A FEDERATED SEARCH APPROACH TO FACILITATE SYSTEMATIC LITERATURE REVIEW IN SOFTWARE ENGINEERING

Mohammad Ghafari[1], Mortaza Saleh[2], Touraj Ebrahimi[3]

[1,2]Dep. of Computer Engineering and Information Technology, Payame Noor University, Tehran, Iran

m.ghafari@pnu.ac.ir, saleh.mortaza@gmail.com

[3]DP Co., Ltd. Development Research Group, Tehran, Iran

ebrahimi@dpco.net

ABSTRACT

To impact industry, researchers developing technologies in academia need to provide tangible evidence of the advantages of using them. Nowadays, Systematic Literature Review (SLR) has become a prominent methodology in evidence-based researches. Although adopting SLR in software engineering does not go far in practice, it has been resulted in valuable researches and is going to be more common. However, digital libraries and scientific databases as the best research resources do not provide enough mechanism for SLRs especially in software engineering. On the other hand, any loss of data may change the SLR results and leads to research bias. Accordingly, the search process and evidence collection in SLR is a critical point. This paper provides some tips to enhance the SLR process. The main contribution of this work is presenting a federated search tool which provides an automatic integrated search mechanism in well-known Software Engineering databases. Results of case study show that this approach not only reduces required time to do SLR and facilitate its search process, but also improves its reliability and results in the increasing trend to use SLRs.

KEYWORDS

Evidence Based Research, Systematic Literature Review, Federated search, Software Engineering

1. INTRODUCTION

It is important to summarize the existing evidence about a topic in order to identify research gaps. In this way, the researcher can evaluate available evidences reported in literature and finally suggest available areas for further investigations. Besides, to impact industry, researchers developing technologies in academia need to provide tangible evidence of the advantages of using them. A Systematic Literature Review (SLR) also referred as Systematic Review (SR), is widely used as a research methodology in medical research since 1990s and has been spread to other sciences where need to be as unbiased as possible by being auditable and repeatable. It provides high quality research evidence relevant to a particular research question, topic area, or

phenomenon of interest. It also has been considered as one of the key research methodologies of Evidence-Based Software Engineering (EBSE) since Kitchenham, Dyba and Jorgensen's seminal paper on EBSE published in 2004[1-3]. Todays, although a broad number of SLRs related to Software Engineering (SE) have been conducted and reported by SE researchers, it is still a new topic to the SE community and has many challenges. SLR involves three main steps as illustrated in Fig. 1:

1. **Planning the review:** Firstly, the need for the review is identified. Then, the research questions are specified, and the review protocol is defined.

2. **Conducting the review:** After identifying the plan, the primary studies are selected. Next, the quality of these studies is assessed to decide about those which should be either excluded or included from study. Then, the data extraction and monitoring are performed. Finally the extracted data are synthesized.

3. **Reporting the review:** At the end, the dissemination mechanisms are specified, and the review report is presented.

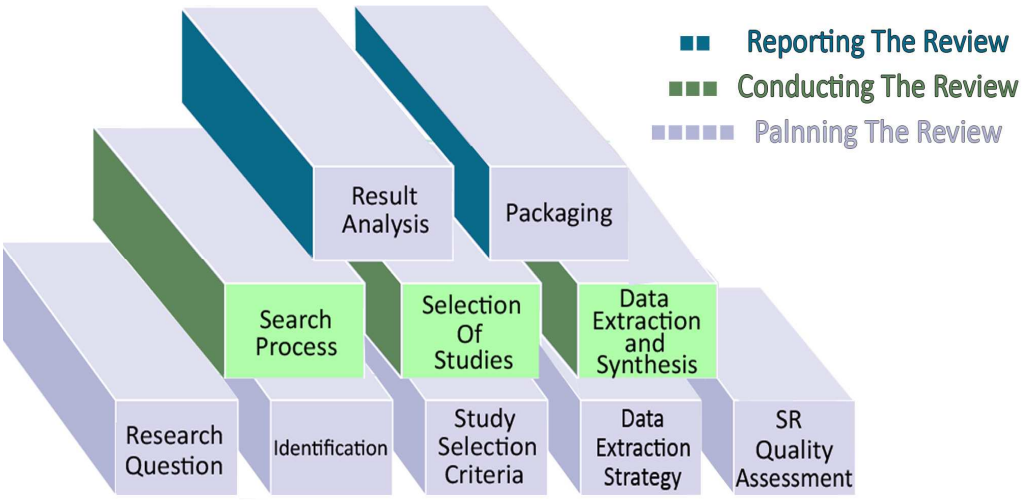

Figure 1. The SLR steps

The main advantage of SLR is that its findings are more reliable. It reviews the background of a topic and attempts to build its body of knowledge. Therefore, it minimizes the level of bias that can be prevalent in Traditional (ad-hoc) Literature Reviews (TLRs). In contrast, one obvious disadvantage (weakness) of SLR is that it requires considerably more effort than TLRs. Although, several researchers have been working on improving the scientific and technological support for SLRs in SE, there is a big consensus that many of SLR challenges come from SR-incompatible nature of SE libraries as the main resources used in SE researches. To the best of our knowledge, no concrete solution has been addressed this problem. This paper provides some tips to do SLR steps in a better way. Also, as its main contribution, it proposes a tool for empowering the SLR search process.

The structure of this paper is organized as follow. Section 2 provides a more detailed review on SLR, its steps, and introduces some solutions to mitigate its difficulties in SE. Next, section 3 focuses on SLR search process concerns and presents a federated search engine which facilitates

this process. A case study is presented in section 4. Related works are reviewed in section 5.The conclusion and future works close this paper.

2. SLR OVERVIEW

SLRs are a means of aggregating knowledge about a SE topic or research question [4-7]. SLRs are referred to as secondary studies and the studies they analyze are referred to as primary studies. The main goal of SLR is to be as unbiased as possible and finally provide more reliable results. In spite of the growing importance that systematic review has been achieving in the past years, it is still a new topic to the SE community and has many challenges. We believe that our findings during the conduct SLR can usefully identify some more general issues that might help other researchers to avoid repeating the mistakes of others. In following, the SLR steps as described by Kitchenham and Charters [7] are briefly overviewed. Also, some tips which increase its quality and precision are presented:

2.1 THE SEARCH PROCESS

In this step, a problem or need for information is converted into one or more research questions. Then some search strings are identified based on these questions. Here, the aim is to find as many information as possible that is related to the research questions using an appropriate search strategy. It is important to consider available semantic synonyms in research area when the search strings are devised. Although some libraries consider synonyms in their search mechanism, they are too limited and in many cases are not as professional as required. Therefore, it is highly recommended to consult with an expert to include all related synonyms in your search strings. Finally, to ensure that no relevant paper will be missed, the search strings should be tested against some known relevant publications to ensure that the studies are returned in results. Now, it is time to search in well-known databases based on paper title, keywords, and abstract. Taking the advantages of reference management software e.g. Zotero in this step is strongly recommended. Using this tools all papers bibliographic data e.g. title, authors, abstract, and etc. are collected into a database.

While the search process is repetitive, it could be time-consuming, boring, and consequently error prone. Also, in some cases databases have different features. Therefore, it is important to researchers to be as familiar as required with databases search mechanism. A comparison of databases capabilities is presented in section 3.

2.2 THE STUDY SELECTION

Study selection identifies the primary studies that give direct evidences related to the research questions. At first, all studies that are obviously irrelevant, or duplicates are excluded. When several duplicated articles of a study exist in different versions that appear as books, journal papers, conference and workshop papers, it is better to include only the most complete version of the study and exclude the others. Unfortunately, in some cases duplicated papers have not same titles, so it is not obvious to identify duplications. In such scenarios the advantages of using reference management software can be helpful to identify these papers. For example, since duplicated papers have same authors, a query can be executed on Zotero database based on the paper authors. Next, to examine papers more carefully, a set of papers of the selection process will be read starting from the title, abstract and if necessary, introduction and conclusion. Finally, an additional reference/citations scanning and analysis is recommended in order to find out whether any paper is missed. This guarantee that a representative set of studies are selected.

In cases which many papers are included, it is very practical to rank them based on their relevance to research questions during scanning. As a result the reviewers could start the review process based on the high ranked papers which are more important.

2.3 THE DATA EXTRACTION AND SYNTHESIS

The most time-consuming step of SLR is data extraction where all included papers should be read entirely. This process should be done based on a template which is provided by domain experts to make sure that all the extracted information will be related to the research questions the review intends answering. All extracted data should be rigorously documented in this template which in many cases can be a spreadsheet. The advantage of using this template is twofold. Firstly, it handles the direction of research and helps reviewers to find interested topics to research questions more carefully. Next, it can be an appropriate source to train reviewers, especially in cases which they have not enough knowledge around research questions. In order to minimize the potential bias during the review process, peer-review is the most common method used by systematic reviewers. In addition, it is strictly recommended that the supervisor checks the reviewers' activities and feedback to them in order to resolve their mistakes. In such evolutionary way, after a while, the reviewers' ambiguities around the research topic will be reduced and they will be mature enough in topics related to research.

2.4 THE DATA ANALYSIS

Finally, extracted data should be studied to provide the results. In this step, data analysis tools e.g. NVivo can be used. These tools provide deep levels of analysis and present some interesting classifications, relationships, etc. which help researchers to make better decisions.

3. FEDERATED SEARCH APPROACH

3.1 MOTIVATION

An important step of SLR is the search of evidence for answering the research question. The more evidence found, the more support to rational decision making is assured. However, an incorrect or incomplete search may consequent missing some evidences, and accordingly wasting time to research on something which has been addressed earlier. Motivated by this, in order to avoid bias in research, using a complete and precise method to find available evidences around the research topic is inevitable. In following, two main issues exist in search process is reviewed:

1. The databases search results are directly dependent on provided search strings and there should be enough care when search strings are devised. Also, some databases e.g. ACM do not support combination of fields, and in order to specify multiple fields within a search string, the command-based search should be used. As a result, people should be familiar with the structure of search string in each database separately. Moreover, while some databases do not automatically include synonyms in their search strings, others provide pretty weak automatic stem variations in searching. Accordingly, synonyms, related terms, and alternative spelling should be included manually.

2. Since each database provides its own results and none of databases has access to search within the resources provided by others, the search process should be repeated within all databases separately. Besides, while each database has its own search string structure, the

search string provided for e.g. ACM cannot be used in SpringerLink. Consequently, all search strings should be constructed manually and be checked within all desired databases one by one.

Motivated by these issues, the search process will be tedious and strongly error prone. Since any loss of data may lead to bias in research, it is too critical to provide a more reliable search process. Despite of broad number of SLRs have been conducted and reported in SE, this community still feels lack of a mechanism which mitigates these issues. In order to provide an integrated search across multiple databases, a mechanism is required which map the search string provided by the user to all desired databases. Also, some automatic enhancements provided by domain experts on the search string proposed by the user, increases the reliability of the search results. In the rest of current section, a federated search approach is proposed to ease these kinds of problems during search process in SLR.

3.2 FEDERATED APPROACH

In order to provide a federated search, a single point is required which direct the user to all his desired databases. Accordingly, we have undertaken a broad comparison between well-known databases to identify their common features and the field codes which are used to devise the search strings. Despite of commonality between databases, there are some advantages in some of them i.e. in "ScienceDirect" it is possible to search the figures, tables and videos separately, in both "IEEEXplore" and "ISI Web of Knowledge", the accented characters are considered automatically.

Accordingly, there is no need to manually search for British and American spellings of any keywords, In "GoogleScholar" and "Wiley", synonyms of any term can be included automatically by use the tilde sign immediately before the search term. As a result, a search now matches both singular and plural forms of the search term e.g. a search for 'foot' will match records containing the word 'foot' and records containing the word 'feet'. Also, there are some weak points in some databases. For example, "SpringerLink" has limitation in its command length, consequently whenever there is a long search string; it should be separated to more than one search string, in "GoogleScholar" searching fields are too limited. For Example it doesn't support searching the abstract, title, and keywords separately. Moreover, "CiteSeerx" database does not support refinement. The comparison is summarized in Table 1.

In order to clarify the table, some definitions about its fields are provided in following:

1- Connectors: These operators can be used to specify the words that we want to include or exclude from our search results and to search for more than one word in a single search:

- AND: This operator indicate that all of the terms in our search must appear in the returned documents, even if the terms are far apart from each other.
- OR: When at least one of our search terms must appear in returned documents, this operator can be used. To search for synonyms, alternate spellings, or abbreviations this operator is so helpful.
- NOT: Exclusion a word from the search should be manipulated by this operator.

Table 1. Comparison between well-known databases

Features / Database	Wildcard			Find by				
	*	%	?	Title	Abstract	Keywords	Title & Abstract	Full-Text
ACM	×	✓	×	Title:Y	Abstract:Y	Keywords:Y	×	×
ScienceDirect	✓	×	✓	Title(Y)	Abstract(Y)	Keywords(Y)	×	×
IEEE	✓	×	×	Document Title:Y	Abstract:Y	Author Keywords:Y	×	Y
Springer	×	×	×	Ti:(Y)	Ab:(Y)	×	Ab:(Y)	Y
Compendex	✓	×	✓	(Y) MN Ti	(Y) WN Ab	×	(Y) MN Ky	(Y) WN All
ISI Web of Knowledge	✓	×	✓	Ti=Y	×	×	Ts=Y	Ts=Y
SCOPUS	✓	×	✓	Srctitle(Y)	Abs(Y)	Key(Y)	Title-Abs-Key(Y)	All(Y)
CiteSeerx	✓	×	×	Title:Y	Abstract:Y	Keyword:Y	Title:Y and Abstract:Y	Text:Y

2- Wildcards: To search for words that have spelling variations or contain a specified pattern of characters, these operators can be used to represent the variations:

- ***:** In order to replace multiple characters anywhere in a word this wildcard is used. E.g. behav* finds behave, behavior, behaviour, behavioural, behaviourism, etc.
- **%:** This wildcard is used to search for different endings of a term. E.g. network% will find network, networks, networking, networked. The % cannot be used with the exact phrase option.
- **?:** In order to replace any single character anywhere in a word, one question mark is used for each character required to be replace. E.g. analy?e finds analyse or analyze.

3- Stemming: This feature allows including various extensions or derivatives of keywords in search strings. It doesn't require special characters or commands. It also seems to automatically un-stem the keywords.

No.	Database	Search String
1	ACM	(Abstract:Architectural and Abstract:Constraint) and (Title:Evolution or Title:Change)
2	IEEE	(Abstract:Architectural Constraints) and (Document Title:Evolution or Document Title:Change)
3	SceinceDirect	Abstract(Architectural Constraints) and Title(Evolution or Change)
4	Compendex	((((Architectural Constraints) WN Ab) and ((Evolution) WN Ti)) or ((Change) WN Ti))

Table 2. Search string comparison between well-known databases

4- Find By: In the columns which are tagged by "Find By", the notation which is used by each database for building its search strings is presented. For example, consider that it is required to search for the papers which their abstracts contain "architectural constraints" and their title include one of the words e.g. "evolution" or "change". According to the field codes which are used in each database, each of them provides its own search string as depicted in table 2.

In order to construct a search string which is compatible within each database, understanding the structure of search string is required. As it is shown in Table 2, the search strings devised for one database, cannot be used by others without extra refinement. Accordingly, we have devised some rules which are used to map a search strings to its corresponding search strings in the target databases.

An abstract view of our federated search tool is presented in Figure 2. As depicted above, it has five main parts named: Admin Panel, Search Panel, Query Generator, Crawler, and an Engine which act as a control unit.

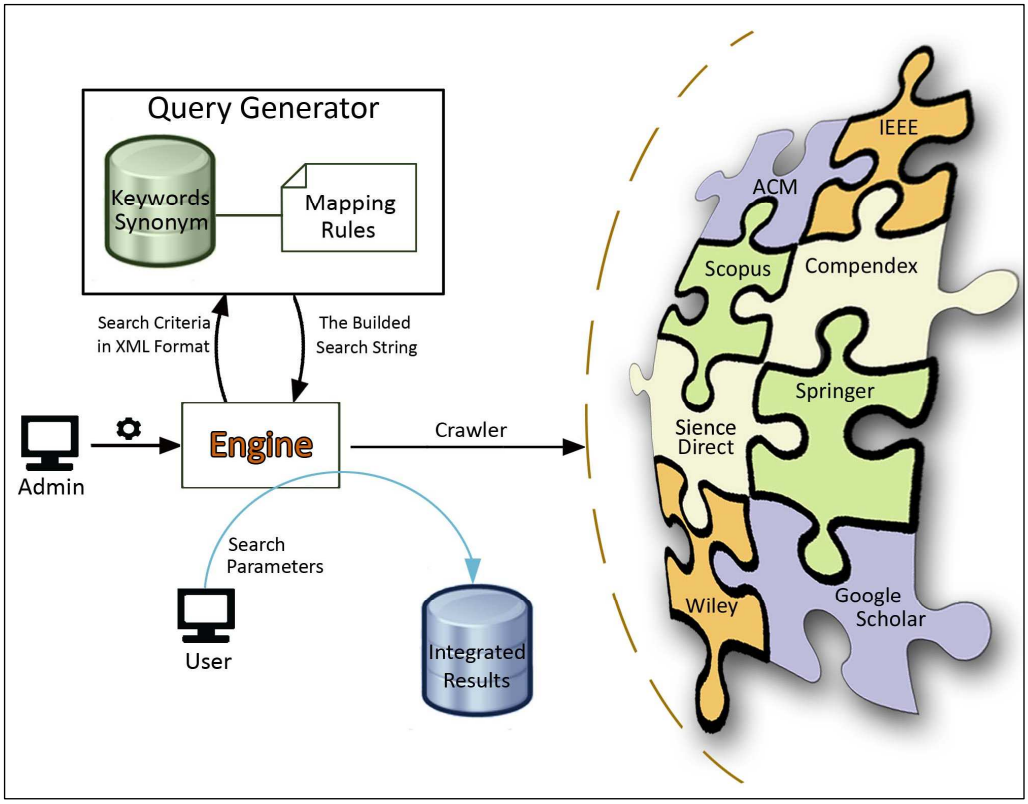

Figure 2. Our proposed model

1. **Admin Panel:** The system administrator can define some research categories and sub-categories which will be managed by their own domain experts. These people are responsible to define available research keywords and their corresponding synonyms for each category. They also direct the search process to prevent the probable biases may exist in students research process. For example, in software reconfiguration which is a sub-category of software evolution, the domain expert defines some keywords e.g. runtime, change, restructuring as synonyms of evolution, to be automatically included in searches. These keywords are stored in a database called “Keywords Synonym”.

2. **Search Panel:** This is an interface which provides advanced search in order to search within particular databases. It offers some search facilities such as the ability to search via a restricted set of field codes e.g. Full Text, Title, Abstract, and Date as well the ability to combine search phrases using boolean operator. Once the search criteria are specified by the user, they are sent to Query Generator.

3. **Query Generator:** This part is responsible to generate the search strings which are compatible to databases selected by the user. In order to build queries, it uses an xml file which contains each database code fields called rules. These rules are used to map a search string to its corresponding structure within target database. Here, regarding the search context, available synonyms to each keyword are considered too. Finally, these queries are prepared to being posted to the target databases i.e. ACM, IEEEXplore, and etc.

4. Crawler: As soon as the queries are prepared, they are sent to the target databases. Then, a crawler starts downloading the search results provided by each databases. Finally, all data are gathered into a single database.

4. CASE STUDY

To evaluate this study, a web-based tool has been developed using Python and a document based database called MongoDB. This tool has been used at Computer Engineering department of PNU to help master students to gather their required evidences for their thesis. At first step, it searches within three well-known databases: ACM, IEEEXplore, and ScienceDirect. Also, their thesis supervisors were responsible to feed the synonym database which is used by Synonym Generator. This tool proposes an interface which is similar to ACM advanced search page. Once the user specifies its search criteria, it is stored in an XML document as follows:

```
<SearchString>
    <abstract connector="main">Architectural</abstract>
    <abstract connector="and">Constraint</abstract>

    <title connector="main">Evolution</title>
    <title connector="or">Change</title>
</SearchString>
```

Figure 3. The search criteria

The XML document which is depicted in Fig. 3, demonstrates the search criteria which is specified by the user by the means of provided Search GUI. The structure which has been shown in Fig. 3 indicates a query that can be used in order to find the papers which their abstracts include both *Architectural* and *Constraint*; also their titles contain *Evolution* or *Change*.

```
<Rules>
    <ACM>
        <title value="Title:X" />
        <abstarct value="Abstract:X" />
    </ACM>
    <Springer>
        <title value="Ti:(X)" />
        <abstarct value="Ab:(X)" />
    </Springer>
    <Compendex>
        <title value="(X) WN Ti" />
        <abstarct value="(X) WN Ab" />
    </Compendex>
</Rules>
```

Figure 4. The code fields in databases

In order to generate the actual search strings, each database filed codes are stored as a rule in an XML file named *rules.xml* depicted in Fig. 4. This file is used to map the search criteria to its corresponding values in each database. In a nutshell, the mentioned rules build the skeleton of each database search strings which can be used by the QueryGenerator. The X character works as a place holder that can be replaced in each rule with its counterpart value specified by the user. If any synonym is available, it will be added to the keywords inputted by the user. Finally, the

connectors (e.g. AND, OR) are used for connecting the entire structure of search strings. Afterwards, the prepared search strings are ready to be used in search process of relevant databases. Accordingly, these queries would be sent to target databases using HTTP GET method.

In order to integrate the results provided by various databases, we have developed a snippet of code which acts as a simple crawler. This crawler is configured manually based on the HTML tags and CSS classes which are used in each database. Accordingly, it extracts the title, abstract, and the link of each paper. Finally, the extracted data are collected into a database.

5. RELATED WORKS

Although SLR has been used in many different SE researches, little contributions have been proposed on adopting SLR in SE. Most SLRs in SE have followed guidelines derived from those used by medical researchers, adapted and applied by Kitchenham and her colleagues [1-3] to reflect the specific problems of SE research. In fact these two papers are the corner stone of SLR in SE. However, some other works around SLR have been done in SE which some of them are introduced in following.

In a work done by Zhang and Babar [8], use of SLRs and its adoption in SE empirically investigated from various perspectives. They used multi-method approach as it is based on a combination of complementary research methods which are expected to compensate each other's limitations. Also in another paper [9] they started an empirical research program that aims to contribute to the growing body of knowledge about SR in SE. In [10], Budgen and his colleagues proposed a cross-domain investigation of empirical practices. The objective of this study is to investigate how other academic disciplines use evidence-based practices in order to help assess the guidelines that the authors have developed for conducting SLR in SE. MacDonell and his colleagues [11] has assessed the reliability of SLR in empirical SE. This paper investigated the consistency of process and the stability of outcomes. The [12] analyzed the quality, coverage of SE topics, and potential impact of published SLRs for education and practice in a special period. Dyba and Dingsoyr [13] assessed the strength of Evidence in SR in SE. They present an overview of some of the most influential systems for assessing the quality of individual primary studies and for grading the overall strength of a body of evidence. Kitchenham and her colleagues [14], provide a comparison about the use of targeted manual searches with broad automated searches. Their study also aims to assess the importance of grey literature and breadth of search on the outcomes of SLRs. In [15] an Evidence-Based Understanding of Search Engines in SLRs is presented. It proposes an initial set of metrics for characterizing the EDS from the perspective of the needs of secondary studies.

6. CONCLUSION AND FUTURE WORKS

Nowadays, a large majority of the researchers are convinced of the value of using a rigorous and systematic methodology for literature reviews. Since the search process and collecting evidences is a critical point to prevent any bias in research, it is important to provide a mechanism which precise this process as much as possible. In this paper, a federated search approach to facilitate SLR search process is presented. It bridges the gap between the spread of databases in SE and integrated search required by SLR. Finally, a partially-automated tool is developed which has been practically used by master students to search for required evidences for their thesis. The results have verified the expected contributions include:

1- It considerably reduces required time as one of the most concerns in SLR. It also improves the search process by including synonyms which are provided by an expert domain, automating the search process rather than manually search in every database for every search criteria, and finally integrating multiple databases search results.

2- Its crawler-enabled feature, facilitate search process and automatically save results in a database. After doing some researches, this database will contain thousands of records which not only could be used locally, but also would be so beneficial as a knowledge base for ongoing researches.

3- It facilitates both the qualitative or quantitative analysis on search results while they are integrated in a database. For example, classifying results based on their meta-data fields e.g. authors, may help the researcher to identify duplicated papers.

As our ongoing work, we are working on developing a full automated tool which is crawler enabled intrinsically and just require a XML file containing the target databases field codes. Also, while some databases do not provide URL-based search mechanism, they cannot be included in our tool. As a result we plan to enable this tool to support the databases which provides search API for federate search.

REFERENCES

[1] Kitchenham B.A., Dyba T., Jorgensen M.; "Evidence-based software engineering", In: 26th International Conference on Software Engineering, pp. 273–281, 2004.

[2] Dyba T., Kitchenham B.A., Jorgensen M.; "Evidence-based software engineering for practitioners", IEEE Software, pp. 183-186, 2005.

[3] Jorgensen M., Dyba T., Kitchenham B.; "Teaching evidence-based software engineering to university students", In: 11th IEEE International Software Metrics Symposium, 2005.

[4] Fink A.; "Conducting research literature reviews: from the internet to paper", SAGE Publication, 2005.

[5] Petticrew M., Roberts H.; "Systematic reviews in the social sciences: a practical guide", Blackwell Publication, 2006.

[6] Kitchenham B.; "Procedures for undertaking systematic reviews, joint technical report", Computer Science Department, Keele University (TR/SE-0401) and National ICT Australia Ltd (0400011T.1), 2004.

[7] Kitchenham B., Charters S.; "Guidelines for performing systematic literature reviews in software engineering technical report", Software Engineering Group, EBSE Technical Report, Keele University and Department of Computer Science University of Durham Vol. 2, 2007.

[8] He Z., Babar M.A.; "An empirical investigation of systematic reviews in software engineering," International Symposium on Empirical Software Engineering and Measurement (ESEM), pp. 87-96, 2011.

[9] Babar M.A., Zhang H.; "Systematic literature reviews in software engineering: preliminary results from interviews with researchers", In 3th International Symposium on Empirical Software Engineering and Measurement, pp. 346–355, 2009.

[10] Budgen D., Bailey J., Turner M., Kitchenham B., Brereton P.,Charters S.; "Cross-domain investigation of empirical practices", Institution of Engineering and Technology, Vol. 3, pp. 410-421, 2009.

[11] MacDonell S., Shepperd M., Kitchenham B., Mendes E.; "How reliable are systematic reviews in empirical software engineering?", IEEE Transactions on Software Engineering, Vol. 38, pp. 676-687, 2009.

[12] Silva F., Santos A., Soares S., França A., Monteiro C., Maciel F.; "Six years of systematic literature reviews in software engineering: An updated tertiary study", Information and Software Technology, Vol. 53, pp. 899–913, 2011.

[13] T. Dyba, T. Dingsoyr, "Strength of evidence in systematic reviews in software engineering", Proceedings of the Second ACM-IEEE international symposium on Empirical software engineering and measurement, pp. 178-187, 2008.

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

[15] Chen L., Babar M., Zhang H.; "Towards an evidence-based understanding of search engines in systematic literature reviews", In Proceedings of 14th International Conference on Evaluation and Assessment in Software Engineering, pp. 1-4, 2010.

Authors

M. Ghafari received the master's degree in Software Engineering from Payame Noor University in 2012. The focus of his research is in the field of software evolution in general and dynamic software reconfiguration in particular. His other fields of interest include component-based software engineering, software architecture, and distributed systems. The work described in this paper is part of his research process in his master thesis.

M. Saleh received his B.Sc. degree in Computer Engineering from Payame Noor University, Iran in 2008. Currently he is pursuing his master degree in the Software Engineering in Payame Noor University, Tehran, Iran. His interests include Service Oriented Architecture, Information Systems, Security, and Coordination Problems.

T. Ebrahimi graduated in software engineering in 2002. He received the bachelor degree from Azad University, South branch, Tehran, Iran. His major research interests include Artificial Intelligence in general and Neural Networks, Fuzzy Logic, and Genetic Algorithms in particular. He is a proficient developer with more than 10 years' experience, especially in developing Network Management Systems.