in the same paper, then "*Custom*" will be designated. If there is no information available about the applied algorithms, then *N/A*.

4.3.4. Empirical validation and reproducibility

Given the importance of making explicit a correct and precise validation of the proposal, as well as showing a reproducible method, it is worth analysing how the proposal is empirically validated and the availability of benchmarks.

- **Number of benchmarks**, which requires a number or *N/A*, if not mentioned.
- **Type of benchmarks**, which specifies the nature of the source benchmarks (one or multiple values from the following list):

[10] "Exact search" is used as general term enclosing both exhaustive search and approximation algorithms, including some heuristics.

    - *Synthetic*, i.e. a randomly generated problem instance.
    - *Academic*, i.e. the benchmark consists in a case study or small instance which is exclusively used in the research field.
    - *Industrial*, i.e. the benchmark has been adopted or extracted from a real-world environment.
- **Source code** available (*Yes, No*), meaning that the proposal is reproducible.
- **Statistical tests** (Derrac et al., 2011; Arcuri and Briand, 2014), where the categorization and name of the conducted test is specified. Multiple choices are possible. The value will have the form *cat*>*subcat1*>…*subcat_n:name* (e.g., "*Pairwise* > *Non-parametric* > *Wilcoxon*"):
    - *Pairwise* comparison
        - *Parametric* tests (e.g. Student's t-test)
        - *Non-parametric* test (e.g. Wilcoxon rank-sum test)
    - *Multiple* comparison
        - *Parametric* test (e.g. ANOVA)
        - *Non-parametric* test (e.g. Friedman)
    - *Effect size* measurement
        - *Parametric* test
        - *Non-parametric* test (e.g. Cliff's Delta)

References

[Arcuri and Briand, 2014] A. Arcuri and Lionel Briand. "A Hitchhiker's guide to statistical tests for assessing randomized algorithms in software engineering". Software Testing, Verification and Reliability, vol. 24, pp. 219-250. 2014.

[Blum et al., 2003] C. Blum and A. Roli. "Metaheuristics in combinatorial optimization: Overview and conceptual comparison". ACM Computing Surveys, vol. 35, num. 3, pp. 268–308. 2003.

[Boussaïd et al., 2013] I. Boussaïd, J. Lepagnot, P. Siarry. "A survey on optimization metaheuristics". Information Sciences, vol. 237, pp. 82-117. 2013.

[Derract et al., 2011] J. Derrac, S. García, D. Molina, F. Herrera. "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms". Swarm and Evolutionary Computation, vol. 1, pp. 3-18. 2011.

[Harman et al., 2012] M. Harman, S. A. Mansouri, Y. Zhang. "Search Based Software Engineering: Trends, Techniques and Applications". ACM Computing Surveys, vol. 45(1), Article No. 11. 2012.

[Harman et al., 2012b] M. Harman, P. McMinn, J. T. de Souza, S. Yoo. "Search Based Software Engineering: Techniques, Taxonomy, Tutorial". Empirical Software Engineering and Verification, vol. 7007 Lectures Notes in Computer Science, pp. 1-59. 2012.