The state of the art on design patterns: a systematic mapping of the literature

B. Bafandeh Mayvan , A. Rasoolzadegan , Z. Ghavidel Yazdi

PII: S0164-1212(16)30232-1
DOI: 10.1016/j.jss.2016.11.030
Reference: JSS 9882

To appear in: *The Journal of Systems & Software*

Received date: 22 June 2016
Revised date: 17 November 2016
Accepted date: 18 November 2016

Please cite this article as: B. Bafandeh Mayvan , A. Rasoolzadegan , Z. Ghavidel Yazdi , The state of the art on design patterns: a systematic mapping of the literature, *The Journal of Systems & Software* (2016), doi: 10.1016/j.jss.2016.11.030

Highlights

- A comprehensive SMS on the design patterns to identify the related topics and detect trends and gaps.
- There are six main topics in the field of design patterns.
- Developing and Mining are the most active topics.
- Describe the demographics of design patterns research.
- Quantify the research load on the design patterns topics.

The state of the art on design patterns: a systematic mapping of the literature

B. Bafandeh Mayvan[a], A. Rasoolzadegan*[,a], Z. Ghavidel Yazdi[b]

[a]Faculty of Engineering, Ferdowsi University of Mashhad, Mashhad, Iran
[b]E-Learning Center, Ferdowsi University of Mashhad, Mashhad, Iran

*Corresponding author: E-mail: rasoolzadegan@um.ac.ir

Abstract— Design patterns are widely used by software developers to build complex systems. Hence, they have been investigated by many researchers in recent decades. This leads to the emergence of various topics in the design patterns field. The objective of this paper is to present an overview of the research efforts on design patterns for those researchers who seek to enter this area. The main contributions are as follows: (a) identifying research topics in design patterns, (b) quantifying the research emphasis on each topic, and (c) describing the demographics of design patterns research. The last secondary study with similar goals in the design patterns field considers the Gang of Four design patterns only. However, the scope of the current study is all of the design patterns. Moreover, our review covers about six additional years and a larger number of publications and venues. In this systematic mapping study, a total of 2775 papers were identified as relevant, and 637 of them were included. According to the results, design patterns can be classified into six different research topics. As a consequence, it is concluded that Pattern Development, Pattern Mining, and Pattern Usage are the most active topics in the field of design patterns.

Keywords

Design patterns, Systematic mapping study, Systematic review

1 INTRODUCTION

Software patterns are solutions to recurring software problems. These solutions are the result of the experiences of experts and can be used over and over again (Gama et al. 1995). Software patterns can be categorized into different classes according to the software development phase. Implementation patterns, testing patterns, analysis patterns, and design patterns (DPs) are some examples of these classes (Hamza 2004).

DPs, such as Gang of Four (GoF), and security DPs, are an active field of research in software engineering (Zhang and Budgen 2012; Zhang and Budgen 2013; Ampatzoglou et al. 2013; Riaz et al. 2015). In recent decades, many articles have been published on DPs topics which contribute to enriching the literature of the field. Hence, performing a comprehensive review and identifying the related themes and detecting trends and gaps in each of them is crucial. In other words, we need a reference guide so that we can conduct an effective investigation into each of the DPs topics and take significant steps to improve and enhance them.

To review the studies carried out in a broad area such as DPs, an advanced search method should be applied. This method should be able to cover this vast area and at the same time it must be reliable, repeatable, impartial, rigorous, and the results should be traceable. Evidence-based software engineering (Dyba et al. 2005) has provided this method. Two well-known methodologies that are provided by evidence-based software engineering are systematic literature review (SLR) and systematic mapping study (SMS) (Kitchenham et al. 2011; Petersen et al. 2015). There are some differences between SLR and SMS (Kitchenham et al. 2010; Kitchenham et al. 2011; Petersen et al. 2015). In general, methodologies for searching and extracting data in SMS and SLR are alike, but data analysis approaches are different, and research questions in SMS are more general than SLR (Kitchenham et al. 2011; Petersen et al. 2015). This is because of the differences between the aim of SMS and SLR. The purpose of SMS is identifying research trend and classifying topics that exist in a research area while SLR tries to find the primary studies and subsequently, extract data from them to answer a specific research question (Kitchenham et al. 2011). When we intend to identify the scope of a topic and classify the studies that exist in this scope, we should use the mapping study method. Therefore, SMS can provide general information on one topic, and it can be done as a pre-review, before SLR.

In the field of DPs, some secondary studies have reviewed the literature. Among these studies, the only similar work to ours is a systematic mapping study that published in 2013 (Ampatzoglou et al. 2013). However, the scope of our study is all the DPs that exist in the literature, while Ampatzoglou et al. (2013) only considered GoF DPs. Moreover, our paper covers about six additional years, and a larger number of publications and venues. The results show that more than 30% of articles have been carried out from 2010 onwards. These differences make this work substantially different and worth investigation. In the Related Works section, the comparison between our work and other secondary studies in the field has been conducted in detail.

In this paper, an SMS on the literature of DPs has been carried out. This study contributes into providing a taxonomy of the research topics related to DPs as well as a breakdown of papers in each topic. Moreover, the researchers, countries, and publication venues that are active in research related to DPs are identified. Determining the taxonomy and demographics of a particular research topic can provide a useful starting point for other researchers. Therefore, using results that are extracted from this study, we want to present a reference guide for those researchers who seek to enter the DPs field and need to identify research trends and gaps.

The rest of the paper is organized as follows: related works is presented in Section 2. In Section 3, the systematic mapping technique is described. In Section 4, according to the research questions, the results of the systematic map are presented and subsequently in Section 5, the results are discussed. In Section 6, the threats to validity are discussed. Finally, in Section 7, conclusions and future works are presented. Note, we have prepared a document called DPs-SM (Design Patterns-Supplementary Material) to present the details of our SMS. This supplementary material is available on the web[1]. Throughout this paper, whenever it is needed, we will refer to this document.

2 Related works

Searching on the DPs field, we found a total of four secondary studies that use a systematic method to review the literature (see Table 1). Secondary study replications are valuable because, whenever new evidence is published after the completion of a secondary study, we should update it (Felizardo et al. 2016). In Table 2 we made a comparison between our paper and the other secondary studies in the field. As seen, among these studies, the only work with the same goals is the work of Ampatzoglou et al. (2013). According to Ampatzoglou et al. (2013), topics that exist in the field of DPs are identified and classified. The context of our study is all the DPs that exist in the literature, while Ampatzoglou et al. (2013) only considered GoF DPs. As illustrated in Table 2, our paper compared to other existing works, covers a larger number of publications and venues. In this paper, we present complete results and analysis to those researchers who seek to enter the field of DPs. Hence, as mentioned before, the result of our work can be used as a reference guide. It should be noted that the second goal in the work of Ampatzoglou et al. (2013) is evaluating the effectiveness of GoF DPs on software quality, while this purpose is not considered in our study.

Table 1: Systematic reviews on DPs field

No.	Title	Ref.
1	How have we evaluated software pattern application? A systematic mapping study of research design practices	(Riaz et al. 2015)
2	Research state of the art on GoF design patterns: A mapping study	(Ampatzoglou et al. 2013)
3	A survey of experienced user perceptions about software design patterns	(Zhang and Budgen 2013)
4	What Do We Know about the Effectiveness of Software Design Patterns?	(Zhang and Budgen 2012)

[1] http://sqlab.um.ac.ir/images/219/files/SQLLabSM.pdf

TABLE 2: COMPARISON BETWEEN OUR WORK AND OTHER REVIEWS

Source	Review type	Number of primary studies	Number of considered journals, conferences, and workshops	Time interval	Scope	Goals
(Riaz et al. 2015)	SMS	30	-	up to September 2014	Software patterns research field	To define the research design of empirical studies which focus on applying software patterns involving human participants.
(Zhang and Budgen 2013)	SMS	10	-	up to the end of 2007	GoF DPs research field	To recognize which patterns from the set cataloged by the GoF are considered to be useful and which ones are regarded as not being useful, and why this is so.
(Ampatzoglou et al. 2013)	SMS	118	29	Up to 2010	GoF DPs research field	To present an overview of the research efforts on GoF DPs.
(Zhang and Budgen 2012)	SMS	10	-	up to the end of 2009	GoF DPs research field	To identify how extensively, the DPs usage has been subjected to empirical study and what proof is available about how and when their usage can offer a useful mechanism for knowledge transfer about design.
Our study	SMS	637	108	Up to March 2016	DPs research field	To provide an overview of the research efforts on DPs for those researchers who seek to enter this field.

3 RESEARCH METHOD

The review process used in this paper (see Fig. 1) is based on the updated guidelines for performing SMS which is presented by Petersen et al. (2015). In the work of Petersen et al. (2015) the authors conducted an SMS on SMSs in software engineering. Moreover, they presented a set of updated guidelines by considering the lessons learned through their mapping study and the guidelines presented in the work of Kitchenham (2004), Arksey and O'Malley (2005), Biolchini et al. (2005), Kitchenham and Charters (2007), Petersen et al. (2008), and Budgen et al. (2008). In the following sub-sections, the details of our method are discussed.

3.1 Planning the mapping study

In this phase, all the protocols and strategies which are needed for conducting a mapping study should be specified. Below, these protocols and strategies are described.

3.1.1 Specifying the scope and research questions (RQs)

The most important part of a systematic review is specifying the questions we intend to answer in the review process. These questions determine our goals, and the process of reviewing will be progressed by responding to these questions.

This study has been conducted from Jun 2015 until March 2016 and so we considered the studies that have been published until March 2016. The main goal is to identify and categorize research topics that exist in the field of DPs. So, the scope of our research is all the studies which have been conducted in this field. Furthermore, we want to present a reference guide for those who want to be active in the DPs area. Therefore, the research questions have been adjusted to achieve this goal. In the following, the research questions of our mapping study are described.

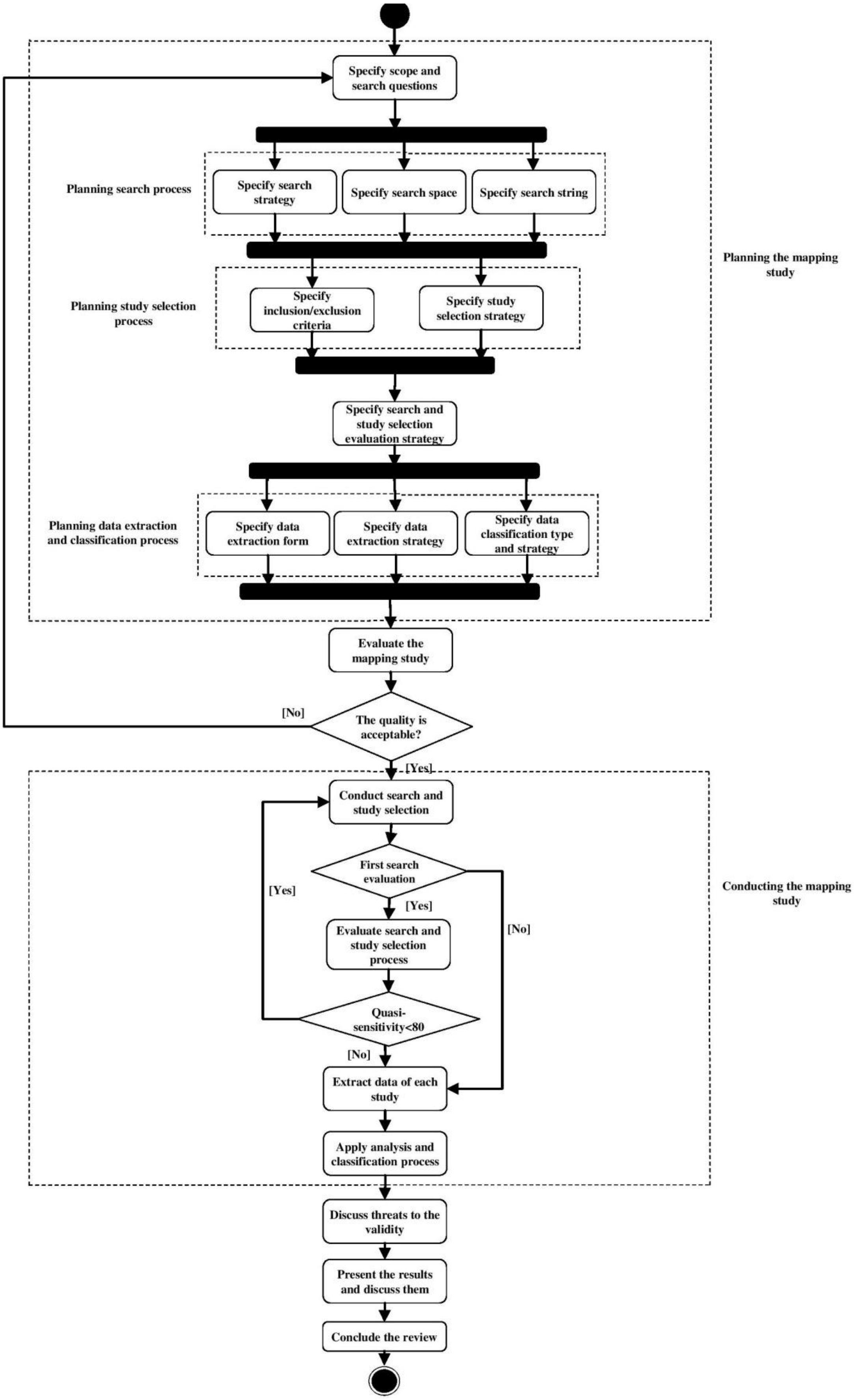
Specify scope and search questions
Planning search process
Specify search strategy
Specify search space
Specify search string
Planning the mapping study
Planning study selection process
Specify inclusion/exclusion criteria
Specify study selection strategy
Specify search and study selection evaluation strategy
Planning data extraction and classification process
Specify data extraction form
Specify data extraction strategy
Specify data classification type and strategy
Evaluate the mapping study
[No]
The quality is acceptable?
[Yes]
Conduct search and study selection
First search evaluation
[Yes]
[Yes]
[No]
Conducting the mapping study
Evaluate search and study selection process
Quasi-sensitivity<80
[No]
Extract data of each study
Apply analysis and classification process
Discuss threats to the validity
Present the results and discuss them
Conclude the review

Fig. 1. The review process

RQ1: what are the core research topics in the field of DPs and how much emphasis has been given towards the field? Rational: by answering this question, we intend to meet the following goals: 1) finding out how attractive and important the research field of DPs has been for other scholars, how much work has been done in this area, and what has been the trend of the activity in this field, 2) identifying and classifying themes that exist in the DPs field (using the obtained taxonomy, researchers can explore existing topics and sub-topics in this field), and 3) identifying the trends and gaps in each topic.

RQ2: which countries, researchers, and venues have been more active in the area of DPs? Rational: by answering this question a useful starting point will be provided for other researchers who are interested in this field.

3.1.2 Planning the search process

After specifying scope and research questions, we defined all the protocols and decisions which are relevant to conducting the search process including specifying search strategy, search space, and search string. The descriptions are given below.

3.1.2.1 Specifying the search strategy

For conducting the search, at first, the search strategy should be specified. In this work, we used manual search and backward snowballing (Jalali and Wohlin 2012) for the identification of relevant studies. We should note some points: 1) before starting the process, the set of search spaces (which is, at first, an empty set and will gradually be completed as the search process progresses) should be defined. The members of this set are pairs of “search space” and “related timestamp”. 2) The activity of identifying the related journals, conferences or workshops of newly included papers is not necessary to be conducted after the manual search. Therefore, we have defined a flag named S_Flag which is set to one in the backward snowballing step and is set to zero in the second step of the process.

Fig. 2 shows all the steps in our search strategy. This strategy consists of five steps. The description of each step is as follows:

Step 1: at the first step, the initial set of papers should be selected. As described in Section 3.3.1, we reached this set by conducting an informal search.

Step 2: in this step, the related journal, conference or workshop of each included paper should be extracted and compared with members of the search space set. If there is any new journal, conference or workshop, the third step should be executed; otherwise, the fifth step of the search process will follow.

Step 3: the extracted journals, conferences or workshops with the related timestamp (current time) should be added to the search space set, and a manual search on them should be done based on the search string (in order to find more relevant papers).

Step 4: in this step, the study selection process (see Section 3.1.3.2) should be applied. Depending on whether a paper is included or not, and whether the included paper is the result of the backward snowballing step or not, we should end the process or move to step 2, or 5.

Step 5: at the last step, the backward snowballing, which is finding the relevant papers by investigating references of included papers, should be conducted (on those papers which have not been snowballed before). If any new paper is found, the search process continues from step 4. Otherwise, it ends.

Note: our search process has been done from Jun 2015 until March 2016. If during this time interval each search space had been examined only once, in some search spaces new articles which were published after the first manual search would have been ignored. This is a threat to the completeness of the included papers set. To remove this threat, in the third step of the search strategy, a timestamp for each included search space should be inserted. After finishing the search process, the timestamps will be checked. If there exists a case which is greater than or equal to six months passed, the related search space should be investigated again (since the timestamp, until the current time).

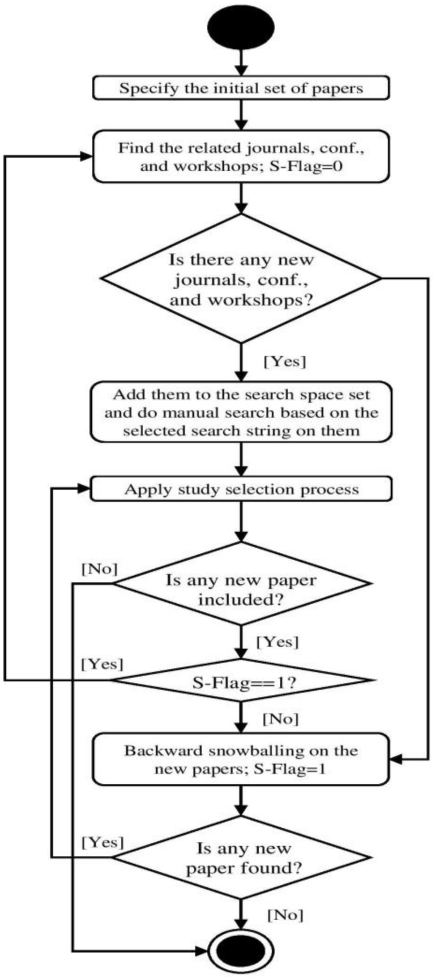
Specify the initial set of papers
Find the related journals, conf., and workshops; S-Flag=0
Is there any new journals, conf., and workshops?
[Yes]
Add them to the search space set and do manual search based on the selected search string on them
Apply study selection process
[No]
[No]
Is any new paper included?
[Yes]
[Yes]
S-Flag==1?
[No]
Backward snowballing on the new papers; S-Flag=1
[Yes]
Is any new paper found?
[No]

Fig. 2. Search strategy

3.1.2.2 Specifying the search space

According to the search strategy, at the starting point of the search process, the search space set (a set of journals, conferences, and workshops) is empty. After conducting an informal search, some search spaces will be added to the search set (see Section 3.3.1). This set will be completed with the progress of the search process. In Appendix → Table C and D, the final search space set and related information are shown. More information can be found in DPs-SM → Chapter 1 → Tables 1-3 to 1-7.

3.1.2.3 Specifying the search string

According to the context of our mapping study which is software DPs, we chose “design pattern” as the search string and applied it in the title, abstract, and full text of the papers.

3.1.3 Planning the study selection process

For the study selection, two important tasks should be done. First, inclusion/ exclusion criteria should be specified and second, the strategy of study selection should be defined

3.1.3.1 Specifying the inclusion/exclusion criteria

For the identification of those primary studies that can answer our research questions, we should define two categories of criteria (called study selection criteria). The first category is the inclusion criteria (see Table 3). In the current mapping study, one inclusion criterion is defined. The second category is the exclusion criteria (see Table 4) which exclude those articles that are not desired. If an article has at least one of the exclusion criteria, it will be excluded. In other words, a paper will be included if the inclusion criterion is met and none of the exclusion criteria is satisfied. Note, according to Table 4, the secondary studies should not be included. However, we considered these studies just to compare them with our study.

Here, the thresholds used for the exclusion criteria are obtained empirically. For selecting the appropriate threshold values, we considered two issues: 1) slight changes in these values should not make many changes in the number of excluded/ included papers and 2) threshold values should be selected in such a way that applying them does not exclude a lot of highly cited papers.

TABLE 3: INCLUSION CRITERIA

No.	Description
1	The focus of the paper is on the software DPs

TABLE 4: EXCLUSION CRITERIA

No.	Description
1	The study relates to other patterns, not DPs, e.g. architectural patterns
2	If the study is a journal paper and the related journal is not indexed in the list of ISI journals
3	If the study is a journal paper and for the related journal $\big((SJR\ is\ not\ available)\ or\ (SJR < 0.4)\big)$
4	If the study is a conference paper and for the related conference $(Qualis < B4)\ and\ (ERA < B)\ and\ (h5 - index < 10)\ and\ (Period < 10)$
5	The study is not a paper, e.g. it is a gray literature, conference cover, poster, etc.
6	The study is not a primary study, e.g. it is a survey, literature review, etc.

3.1.3.2 Specifying the study selection strategy

To include relevant papers, we specified the study selection strategy which is shown in Fig. 3. This strategy consists of two evaluation stages. At the first stage, the paper should be evaluated by title, keywords, information of the publication venue, and abstract. At the second stage, the paper should be evaluated by full-text. After each stage, a decision should be made. Note that in each evaluation stage both inclusion and exclusion criteria should be considered. In each stage, meaningful terms and keywords of included papers should be highlighted. This is useful for the classification of these papers.

Each paper should be evaluated by two reviewers (first and third author) separately, and they should check their results with each other at the end. In Table 5, the decisions rules are defined. According to this table, nine states may occur. In the case of "uncertain" condition, one other reviewer (second author) should judge between them. Note, if there is still a disagreement, even after the judgment, the paper should be included.

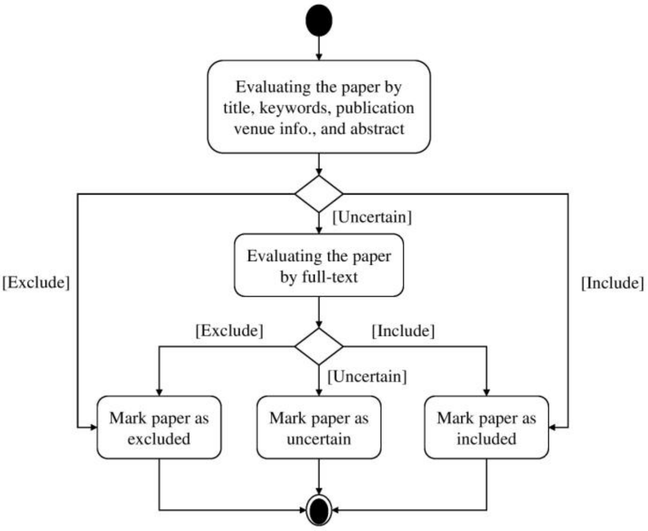

Fig. 3. Study selection strategy

TABLE 5: DECISIONS RULES

STATE	INCLUDE	EXCLUDE	UNCERTAIN
Include	Include	Uncertain	Include
Exclude	Uncertain	Exclude	Exclude
Uncertain	Include	Exclude	Uncertain

3.1.4 Specifying the search and the study selection evaluation strategy

Evaluating the search strategy can be done in two ways: 1) subjective evaluation (using the expert review), 2) objective evaluation (using the quantitative criteria). The subjective evaluation could be biased in terms of the expert opinion. Therefore, in this study, we used a quantitative criterion (quasi-sensitivity) to evaluate the search and the study selection process. “Sensitivity” is the proportion of the included studies for a given topic and can be calculated by the following equation:

$$\text{Sensitivity} = (\text{Number of included studies} / \text{Number of studies in a full set of studies})*100 \qquad (1)$$

However, obtaining a full set of primary studies (gold standard) is almost impossible. So, we used the Quasi-Gold Standard (QGS) to obtain the sensitivity (Zhang et al. 2011). QGS is a set of known studies from key sources, which are recognized by the community in the subject. In this mapping study, to establish the QGS, we got help from eight members of the software quality lab[2].

The lab members identify some key researchers in the field of DPs, during the informal search from sources such as c2.wiki website, books, courses, and editor-in-chief of journals or conferences. Afterward, referring to web-pages of the key researchers and gathering their related papers (after applying the inclusion and exclusion criteria), the QGS was generated. In Appendix →Table E and Table F, the authors, web-pages, and extracted papers are shown. More information can be found in DPs-SM → Chapter 2 → Table 2-8.

After establishing the QGS, and calculating the quasi-sensitivity, we should compare the obtained value with a threshold. If the quasi-sensitivity was less than the predefined threshold, we should conduct the search and study selection again on the newly found papers in QGS. According to Zhang et al. (2011), it is suggested that an acceptable threshold should be between 70% and 80%. In this study, we chose 80% as the threshold.

3.1.5 Planning the data extraction and the classification process

To extract data from the included studies, we prepared data extraction forms. In Table 6, the items of these forms and the reason for considering each item is shown. Data extraction process was done in two stages. At first, we divided included papers into two sets, and the data extraction operation was performed by two reviewers (for each set, one reviewer). At the second stage, the results of each reviewer were checked and traced back by a third reviewer. If the third researcher found any inconsistencies, the issue was raised in the group meetings to reach an agreement.

TABLE 6: ITEMS OF DATA EXTRACTION FORM

Item	Rational
Article name	For identifying the article
Authors name	For answering RQ2
Year	For answering RQ1
Venue	For answering RQ2
Extracted keywords	For answering RQ1
General concept	For answering RQ1
Country	For answering RQ2

[2] http:/ / sqlab.um.ac.ir/ index.php?lang=en

According to Petersen et al. (2015) the differences between general classification and topic dependent classification are distinguished. In this study, for the general classification, applied facets are the venue, year, and country. To achieve the research tree of the DPs field and identifying research topics, we applied keyword clustering which is a common method for topic dependent classification (see Section 3.3.4).

3.2 Evaluating the mapping study

To evaluate our mapping study, we used the evaluation rubrics presented by Petersen et al. (2015). In Table 7, the rubrics, the related actions which have been done in this study, and the evaluation degree are shown. According to the related actions, we assigned a degree (no description, minimal, partial, and full) to each rubric based on the requirements defined by Petersen et al. (2015). In the last column of Table 7, for each rubric, the ratio of the number of mapping studies with the same evaluation degree, to the total number of mapping studies investigated by Petersen et al. (2015) is shown. The evaluation results demonstrate that this mapping study has been conducted on the acceptable level of quality, compared with other studies investigated by Petersen et al. (2015). Note, if the quality was not acceptable, the planning phase should be revised (according to each rubric).

TABLE 7: EVALUATION RUBRICS

Rubric	Actions	Evaluation degree	Evaluation degree of the mapping studies investigated by Petersen et al. (2015)
Specifying scope and research questions	Need of the study, motivations, objectives and research questions are defined	partial	100% partial
Specifying search strategy	Manual search and backward snowballing are used	minimal	About 40% no description, about 23% minimal, about 19% full
Evaluating search and study selection process	The quasi-gold standard is used to evaluate the search and selection process; an additional person is added to judge when disagreement occurs, and decision rules are taken	partial	About 44% no description, about 42% minimal, about 14% partial
Data extraction and classification	An additional person is added to trace back and check extracted data, two classification schemes (general classification and topic dependent classification) were used	minimal	About 10% no description, about 61% minimal, about 29% partial
Threats to validity	The threats to validity are discussed	full	About 14% no description, about 86% full

3.3 Conducting the mapping study

After specifying the protocols and strategies of the mapping process, they should be applied in the conducting phase. The details are explained below.

3.3.1 Conducting the search and the study selection process

As mentioned before, to start the search process, the initial set of papers should be determined. After obtaining the initial set, the rest of the search process was performed according to the search and the selection strategies. The second author identified the initial set by performing the following steps. In the first step, an informal search was carried out to find the secondary studies which had been conducted in the field of DPs. The informal search was undertaken using 'Google Scholar' and the search term 'design pattern' combined with terms such as 'review', 'systematic review', 'systematic mapping study', 'systematic literature review', or 'survey'. Then, according to the study selection criteria (exclusion criteria (6)), some secondary studies were selected (see Appendix →Table A). In the next step, by investigating the primary studies which had been considered by the secondary studies obtained from the previous step, 303 papers were identified as the candidate members of the initial set. Afterwards, by applying the selection process on the candidate members, 101 papers remained and formed the final initial set (see Appendix →Table B).

In general, by conducting the search process, 36 journals and 53 conferences (or workshops) were identified as the desired search spaces and a total of 1484 conference papers and 965 journal papers were identified as relevant papers. After applying the study selection strategy, 155 journal papers and 368 conference or workshop papers remained and were included. Note, in Appendix →Tables C, and D, the number of the relevant papers (before applying study selection process) and the number of the included papers (after applying study selection process) of each search space are shown.

3.3.2 Evaluating the search and the study selection process

In our study, we achieved the sensitivity of 79%. So, we conducted the search and study selection again on the newly found papers in QGS. In the QGS, there were 23 new papers which we did not include (because the venues of them were not considered in our search). We took these papers as the initial set and conducted the search strategy on them. Finally, 3 new journals and 16 new conferences were added to the search space set and subsequently 93 conference papers and 21 journal papers were added to our included set. In Appendix → Table F, data extracted for the evaluation phase are shown. More details of this phase are presented in DPs-SM → Chapter 2 → Table 2-8.

3.3.3 Data extraction

At this stage, according to the data extraction forms, useful data were extracted. As we mentioned earlier, the extraction has been conducted by two reviewers and checked by the third one. In the DPs-SM →Chapter 1 → Tables 1-8 to 1-11, the filled forms are available.

3.3.4 Analysis and classification

The data extracted in the previous step were analyzed and tabulated according to the related research question(s). For example, to answer the RQ1, the papers were tabulated based on the year and counted afterward. In Section 4, the results of this stage are visually illustrated.

As mentioned before, the main goal of an SMS is classifying the primary studies that have been included in the intended subject. In the current systematic mapping, using keyword clustering we classified those primary studies that have been included in the field of DPs. We applied the keyword clustering, as mentioned by Febrero et al. (2014) and Noyons et al. (2000), on the meaningful terms (keywords) which was extracted at the first step from the title, author's keywords, and abstract of the primary studies. These meaningful terms should reflect the purpose or contribution of the paper. Here, at first, about 1135 keywords and phrases were extracted as the meaningful terms. Then, we characterized the most frequent terms. We identified about 30 frequent terms which were repeated more than 10 times (see DPs-SM →Chapter 1 →Tables 1-11), and clustered them into 5 groups on the basis of their similar cognitive orientation. These groups were obtained by a hierarchical clustering algorithm with complete linkage on the keyword normalized co-occurrence matrix (Noyons et al. 2000). In the keyword co-occurrence matrix both the rows and the columns represent the keywords and each cell shows the number of times that the keywords corresponding to the row and the column of that cell appear together in a study (see DPs-SM →Chapter 1 →Tables 1-12). Then, based on the cosine index (see eq. (2)), we normalized the co-occurrence matrix.

$$\mathrm{sim(i,j)} = \frac{\sum_{k=1}^{n} X_{ik}X_{kj}}{\sqrt{((\sum_{k=1}^{n} X_{ik}^2)(\sum_{k=1}^{n} X_{kj}^2))}}. \qquad (2)$$

Where X_{ij} is the number of co-occurrences of keyword i with keyword j.

In the recalculated matrix, the similarity of keywords will be based on the cognitive orientation of two keywords in relation to all other keywords. Note, we determine the number of clusters empirically. We chose this number in such a way that the papers can be divided into balanced clusters (as much as possible). In the last step, a proper name for each cluster should be selected according to the most frequent keywords and the whole concept of the cluster (see Table 8).

By applying the above-mentioned method on the frequent keywords in the field of DPs, we identified the following research topics: 1- Pattern Development, 2- Pattern Usage, 3- Pattern Mining, 4- Quality Evaluation, and 5- Pattern Specification.

After specifying the main category, we specialized and divided these topics into some possible sub-topics. To do so, based on the authors' field experience, we considered a suitable discriminator for each topic. Then, each topic was divided based on the corresponding discriminator, and a complete classification was achieved. The details of each category and the related sub-categories are given in Section 4.

Finally, we mapped each paper to its corresponding category based on its extracted keywords. If in a particular paper any of the frequent keywords existed, the paper was mapped to the category to which the keywords belong. Otherwise, we considered the frequent keywords and found semantically similar keywords in the paper. Then, based on these similar keywords, we mapped the paper to its corresponding category. If a paper was found which could not be mapped to any of the categories, we mapped it to the Miscellaneous Issues category. Note that in some cases, according to the extracted keywords, a paper may belong to more than one category. These papers were judged, and the most related category was chosen.

TABLE 8: KEYWORDS AND RELATED CONCEPTS OF CATEGORIES

Category	Keywords	Concept
Pattern Development	Name of a new pattern (e.g. OCO design pattern), Categorization/ Classification/ Cataloging, (Design) Pattern language(s), Pattern(s) Composition(s)/ Composition, (Software) Evolution	Any development in the field of DPs research
Pattern Usage	Using, (DP(s)) Applying/ Application	Designing domain-specific applications with patterns, presenting pattern-based methods, or Incorporating and applying DPs in the software design/ code
Pattern Mining	Detection, Recovery, Identification, Recognition	Finding instances of the pattern in the system code/ design.
Quality Evaluation	Quality, (Quality) Evaluation/ Measurement/ Assessment, Quality attributes (such as maintenance and reusability), (Evaluation by) Metric(s), (Evaluation by) Empirical/ Experimental (Methods)	Assessing the quality of the patterns or assessing the impact of applying DPs on the quality of the system
Pattern Specification	(Pattern) Specification/ Specifying, Formalism/ Formal (method(s)), Visualization/ Visual, Modeling Languages (such as UML), Verification/ Verifying	Using different methods and notations for describing the patterns

4 RESULTS

In this section, the results are presented with respect to the research questions specified in Section 3.1.1. First, in Section 4.1 research topics and sub-topics that have been identified in this review, and frequency of papers are presented. Next, in Section 4.2, active countries, researchers, and venues in the field of DPs are provided.

4.1 Identifying the research topics and frequency of papers (RQ1)

In the late 1960s and early 70s, Christopher Alexander introduced the patterns concept in architecture (Najari et al. 2016). In 1993, it was suggested to forge a marriage of objects and patterns. Then, in 1995, the idea of patterns was adopted in object-oriented software development (c2.wiki 2016). In the same year, Gamma's book was released (Gamma et al. 1995). According to scholars, Gamma's book is the most effective activity to pick up the topic of DPs (Ampatzoglou et al. 2015). In Gamma's book, with the aim of solving some commonly-recurring design problems, 23 DPs are cataloged (Gamma et al. 1995).

According to the above-mentioned explanations, we considered the time interval starting from 1995. Fig. 4 illustrates the number of papers on DPs during the last two decades (from 1995 to 2015). Note, since we did not consider the full year of 2016 in our review, the number of the publications in 2016 is not presented in this figure (from the beginning of 2016 to March, only two papers were included). As seen in Fig. 4, from the total number of 637 papers including archival journal and conference articles, about 62% of articles (393 papers) have been published in the recent decade. Therefore, it can be concluded that similar to some old topics such as software architectural decisions (Tofan et al. 2015) or trending topics such as technical debt (Li et al. 2015), DPs research is a hot topic and the attention to the subject is increasing.

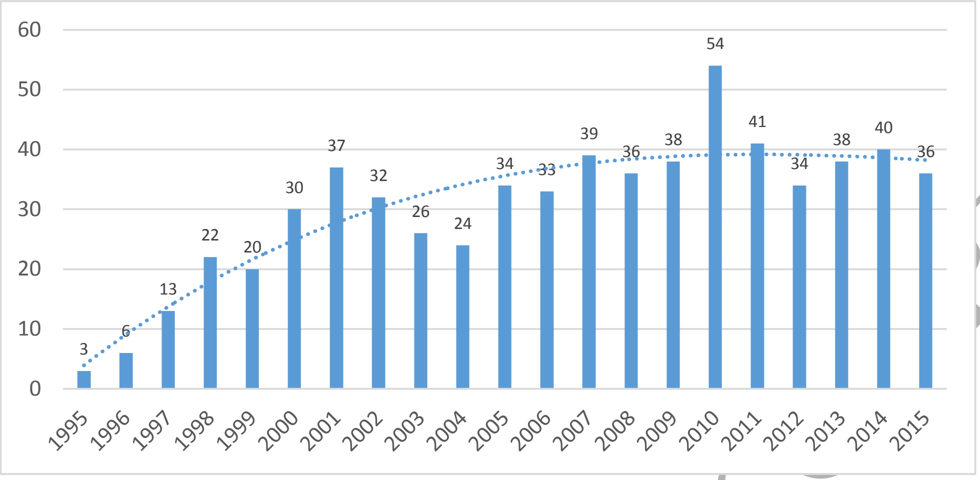

Fig. 4. Frequency of publications per year

For identifying the research topics, as mentioned in Section 3.3.4, we used the keyword clustering method. Extracting the keywords and meaningful terms of the primary studies and presenting the most repeated of them is a common and good practice in an SMS. In this mapping study, about 1135 keywords and phrases were extracted. Fig. 5, shows 30 most frequent of these keywords and phrases. As seen, "Security Patterns" (representative of words such as (Software) Security pattern(s), Security design pattern(s), or Security), "Object Orientation" (representative of words such as Object Orientation, or OO), and "Pattern Detection" (representative of words such as ((Design) pattern(s)) Detection, or Detecting) are the most frequent phrases in the literature of DPs.

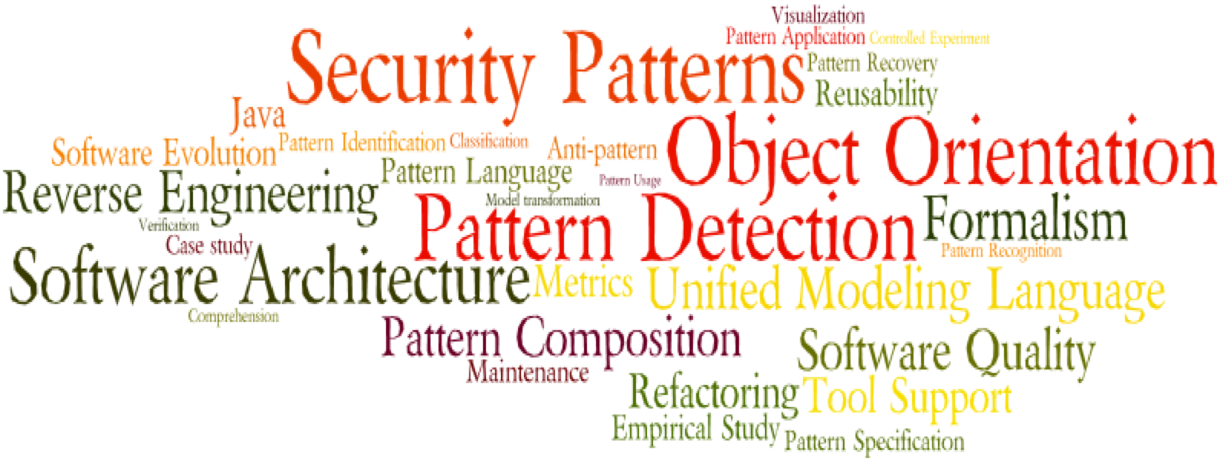

Fig. 5. Common keywords and phrases in DPs field, drawn with http://www.wordle.net/

By applying the keyword clustering algorithm (see section 3.3.4), we obtained a classification which is shown in Fig. 6. The main categories of this classification are Pattern Development, Pattern Usage, Pattern Mining, Quality Evaluation, Pattern Specification, and Miscellaneous Issues. In Table 8, the extracted keywords that were useful for classification is shown. Below, the details of each category (topic) and sub-category (sub-topic) are given.

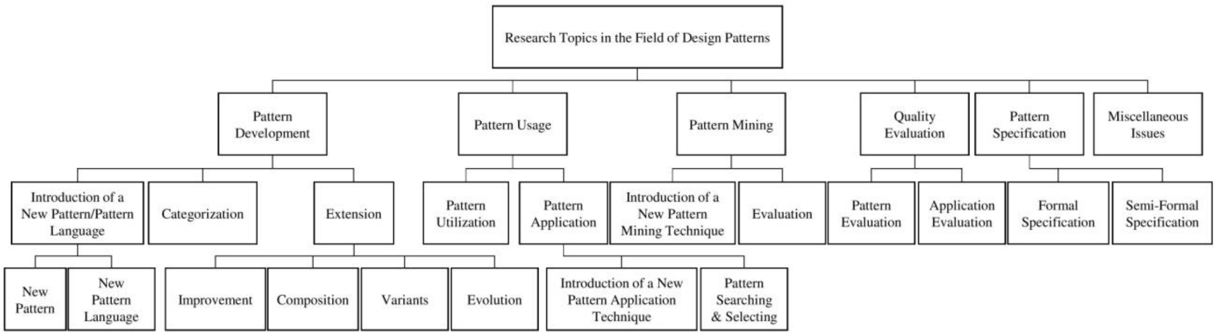

Fig. 6. DPs research tree

The Pattern Development category

Pattern Development category indicates any development in the field of DPs research, including:

- Introducing a new pattern or new pattern language. A pattern language is a collection of interrelated DPs that are carefully organized to characterize a design methodology. Note that a pattern language is more than only a catalog of patterns and the structure of the pattern language guides the user to design complex systems using the patterns (Massingill et al. 2001; Zhao et al. 2008; Pauwels et al. 2010; Hafiz 2013). "New pattern language" and "New pattern" sub-categories, refer to these definitions.
- Discussing pattern variants, pattern evolution, composing patterns or improving a specific pattern. Variability in patterns refers to the changes (in structure or behavior) that can occur inside of a DP when it is implemented while keeping the main parts of the pattern (Kim and Shen 2008). DPs embed some special ways for future changes. Evolution in DPs implies future changes in patterns during the evolution of the software (Schanz and Izurieta 2010; Griffith and Izurieta 2014; Rossi and Russo 2014). Pattern composition means assigning multiple elements of the pattern into overlapping sets of system classes (Dong 2002; Dong et al. 2007; Cacho et al. 2014). Pattern improvement indicates introducing an alternative for an existing pattern with the aim of overcoming its limitations (Lyon and Castellanos 2007; Fortuin 2010; Pati and Hill 2012). To cover these issues, the "Extension" sub-category which is divided into "Variants", "Composition", "Improvement", and "Evolution" is defined.
- Categorizing existing DPs in various fields such as mobile applications (Nilsson 2009), security (Hafiz et al. 2007; Van-Hilst 2009), model transformation (Lano and Kolahdouz-Rahimi 2014), and ontology (Tsai et al. 2013). These issues are listed under the "Categorization" sub-category.

In Table 9, the most frequent keywords in "Pattern Development category" are shown. Term frequency for each keyword refers to the number of papers in which the keyword existed. As shown, the main keywords of the Pattern Development category are "(Software) Security pattern(s)/ Security design pattern(s)/ Security", "Pattern(s) Composition(s)/ Composition", "(Software) Architecture", "(Software) Evolution", "(Design) Pattern language(s)", "Object Orientation/ OO", and "Categorization/ Classification/ Cataloging". As mentioned, one of the sub-topics in this class is introducing a new pattern or a new pattern language. High repetition of the phrase "Security pattern(s)" indicates that security is one of the most interesting topics in which researchers tend to introduce new pattern or pattern language.

TABLE 9: THE MOST FREQUENT KEYWORDS IN "PATTERN DEVELOPMENT" CATEGORY

Keywords	Frequency
(Software) Security pattern(s)/ Security design pattern(s)/ Security	31
Pattern(s) Composition(s)/ Composition	27
(Software) Architecture	23
(Software) Evolution	19
(Design) Pattern language(s)	16
Object Orientation/ OO	15
Categorization/ Classification/ Cataloging	12

The Pattern Specification category

The Pattern Specification category includes those papers which use different methods and notations for representing the patterns. There are various specification schemes which are proposed to complement the natural language description of the patterns. These schemes can be classified into two main groups:

- Formal specification schemes: specification of a program's properties by mathematically based techniques (formal languages) which are well understood. A formal language for DPs is a language with a precisely defined vocabulary, syntax, and semantics which aims to reduce the ambiguities by rigorously reasoning about the structural and behavioral aspects of patterns (Kim and Carrington 2009; Zhu and Bayley 2013).
- Semi-Formal specification schemes: specifying DPs with semi-formal languages. A semi-formal language is a language with a precisely defined vocabulary and syntax, but without a precisely defined semantics (Dong et al. 2007; Porras and Guéhéneuc 2010).

"Formalism/ Formal (method(s))", "Unified Modeling Language (UML)", and "(Pattern) Specification/ Specifying", are the most repeated keywords in the Pattern Specification group (Table 10). This means that formal methods have been greatly considered by researchers in the field of specifying patterns and UML is the most attractive semi-formal model for visualizing DPs.

TABLE 10: THE MOST FREQUENT KEYWORDS IN "PATTERN SPECIFICATION" CATEGORY

Keywords	Frequency
Formalism/ Formal (method(s))	32
Unified Modeling Language (UML)	22
(Pattern) Specification/ Specifying	17
Object Orientation/ OO	9
Verification/ Verifying	8
Visualization/ Visual	6

The Pattern Usage category

This category consists of the papers with the contribution of employing patterns in the software development process. This topic can be classified into two main classes:

- Pattern Utilization: presenting pattern-based methods or designing domain-specific applications with DPs. DPs can be used in various fields and applications such as security (Fernandez et al. 2013; Uzunov et al. 2015), CAD models (Bai et al. 2016), and object-oriented programming (Kienzle and Romanovsky 2002). Using DPs can help to standardize the design concepts, capture design experiences, reuse elegant and efficient solutions, and improve documentation and maintenance.
    - Pattern Application: incorporating and applying DPs in the software design/ code. Papers in the "Introduction" subcategory try to assist the process of applying patterns using manual or automatic methods. However, the "Pattern Searching & Selecting" subcategory deals with the problem of finding a suitable DP to apply and solve a particular problem (Aliaksandr Birukou 2010). Searching patterns imply obtaining information about existing patterns in the literature and online repositories, and selecting patterns means choosing an appropriate pattern in the list which is obtained at the stage of searching patterns or is pre-defined (Aliaksandr Birukou 2010; Sarun and Weenawadee 2007; Zdun 2007; Hasheminejad and Jalili 2012).

According to Table 11, "(software) Security pattern(s)/ security design pattern(s)", "(Software) Architecture", "Object Orientation/ OO" are the most common keywords in this field. This shows that the security and the software architecture are areas in which DPs are most widely used.

TABLE 11: THE MOST FREQUENT KEYWORDS IN "PATTERN USAGE" CATEGORY

Keywords	Frequency
(software) Security pattern(s)/ security design pattern(s)	30
(Software) Architecture	14
Object Orientation/ OO	12
Java	10
Using (design) pattern	10
(DP(s)) Applying/ Application	10

The Quality Evaluation category

Whenever a designer or a developer employs a pattern, one of the most important concerns is the quality of the pattern and the effect of the pattern on the system after applying. The Quality Evaluation category includes those papers with the purpose of quality evaluation in the field of DPs. This topic can be classified into two main classes:

- Pattern evaluation: assessing the quality of the patterns (Halkidis et al. 2006; Ampatzoglou et al. 2011; Elish and Mohammed 2015).
- Application evaluation: assessing the impact of applying DPs on the quality of the software system (Jaafar et al. 2014; Scanniello et al. 2015; Ampatzoglou et al. 2015).

Assessing the pattern quality is a challenging task. Similarly, evaluating the effect of patterns on quality attributes is tough. As the current state of the art stands, evaluation of DPs has been performed through two main activities: 1) computing code metrics (quantitative methods), 2) using expert opinion (empirical methods) (Ampatzoglou et al. 2013). As expected, concerning Table 12, it is observed that "(Software) Quality/ Qualitative analysis", "Metric(s)", "Empirical (study(s))", and "(Controlled) Experiment" are frequent keywords in the studies that evaluate the quality of the patterns or the impact of applying them on system quality.

TABLE 12: THE MOST FREQUENT KEYWORDS IN "QUALITY EVALUATION" CATEGORY

Keywords	Frequency
(Software) Quality/ Qualitative analysis	33
Metric(s)	13
Empirical (study(s))	13
Object Orientation/ OO	12
(Controlled) Experiment	9

The Pattern Mining category

Finding instances of the pattern in the systems source code or design is the purpose of the papers which were included in this category. Detection of DP motifs can provide a better glimpse to recognize the original design decisions. Hence, it is important in the process of refactoring, reengineering, and maintenance. Furthermore, it can be helpful for software quality measurement, program understanding, and improving software documentation. However, this is not a straightforward task. Papers which discuss the Pattern Mining field can be classified into two main groups:

- Introduction: Introducing a new method for the detection of DPs (Yu et al. 2015; Zanoni et al. 2015; Wen-Jin et al. 2015).
- Evaluation: Evaluating mining techniques for DPs (Bouassida et al. 2013; Ujhelyi et al. 2015).

Table 13 elucidates that the keywords used in the published papers on mining DPs are mostly "((Design) pattern(s)) Detection/ Detecting", "Reverse engineering/ Re-engineering", "((Design) pattern(s)) Recovery", and "((Design) pattern(s)) Identification/ Identifying". "Detection", "Recovery", and "Identification" are synonymous with the word "Mining". DP Mining is a significant part of the reverse engineering process that can aid program comprehension. Hence, the "Reverse engineering" keyword is repeated commonly in the Pattern Mining studies.

TABLE 13: THE MOST FREQUENT KEYWORDS IN "PATTERN MINING" CATEGORY

Keywords	Frequency
((Design) pattern(s)) Detection/ Detecting	64
Reverse engineering/ Re-engineering	36
((Design) pattern(s)) Recovery	19
((Design) pattern(s)) Identification/ Identifying	18
Tool (Support)	14
Object Orientation/ OO	12
((Design) pattern(s)) Recognition	12
Antipattern	10

The Miscellaneous Issues category

This class includes studies that cannot be classified into any other previous topics (Ampatzoglou et al. 2013; Gaitani et al. 2015; Aljasser 2016). These studies typically discuss issues such as, refactoring, code smells & anti-patterns, and implementing patterns in a special language. Miscellaneous Issues cannot be regarded as individual topics because they reflect enormous heterogeneity, and they lack enough maturity.

Table 14 demonstrates that the most frequent keyword in the Miscellaneous Issues is "Refactoring". Refactoring is the process of restructuring a software system in such a way that it improves the internal structure of the system without altering the external behavior of the code. Designs can be improved with patterns by applying sequences of design transformations, known as refactoring (Kerievsky 2005).

TABLE 14: THE MOST FREQUENT KEYWORDS IN "MISCELLANEOUS ISSUES" CATEGORY

Keywords	Frequency
Refactoring	20
Object Orientation/ OO	17
(Software) Architecture	8
Tool (Support)	8
Reuse/ Reusability	8

Fig. 7 depicts the percentage of publications for the diverse topics of DPs. As observed, most of the scholars have focused on Pattern Development, Pattern Mining, and Pattern Usage, while less consideration is given to Pattern Specification, and Quality Evaluation categories.

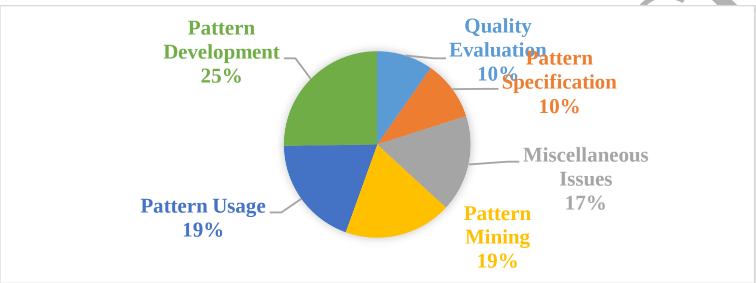

Fig. 7. Percentage of publications per topic

Table 15 has been presented to show the number of publications in different fields of DPs (as discussed in Fig. 7) for different years. It is observed that, in general, three branches of DPs including Pattern Development, Pattern Mining, and Pattern Usage had the highest growth in the number of publications during the recent years. Moreover, in this table, we have highlighted the most frequent categories for each year.

TABLE 15: EVOLUTION OF DPS TOPICS PUBLICATIONS OVER TIME

		Category					
		Quality Evaluation	Pattern Specification	Miscellaneous Issues	Pattern Mining	Pattern Usage	Pattern Development
Number of publications per year	1995	0	0	1	1	0	1
	1996	0	1	2	1	0	2
	1997	0	1	7	1	3	1
	1998	0	2	6	1	7	6
	1999	0	1	2	2	10	5
	2000	2	4	4	3	11	6
	2001	6	4	2	6	6	13
	2002	2	3	7	4	7	9
	2003	1	4	4	4	10	3
	2004	3	6	5	4	2	4
	2005	3	2	3	13	4	9
	2006	2	4	6	8	5	8
	2007	5	4	5	5	6	14
	2008	5	3	4	9	4	11
	2009	4	3	7	12	4	8
	2010	5	6	14	12	5	12
	2011	5	8	6	6	7	9
	2012	8	0	4	4	10	8
	2013	3	8	3	6	9	9
	2014	4	1	8	6	7	14
	2015	4	1	6	11	5	9

4.2 Identification of active countries, researchers, and venues

The results obtained from the data analysis for this research question are shown in the following figures and tables.

The geographical distribution of publications (GDP) for a particular topic is an indicator of the productivity of the research topic of individual countries and has become a field of interest. In Fig. 8, the geographical distribution of publications in the field of DPs from 1995 to March 2016 is shown. As seen, USA has the highest share (23%) from the total number of publications. Canada, Germany, and Italy stand in the next ranks after the USA. According to our inspection of the country rankings provided by SCImago Country Rank in 2016 (SCImago 2016), USA, China, UK, Germany, Japan, France, Canada, and Italy are the most active countries in the number of scientific articles respectively. Similarly, USA, Canada, Germany, Italy, UK, China, and France are among the countries with the highest number of publications in the field of DPs. However, as shown in Fig. 8, there are some countries such as Greece and Brazil which their activity level in the DPs field are beyond what was expected. Considering the affiliation and the nationality of communities, projects, pioneers, and renowned researchers in the field of DPs research which illustrated in DPs-SM →Chapter 2 →Tables 2-1 to 2-2 and Table 2-7 and DPs-SM →Chapter 3 →Tables 3-2, it can be concluded that there exists a strong relationship between them and the number of papers produced in each country.

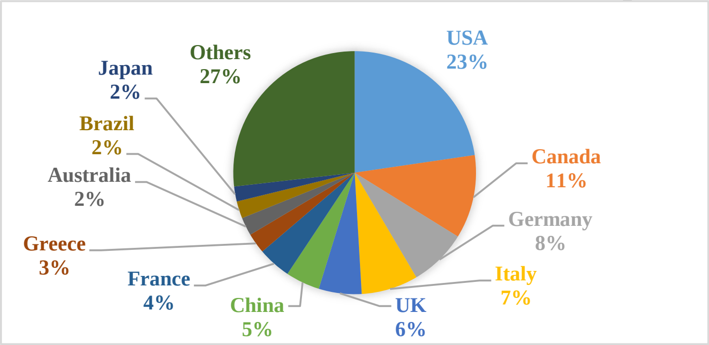

Fig. 8. Geographical distribution of publications

Identification of key scholars in a field of research can help those researchers who want to enter the research field. As mentioned before, from each included paper, we extracted the name of the authors. In total, 1078 authors were identified in the DPs field. In Table 16, ten most active researchers in the DPs field with their number of publications are presented. Moreover, in this table, we have highlighted the most important researcher per domain.

TABLE 16: TEN MOST ACTIVE RESEARCHERS IN DPS FIELD

	Quality Evaluation	Pattern Specification	Miscellaneous Issues	Pattern Mining	Pattern Usage	Pattern Development	Total
Yann-Gaël Guéhéneuc	7	2	2	20	3	2	36
Jing Dong	0	4	2	5	2	15	28
Eduardo B. Fernandez	0	0	1	0	6	8	15
Francesca Arcelli Fontana	1	0	0	13	0	0	14
Giuliano Antoniol	2	0	2	6	0	2	12
Dae-Kyoo Kim	0	5	2	0	2	2	11
Yajing Zhao	0	1	2	4	0	4	11
Apostolos Ampatzoglou	8	0	1	0	1	0	10
Giuseppe Di Lucca	2	1	1	6	0	0	10
Alexander Chatzigeorgiou	6	0	0	1	1	1	9

Table 17 demonstrates the most popular conferences in the field of DPs. The International Conference on Pattern Languages of Programs (PLoP) has published the largest number of conference papers related to DPs. European Conference on Object-Oriented Programming (ECOOP), Asia-Pacific Software Engineering Conference (APSEC), International Conference on Software Engineering (ICSE), and OOPSLA stand in the next ranks after the PLoP.

TABLE 17: ACTIVE CONFERENCES IN DPS FIELD

Name (or acronym)	Number of publications	Name (or acronym)	Number of publications	Name (or acronym)	Number of publications
PLoP	51	ICSOFT	6	ARES	3
ECOOP	45	SEW	6	HiPC	3
APSEC	24	SAC	5	PATTERNS	3
ICSE	22	QUATIC	5	CSEE&T	2
OOPSLA	22	DEXA	5	CSE	2
CSMR	19	CAiSE	5	SEFM	2
ASE	18	FASE	5	SysCon	2
WCRE/ SANER	14	ENASE	5	ICSC	2
TOOLS	14	SOSE	4	TASE	2
ICSME	12	ICPC	4	ICACTE	2
COMPSAC	12	MODELS/ UML	4	CiSE	2
EuroPLoP	12	PCI	4	SERP	2
ESEM	10	SNPD	4	EmbeddedCom	1
HICSS	7	ICECCS	4	SPLASH	1
IRI	7	CASCON	3	ASWEC	1
ICSEA	7	IJCAI	3	UKCI	1
ASEA	7	RE	3	SCAM	1
ICSESS	7	ECBS	3	ICIS	1
FSE	6	Mindtrek	3	DASC	1
SEKE	6	ICEIS	3	PASTE	1
ICQS	6	SERA	3	ICCES	1
STEP	6	swmetrics	3	IACC	1

Table 18 represents the distribution of the publications in DPs topics over the 13 most active conferences in the field. For example, this table shows that in the pattern development category, 28.15 percent of the conference papers are published in the PLoP conference. In this table, we have highlighted the two most active conferences in each topic. Note that the row entitled "Others" includes the total percentage of participation of the other conferences.

TABLE 18: DISTRIBUTION OF DPS TOPICS PUBLICATIONS OVER ACTIVE CONFERENCES IN THE FIELD

Topics Name	Pattern Development (%)	Pattern Usage (%)	Pattern Mining (%)	Quality Evaluation (%)	Pattern Specification (%)	Miscellaneous Issues (%)	Total (%)
PLoP	28.15	4.47	0	5.19	7.61	9.30	11.06
ECOOP	7.56	16.76	7.32	2.60	6.52	12.79	9.76
APSEC	2.10	6.70	2.44	5.19	7.61	9.30	5.21
ICSE	2.52	0	6.10	2.60	8.70	10.47	4.77
OOPSLA	3.78	11.73	0	5.19	0	5.81	4.77
CSMR	0.42	1.68	15.85	7.79	0	1.16	4.12
ASE	3.78	5.03	6.71	2.60	3.26	1.16	3.90
WCRE/ SANER	0	0	9.76	10.39	0	2.33	3.04
TOOLS	4.62	5.59	0.61	0	0	3.49	3.04
ICSME	1.26	0	7.32	5.19	1.09	2.33	2.60
COMPSAC	2.52	1.12	4.27	2.60	7.61	0	2.60
EuroPLoP	5.04	0	2.44	0	0	4.65	2.60
ESEM	4.20	0	0	10.39	0	1.16	2.17
Others	34.03	46.93	37.20	40.26	57.61	36.05	40.35
Total	100	100	100	100	100	100	100

In Table 19, the most popular journals in the field of DPs are shown. The Journal of Systems and Software has published the largest number of journal papers related to DPs. Journal of Object Technology, Software Practice and Experience, Software & Systems Modeling, and IEEE Transactions on Software Engineering stand in the next ranks after the Journal of Systems and Software.

TABLE 19: ACTIVE JOURNALS IN DPS FIELD

Name	Number of publications	Name	Number of publications
J SYST SOFTWARE	26	ACM T SOFTW ENG METH	2
J OBJ TECHNOL	15	AUTOMAT SOFTW ENG	2
SOFTWARE PRACT EXPER	14	COMPUT LANG SYST STR	2
SOFTW SYST MODEL	12	INT J SOFTW TOOL TECHNOL TRAN	2
IEEE T SOFTWARE ENG	11	COMPUT EDUC	2
INFORM SOFTWARE TECH	10	COMPUT SECUR	2
SCI COMPUT PROGRAM	9	FORM ASP COMPUT	2
SOFTWARE QUAL J	6	INFORM SCIENCES	2
ADV ENG SOFTW	5	ACM T PROGR LANG SYS	1
EMPIR SOFTW ENG	5	APPL SOFT COMPUT	1
COMPUT J	5	COMPUT HUM BEHAV	1
COMPUT STAND INTER	4	GOV INFORM Q	1
IEEE Software	4	IEEE T DEPEND SECURE	1
IET Software	4	IEEE T POWER SYST	1
INTERNET RES	3	IEEE T SYST MAN CY A	1
Informatica	3	INNOV SYST SOFTW ENG	1
INT J SOFTW ENG KNOW	3	INTERACT COMPUT	1
J SOFTW EVOL PROC	3	MANAG INF SYST Q	1
J VISUAL LANG COMPUT	3	PATTERN ANAL APPL	1
REQUIR ENG	3	SECUR COMMUN NETW	1

Table 20 represents the distribution of the publications in DPs topics over the 13 most active journals in the field. For example, this table shows that in the pattern development category, 14.81 percent of the journal papers are published in the Journal of Systems and Software. In this table, we have highlighted the most active journal in each topic. Note that the row entitled "Others" includes the total percentage of participation of the other journals.

TABLE 20: DISTRIBUTION OF DPS TOPICS PUBLICATIONS OVER ACTIVE JOURNALS IN THE FIELD

Topics Name	Pattern Development (%)	Pattern Usage (%)	Pattern Mining (%)	Quality Evaluation (%)	Pattern Specification (%)	Miscellaneous Issues (%)	Total (%)
J SYST SOFTWARE	14.81	24.24	20	14	6.98	0	14.77
J OBJ TECHNOL	12.35	1.52	8.57	0	16.28	14.29	8.52
SOFTWARE PRACT EXPER	12.35	6.06	5.71	4	13.95	4.76	7.95
SOFTW SYST MODEL	8.64	6.06	8.57	2	9.30	4.76	6.82
IEEE T SOFTWARE ENG	4.94	0	7.14	16	11.63	0	6.25
INFORM SOFTWARE TECH	2.47	3.03	5.71	20	0	4.76	5.68
SCI COMPUT PROGRAM	0	6.06	2.86	0	9.30	19.05	5.11
SOFTWARE QUAL J	9.88	0	2.86	0	0	4.76	3.41
ADV ENG SOFTW	2.47	3.03	4.286	0	2.33	4.76	2.84
EMPIR SOFTW ENG	2.47	0	0	12	4.65	0	2.84
COMPUT J	2.47	3.03	0	0	0	14.29	2.84
COMPUT STAND INTER	2.47	6.06	0	0	0	4.76	2.27
IEEE Software	2.47	0	0	4	0	9.52	2.27
Others	22.22	40.90	34.29	28	25.58	14.29	28.41
Total	100	100	100	100	100	100	100

5 DISCUSSION

In this section, we have discussed and summarized the main achievements of our review with respect to the research questions specified in Section 3.1.1. In Section 5.1, we have summarized the current state of the art concerning DPs research. In Section 5.2 we have discussed the findings of the demographics of patterns research. Finally, in Section 5.3, we have determined the research emphasis on each topic.

5.1 Research topics on DPs

As mentioned before, the main goal of our study is to provide a taxonomy of the research topics related to DPs. Our results suggest six main topics on DPs research, namely (a) Pattern Development, (b) Pattern Usage, (c) Pattern Mining, (d) Quality Evaluation, (e) Pattern Specification, and (f) Miscellaneous Issues.

Our new research taxonomy, compared to the previous classification scheme in this area (Ampatzoglou et al. 2013), has one additional category titled “Pattern Development”. The main reason for this difference is the scope of our study which is all the DPs that exist in the literature, while Ampatzoglou et al. (2013) only considered GoF DPs. The Pattern Development category indicates any development in the field of DPs research, including (a) introducing a new pattern or new pattern language, (b) discussing pattern variants, pattern evolution, composing patterns, or improving a specific pattern, and (c) categorizing existing DPs in various fields and domains. Papers which are included in this category are not limited only to the GoF patterns.

5.2 Demographics of DPs research

As mentioned before, the attention to the subject of DPs is increasing (see Fig. 4). The variation of publications during different years can be measured by a polynomial trend line through the data. As illustrated in Fig. 4, rapid growth is found in the number of publications from 1995 to 2010, and after that, the gradient of the fitted curve tends to zero. It should be noted that for year 2010, a significant deviation is observed from the trend line. The reason behind the remarkable deviation could be attributed to the relatively high number of published papers in QUATIC and PLoP conferences proceedings. In 2010 ten papers were published in proceedings of these two international conferences. Similarly, a considerable number of conference papers in 1998, 2001, and 2007 has had a major effect on the deviation of data from the plotted trend line.

The results of Fig. 8 suggest that most publications come from wealthy countries. Considering the absolute numbers, USA and Canada seem to be the most productive countries. However, smaller affluent countries like Germany and Italy are more productive when the numbers are normalized to population or geographical distribution of the publication. The figure elucidates the considerable investment of developed countries in the field of DPs. On the other hand, it discloses the need for more investment by developing countries in the subject.

Table 16 demonstrates that the most active researcher in the DPs field is Yann-Gaël Guéhéneuc. He is the leader of the Ptidej (The Pattern Trace Identification, Detection, and Enhancement in Java) Team, which seeks to develop theories, methods, and tools, to evaluate and to enhance the quality of object-oriented programs. To achieve this goal, one of the research areas that the Ptidej team is focusing on is DPs. Useful information about the other active researchers in Table 16 is shown in in DPs-SM →Chapter 2 →Tables 2-7.

5.3 Research emphasis on each topic

As mentioned in section 4.1, the most popular topics of DPs research are Pattern Development, Pattern Mining, and Pattern Usage. Some conferences including PLoP, EuroPLoP, and OOPSLA are annually held on the DPs which have a special focus on the Pattern Development domain. This motivates the scholars to attend such credible conferences and as a result, the number of publications in this field is higher than other fields of DPs. In addition, in the Pattern Mining field, the lack of a comprehensive algorithm to detect the patterns with an acceptable accuracy encourages the researchers to improve the present algorithms continuously. Furthermore, as seen in DPs-SM →Chapter 3 →Table 3-1 there are various domains and technologies in which DPs can be used. Since practical studies are important in all fields of science and technology, the Pattern Usage branch is no exception and it has also attracted the attention of scholars in the DPs field. These notes are the possible reasons for the higher number of publications in Pattern Development, Pattern Mining, and Pattern Usage branches compared to other fields.

6 THREATS TO VALIDITY

One of the main threats to systematic reviews is the problems that might arise during the search and the selection of the primary studies (problems which threaten the completeness of the included papers set). This threat arises from search and selection process. In this paper, we designed a search strategy which takes advantages of two search methods (manual search and snowballing). We conducted a manual search on the selected search spaces and completed our search by snowballing. So, our search strategy is reliable.

For selecting the primary studies, we designed a process in two phases. For the sake of reliability, two reviewers conducted the selection separately, and disagreements were resolved by decision rules and one other reviewer (see section 3.1.3.2). So, we have reduced the threats which may arise in the selection process (such as biases, misunderstandings, etc.).

After finishing the search process, 36 journals and 53 conferences or workshops were selected as the search spaces to find the primary studies which are in the form of paper. So, it is possible that some articles in other journals or conferences are ignored. For reducing this threat and checking the completeness of the included papers, we applied the search and selection evaluation. Our evaluation test set was generated by a separate team (for removing any biases). After applying the evaluation, 3 new journals and 16 new conferences were added to the search space set. However, as mentioned before, we selected the journals and conferences or workshops under consideration of the defined criteria (see section 3.1.3.1). So, in the current mapping study, a good set of journals and conferences or workshops is considered that can be used to find relevant papers with high quality. Furthermore, we selected "design pattern" term as the search string and searched this term in the title, abstract, and full-text of the papers (for manual search). We assumed that "design pattern" term exists in the full-text of most the papers that deal with DPs and according to the purpose of this study, it is not unexpected.

During the data extraction, some threats such as researcher biases and misunderstandings could appear. To remove these threats from our study, the data extraction forms were traced back and checked by another reviewer.

The other threat to validity is selecting a name for each category. As mentioned in Section 3.3.4, we chose a name for each category according to the most frequent keywords in it and its concept. For example, for the category of Pattern Mining, there were some keywords like "Detection", "Mining", and "Recovery". Nevertheless, we chose "Pattern Mining" as the name of the category because it is more inclusive. However, for some categories, the selected name is not within the keywords of that category. Therefore, it is possible that there was a more appropriate term for naming a category which may be ignored. In these cases, we discussed the selected names (according to the concept and keywords) in our team to remove potential biases.

7 CONCLUSION AND FUTURE WORKS

In this paper, we conducted an SMS on the field of DPs to identify some key issues including: (a) topics and sub-topics, (b) frequency of publications in each topic (c) trends and gaps, (d) active countries, researchers, and venues, and (e) common keywords and phrases. So, a total of 2775 journal and conference papers were identified (during conducting the search process and the evaluation) and 637 papers were included. Then, the data necessary to answer the research questions were extracted, and the included papers were tabulated based on these data.

The results show that the DPs field is an active and attractive research topic and in recent decades, the trend of the publications in this field is increasing. Concerning the geographical distribution of publications, USA and Canada are the most active countries in this field.

The most important contribution of this paper is proposing a classification for the research topics in the field of DPs. As shown in Fig. 6, we presented a taxonomy which includes six classes: 1- Pattern Development, 2- Pattern Usage, 3- Pattern Mining, 4- Quality Evaluation, 5- Pattern Specification, and 6- Miscellaneous Issues.

Pattern Development, Pattern Mining, and Pattern Usage are the most active topics, while Pattern Evaluation and Pattern Specification attract less attention. So, future works should consider these gaps and trends. For example, proposing a systematic review on Pattern Development, Pattern Usage, or Pattern Mining can be a valuable endeavour. Moreover, the deficiency of research with the contribution of proposing a method for the evaluation or specification of DPs is noticeable and more research is required in these topics. Concerning the active venues, PLoP and Journal of Systems and Software devoted the most publications. According to the results, publishing papers (with the focus on the DPs) in 39 ISI journals and 67 conferences with high quality indicates that DPs are an important and noteworthy field in software engineering.

APPENDIX

Note: In the following tables, due to space limitations, only information about included items are provided. See our supplementary materials available in the web (http://sqlab.um.ac.ir/images/219/files/SQLLabSM.pdf) for more information.

Table A: The Secondary Studies Used for Generating Initial Set of Our Systematic Mapping Study

Param. ID	Secondary study	Research Type	Date
1	What Do We Know about the Effectiveness of Software Design Patterns?	Systematic Literature Review	2012
2	A survey of experienced user perceptions about software design patterns	Survey	2013
3	How have we evaluated software pattern application? A systematic mapping study of research design practices	Systematic Mapping Study	2015
4	A review of design pattern mining techniques	Review	2009
5	Research state of the art on GoF design patterns: A mapping study	Systematic Mapping Study	2013
6	Evaluation of Accuracy in Design Pattern Occurrence Detection	Survey	2010
7	Architecture and Design Pattern Discovery Techniques – A Review	Review	2007
8	A Survey of Existing Approaches for Pattern Search and Selection	Survey	2010

Table B: The Initial Set of Our Systematic Mapping Study

	Title	Ref. ID*	Journal/ Conference Name	Date
1	The Factory Pattern in API Design: A Usability Evaluation	1, 2, 3, 5	International Conference on Software Engineering (ICSE)	2007
2	Do Maintainers Utilize Deployed Design Patterns Effectively?	1, 2, 3, 5	International Conference on Software Engineering (ICSE)	2007
3	A controlled experiment in maintenance comparing design patterns to simpler solutions	1, 2, 3, 5	IEEE Transactions on Software Engineering	2001
4	Two controlled experiments assessing the usefulness of design pattern documentation in program maintenance	1, 2, 3, 5	IEEE Transactions on Software Engineering	2002
5	A Controlled Experiment Comparing the Maintainability of Programs Designed with and without Design Patterns—A Replication in a Real Programming Environment	1, 2, 3, 5	Empirical Software Engineering	2004
6	Impact of the Visitor Pattern on Program Comprehension and Maintenance	1, 3	ACM/ IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)	2009
7	Cognitive learning efficiency through the use of design patterns in teaching	3	Computers & Education	2010
8	An empirical study on students' ability to comprehend design patterns	3	Computers & Education	2008
9	Documenting pattern use in java programs	1, 2, 3, 5	IEEE International Conference on Software Maintenance (ICSM)	2002
10	Human and program factors affecting the maintenance of programs with deployed design patterns	3	Information and Software Technology	2012
11	Evaluation of the Pattern-based method for Secure Development (PbSD): A controlled experiment	3	Information and Software Technology	2012
12	Distributed cognition in software design: An experimental investigation of the role of design patterns and collaboration	3	Management Information Systems Quarterly	2014
13	Instantiating and detecting design patterns: Putting bits and pieces together	4, 5, 7, 8	IEEE/ ACM International Conference on Automated Software Engineering (ASE)	2001

14	Relating expectations to automatically recovered design patterns	4, 5, 7	Working Conference on Reverse Engineering (WCRE)	2002
15	Mining design patterns from C + + source code	4, 5, 6, 7	International Conference on Software Maintenance and Evolution (ICSME)	2003
16	Automatic verification of Java design patterns	4, 5, 7	IEEE/ ACM International Conference on Automated Software Engineering (ASE)	2001
17	Design pattern recovery by visual language parsing	4, 5, 6, 7	Conference on Software Maintenance and Reengineering (CSMR)	2005
18	Case studies of visual language based design patterns recovery	4, 5, 6, 7	Conference on Software Maintenance and Reengineering (CSMR)	2006
19	A formal framework for design component contracts	4	The IEEE International Conference on Information Reuse and Integration (IRI)	2003
20	DP-Miner: Design pattern discovery using matrix	4, 7	Annual IEEE International Conference and Workshops on the. Engineering of Computer Based Systems (ECBS)	2007
21	Design pattern detection by template matching	4, 6	ACM/ SIGAPP Symposium on Applied Computing (SAC)	2008
22	Compound record clustering algorithm for design pattern detection by decision tree learning	4	The IEEE International Conference on Information Reuse and Integration (IRI)	2008
23	Visualizing design patterns in their applications and compositions	4, 5	IEEE Transactions on Software Engineering	2006
24	Classification of design pattern traits	4	International Conference on Software Engineering and Knowledge Engineering (SEKE)	2007
25	Design pattern mining enhanced by machine learning	4, 5, 7	International Conference on Software Maintenance and Evolution (ICSME)	2005
26	Fingerprinting design patterns	4, 5, 7	Working Conference on Reverse Engineering (WCRE)	2004
27	A composite design-pattern identification technique	4, 7	Informatica	2005
28	Generating design pattern detectors from pattern specifications	4, 7	IEEE/ ACM International Conference on Automated Software Engineering (ASE)	2003
29	A practical pattern recovery approach based on both structural and behavioral analysis	4, 5	Journal of Systems and Software	2005
30	Efficient identification of design patterns with bit-vector algorithm	4, 5, 6, 7	Conference on Software Maintenance and Reengineering (CSMR)	2006
31	Pattern-based reverse-engineering of design components	4, 5, 6, 7	International Conference on Software Engineering (ICSE)	1999
32	Design recovery by automated search for structural design patterns in object-oriented software	4, 5, 6, 7	Working Conference on Reverse Engineering (WCRE)	1996
33	JBOORET: an automated tool to recover OO design and source models	4, 7	IEEE Computer Society International Conference on Computers, Software & Applications (COMPSAC)	2001
34	Towards pattern based design recovery	4, 6, 7	International Conference on Software Engineering (ICSE)	2002
35	Reverse engineering of design patterns from java source code	4, 5, 6, 7	IEEE/ ACM International Conference on Automated Software Engineering (ASE)	2006

36	SPQR: Flexible automated design pattern extraction from source code	4, 6, 7	IEEE/ ACM International Conference on Automated Software Engineering (ASE)	2003
37	Searching design patterns in source code	4, 5, 7	IEEE Computer Society International Conference on Computers, Software & Applications (COMPSAC)	2005
38	Object oriented design pattern inference	4, 5, 7	International Conference on Software Maintenance and Evolution (ICSME)	1999
39	Design Pattern Detection Using Similarity Scoring	4, 5, 6, 7	IEEE Transactions on Software Engineering	2006
40	Design pattern detection in Eiffel systems	4, 5, 7	Working Conference on Reverse Engineering (WCRE)	2005
41	A pattern-based approach to structural design composition	5	IEEE Computer Society International Conference on Computers, Software & Applications (COMPSAC)	1999
42	Evaluation of object-oriented design patterns in game development	5	Information and Software Technology	2007
43	Object-oriented design patterns recovery	5	Journal of Systems and Software	2001
44	Using metrics to identify design patterns in object-oriented software	5	International Software Metrics Symposium (swmetrics)	1998
45	Measuring and improving design patterns testability	5	International Software Metrics Symposium (swmetrics)	2003
46	Formal specification of the variants and behavioural features of design patterns	5	Journal of Systems and Software	2010
47	Specifying Behavioral Features of Design Patterns in First Order Logic	5	IEEE Computer Society International Conference on Computers, Software & Applications (COMPSAC)	2008
48	Design patterns and change proneness: An examination of five evolving systems	5	International Software Metrics Symposium (swmetrics)	2003
49	A model-driven framework for representing and applying design patterns	5	IEEE Computer Society International Conference on Computers, Software & Applications (COMPSAC)	2007
50	Automated refactoring to introduce design patterns	5	International Conference on Software Engineering (ICSE)	2000
51	Design pattern recovery through visual language parsing and source code analysis	5	Journal of Systems and Software	2009
52	A two phase approach to design pattern recovery	5	Conference on Software Maintenance and Reengineering (CSMR)	2007
53	Behavioral Pattern Identification through Visual Language Parsing and Code Instrumentation	5	Conference on Software Maintenance and Reengineering (CSMR)	2009
54	Improving Behavioral Design Pattern Detection through Model Checking	5	Conference on Software Maintenance and Reengineering (CSMR)	2010
55	An empirical study of the relationships between design pattern roles and class change proneness	5	International Conference on Software Maintenance and Evolution (ICSME)	2008
56	Precise specification and automatic application of design patterns	5	IEEE/ ACM International Conference on Automated Software Engineering (ASE)	1997
57	Static and dynamic structure in design patterns	5	International Conference on Software Engineering (ICSE)	2002

58	A UML-based pattern specification technique	5	IEEE Transactions on Software Engineering	2004
59	Design Patterns and Change Proneness: A Replication Using Proprietary C# Software	5	Working Conference on Reverse Engineering (WCRE)	2009
60	DeMIMA: A Multilayered Approach for Design Pattern Identification	5, 6	IEEE Transactions on Software Engineering	2008
61	Architecture-centric software evolution by software metrics and design patterns	5	Conference on Software Maintenance and Reengineering (CSMR)	2002
62	Design pattern directed clustering for understanding open source code	5	The International Conference on Program Comprehension (ICPC)	2009
63	A quantitative approach for evaluating the quality of design patterns	5	Journal of Systems and Software	2008
64	The effects of design pattern application on metric scores	5	Journal of Systems and Software	2001
65	OO design patterns, design structure, and program changes: an industrial case study	5	International Conference on Software Maintenance and Evolution (ICSME)	2001
66	Identification of design motifs with pattern matching algorithms	5	Information and Software Technology	2010
67	Do Design Patterns Impact Software Quality Positively?	5	Conference on Software Maintenance and Reengineering (CSMR)	2008
68	Playing roles in design patterns: An empirical descriptive and analytic study	5	International Conference on Software Maintenance and Evolution (ICSME)	2009
69	A role-based meta-modeling approach to specifying design patterns	5	IEEE Computer Society International Conference on Computers, Software & Applications (COMPSAC)	2003
70	Standing on the shoulders of giants - A data fusion approach to design pattern detection	5	The International Conference on Program Comprehension (ICPC)	2009
71	Facilitating software extension with design patterns and Aspect-Oriented Programming	5	Journal of Systems and Software	2008
72	Generative design patterns	5	IEEE/ ACM International Conference on Automated Software Engineering (ASE)	2002
73	Deferring design pattern decisions and automating structural pattern changes using a design-pattern-based programming system	5	ACM Transactions on Programming Languages and Systems(TOPLAS)	2009
74	Precise modeling of design patterns in UML	5	International Conference on Software Engineering (ICSE)	2004
75	Coupling of design patterns: Common practices and their benefits	5	IEEE Computer Society International Conference on Computers, Software & Applications (COMPSAC)	2001
76	A declarative evolution framework for object-oriented design patterns	5	International Conference on Software Maintenance and Evolution (ICSME)	2001
77	Formalizing design patterns	5	International Conference on Software Engineering (ICSE)	1998
78	A static reference flow analysis to understand design pattern behavior	5, 7	Working Conference on Reverse Engineering (WCRE)	2004
79	An empirical study on the efficiency of different design pattern representations in UML class diagrams	5	Empirical Software Engineering	2010
80	Design pattern density defined	5	OOPSLA	2009
81	Design pattern recovery based on annotations	5	Advances in Engineering Software	2010

82	Evolution support by homogeneously documenting patterns, aspects and traces	5	Conference on Software Maintenance and Reengineering (CSMR)	2002
83	A controlled experiment for assessing the contribution of design pattern documentation on software maintenance	5	ACM/ IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)	2010
84	Object Oriented Design Pattern Decay: A Taxonomy	5	ACM/ IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)	2010
85	Responsibilities and rewards: Specifying design patterns	5	International Conference on Software Engineering (ICSE)	2004
86	On the role of design patterns in quality-driven re-engineering	5	Conference on Software Maintenance and Reengineering (CSMR)	2002
87	Defect frequency and design patterns: an empirical study of industrial code	5	IEEE Transactions on Software Engineering	2004
88	Assessment of Design Patterns during Software Reengineering: Lessons Learned from a Large Commercial Project	5	Conference on Software Maintenance and Reengineering (CSMR)	2001
89	Integration in component-based software development using design patterns	5	IEEE Computer Society International Conference on Computers, Software & Applications (COMPSAC)	2000
90	Tool Support for Design Pattern Recognition at Model Level	5	IEEE Computer Society International Conference on Computers, Software & Applications (COMPSAC)	2009
91	A transformational viewpoint on design patterns	5	IEEE/ ACM International Conference on Automated Software Engineering (ASE)	2000
92	An efficient tool for recovering Design Patterns from C++ Code	6	Journal of Object Technology	2006
93	Specialization patterns	6	IEEE/ ACM International Conference on Automated Software Engineering (ASE)	2000
94	Verifying Behavioral Correctness of Design Pattern Implementation	6	International Conference on Software Engineering and Knowledge Engineering (SEKE)	2008
95	A Non-Conservative Approach to Software Pattern Detection	6	The International Conference on Program Comprehension (ICPC)	2007
96	An approach for reverse engineering of design patterns	6	Software & Systems Modeling	2005
97	Towards a benchmark for evaluating design pattern miner tools	6	Conference on Software Maintenance and Reengineering (CSMR)	2008
98	From Non-Functional Requirements to Design through Patterns	8	Requirements Engineering	2001
99	From software architecture to design patterns: A case study of an NFR approach	8	International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/ Distributed Computing (SNPD)	2005
100	Selecting Security Patterns that Fulfill Security Requirements	8	International Requirements Engineering Conference (RE)	2008
101	Systematic pattern selection using pattern language grammars and design space analysis	8	Software Practice and Experience	2007

*:The ref. ID for each paper in Table B refers to the ID of the papers in Table A which the intended paper is obtained from.

Table C: Information of the Related Journals

	Journal Name	N1*	N2**	Coverage	SJR	Q (JCR 2014)	IF
1	Journal of Systems and Software	151	26	1979-2015	1.38	Q_2	1.352
2	Journal of Object Technology	23	15	2002-2014	0.4	Q_3 (SJR)	N/ A
3	Software Practice and Experience	111	14	1972-2015	0.96	Q_2	0.897
4	Software & Systems Modeling	29	12	2005-2014	0.83	Q_2	1.408
5	IEEE Transactions on Software Engineering	48	11	1975-2014	2.19	Q_1	1.614
6	Information and Software Technology	107	10	1987-2015	1.39	Q_2	1.046
7	Science of Computer Programming	52	9	1981-2015	0.8	Q_3	0.715
8	Software Quality Journal	25	6	1992-1999, 2001-2014	1.27	Q_2	1.143
9	Computer Journal	27	5	1967-2014	0.67	Q_1 (SJR)	0.787
10	Empirical Software Engineering	36	5	1996-2015	1.566	Q_1	2.161
11	Advances in Engineering Software	28	5	1980-1985, 1992-2014	0.84	Q2	1.402
12	Computer Standards & Interfaces	21	4	1985-2015	0.744	Q_3	0.879
13	IEEE Software	5	4	1984-2014	1.1	Q_2	1.053
14	IET Software	7	4	2007-2014	0.5	Q_2	N/ A
15	Informatica	9	3	1996-2014	0.42	Q_2	0.873
16	International Journal of Software Engineering and Knowledge Engineering	37	3	1996-2014	0.49	Q_4	0.362
17	Internet Research	5	3	1995-2014	1.08	Q_3	1.661
18	Journal of Visual Languages and Computing	15	3	1990-2015	0.56	Q_3	0.893
19	Requirements Engineering	13	3	1996, 1998-2002, 2005-2014	1.221	Q_3	0.882
20	ACM Transactions on Software Engineering and Methodology	24	2	1992-2014	1.51	Q2	1.17
21	Automated Software Engineering	90	2	1994-2014	1.5	Q_1	1.733
22	Computer Languages, Systems and Structures	24	2	2002-2014	0.84	Q_4	0.44
23	Computers & Education	19	2	1976-2015	2.58	Q_1	2.56
24	Computers & Security	7	2	1982-2015	1.051	Q_2	1.031
25	Formal Aspects of Computing	7	2	1989-2014	1.31	Q_3	0.806
26	Information Sciences	5	2	1968-2014	3.29	Q_1	4.038
27	International Journal on Software Tools for Technology Transfer	19	2	1997-2014	0.81	Q_2 (SJR)	N/ A
28	Journal of Software: Evolution and Process	16	3	2012-2014	0.570	Q_4	0.624
29	ACM Transactions on Programming Languages and Systems (TOPLAS)	17	1	1986-2014	2.575	Q_3	0.897
30	Applied Soft Computing	7	1	2001-2015	2.22	Q_1	2.81
31	Computers in Human Behavior	25	1	1985-2015	1.58	Q_1	2.694
32	Government Information Quarterly	1	1	1984-2014	1.203	Q_1	2.321
33	IEEE Transactions on Dependable and Secure Computing	2	1	2004-2014	1.874	Q_2	1.351
34	IEEE Transactions on Power Systems	2	1	1985-2015	3.01	Q_1	2.814
35	IEEE Transactions on Systems	2	1	2012-2014	1.21	Q_1	2.171
36	Innovations in Systems and Software Engineering	14	1	2005-2014	0,402	Q_3	1,168
37	Interacting with Computers	16	1	1989-2014	1.02	Q_2	1.268
38	Management Information Systems Quarterly	1	1	1980-2014	6.18	Q_1	2.062
39	Pattern Analysis and Applications	4	1	1998-2015	0.483	Q_4	0.646
40	Security and Communication Networks	5	1	2009-2015	0.449	Q_3	0.720

*N1: Number of papers in the first phase (Before applying the exclusion criteria)
**N2: Number of papers in the second phase (After applying the exclusion criteria)

Table D: Information of the Related Conferences and Workshops

	Name	N1	N2	Rank	Period
1	The International Conference on Pattern Languages of Programs (PLoP)	59	51	B (ERA) B3 (Qualis)	22nd
2	The European Conference on Pattern Languages of Programs (EuroPLoP)	46	12	B (ERA)	21st
3	OOPSLA	43	22	A1 (Qualis)	30th

				A (ERA) 34 (h5-index)	
4	- ACM/ IEEE International Conference on Model Driven Engineering Languages and Systems (MODELS) - Unified Modeling Language (UML)	80	4	A2 (Qualis) B (ERA) 26 (h5-index)	19th
5	- International Conference on Technology of Object-Oriented Languages and Systems (TOOLS) - International Conference on Objects, Models, Components, Patterns (TOOLS) - International Conference on Technology of Object-Oriented Languages (TOOLS)	347	14	B1 (Qualis)	50th
6	- Working Conference on Reverse Engineering (WCRE) - Software Analysis, Evolution, and Reengineering (SANER)	56	14	A2 (Qualis) B (ERA) 21 (h5-index)	23rd
7	ACM/ IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)	22	10	A (ERA) 22 (h5-index)	10th
8	ACM/ SIGAPP Symposium on Applied Computing (SAC)	17	5	A1 (Qualis) B (ERA) 35 (h5-index)	31st
9	Advanced Software Engineering & Its Applications (ASEA)	16	7	14 (h5-index)	9th
10	Annual Hawaii International Conference on System Sciences (HICSS)	55	7	A1 (Qualis) A (ERA)	50th
11	Annual IEEE International Conference and Workshops on the Engineering of Computer Based Systems (ECBS)	37	3	B1 (Qualis) B (ERA) 13 (h5-index)	20th
12	Annual IEEE International Systems Conference (SysCon)	7	2	C (ERA) 11 (h5-index)	10th
13	Annual UK Workshop on Computational Intelligence (UKCI)	3	1	N/ A	16th
14	Asia-Pacific Software Engineering Conference (APSEC)	76	24	C (ERA) 13 (h5-index)	23rd
15	Australasian Software Engineering Conference (ASWEC)	18	1	B (ERA)	24th
16	Conference of the Centre for Advanced Studies on Collaborative Research (CASCON)	8	3	B1 (Qualis)	26th
17	Conference on Software Maintenance and Reengineering (CSMR)	63	19	A2 (Qualis) C (ERA) 23 (h5-index)	18th
18	European Conference on Object-Oriented Programming (ECOOP)	169	45	A1 (Qualis) A (ERA) 25 (h5-index)	N/ A
19	Evaluation of Novel Approaches to Software Engineering (ENASE)	10	5	B4 (Qualis) B (ERA) 7 (h5-index)	12th
20	High Performance Computing (HiPC)	23	3	B1 (Qualis) A (ERA)	N/ A
21	IEEE Computer Society International Conference on Computers, Software & Applications (COMPSAC)	94	12	A2 (Qualis) B (ERA)	40th
22	IEEE Conference on Software Engineering Education and Training (CSEE&T)	16	2	C (ERA) 12 (h5-index)	29th
23	IEEE International Advance Computing Conference (IACC)	6	1	14 (h5-index)	6th
24	IEEE International Conference on Computational Science and Engineering (CSE)	28	2	B4 (Qualis) 11 (h5-index)	18th
25	IEEE International Conference on Computer Engineering and Systems (ICCES)	5	1	N/ A	10th
26	IEEE International Conference on Embedded Computing (EmbeddedCom)	1	1	N/ A	12th
27	IEEE International Workshop on Software Technology and Engineering Practice (STEP)	13	6	B3 (Qualis)	13th
28	IEEE/ ACIS International Conference on Computer and Information Science (ICIS)	29	1	C (ERA) 11 (h5-index)	15th
29	IEEE/ ACM International Conference on Automated Software Engineering (ASE)	35	18	A (ERA) 33 (h5-index)	31st
30	International Academic Mindtrek Conference	6	3	14 (h5-index)	17th
31	International Conference on Advanced Computer Theory and Engineering (ICACTE)	5	2	15 (h5-index)	10th

32	International Conference on Advanced Information Systems Engineering (CAiSE)	9	5	A2 (Qualis) A (ERA) 12 (h5-index)	28th
33	International Conference on Availability, Reliability and Security (ARES)	18	3	B1 (Qualis) B (ERA) 19 (h5-index)	11th
34	International Conference on Computational Intelligence and Software Engineering (CiSE)	12	2	C (ERA) 11 (h5-index)	7th
35	International Conference on Coordination Models and Languages (Coordination)	6	0	A (ERA)	N/ A
36	International Conference on Dependable, Autonomic and Secure Computing (DASC)	23	1	B3 (Qualis) C (ERA)	14th
37	International Conference on Engineering of Complex Computer Systems (ICECCS)	26	4	B1 (Qualis) A (ERA) 10 (h5-index)	20th
38	International Conference on Enterprise Information Systems (ICEIS)	30	3	B1 (Qualis) C (ERA) 12 (h5-index)	18th
39	International Conference on Fundamental Approaches to Software Engineering (FASE)	10	5	A2 (Qualis) B (ERA)	19th
40	International Conference on Quality Software (ICQS)	27	6	B1 (Qualis) B (ERA) 15 (h5-index)	18th
41	International Conference on Semantic Computing (ICSC)	15	2	B2 (Qualis) 18 (h5-index)	10th
42	International Conference on Software Engineering (ICSE)	68	22	A1 (Qualis) A (ERA)	39th
43	International Conference on Software Engineering Advances (ICSEA)	30	7	B3 (Qualis) C (ERA)	10th
44	International Conference on Software Engineering and Applications (ICSOFT)	50	6	B4 (Qualis) B (ERA)	11th
45	International Conference on Software Engineering and Formal Methods (SEFM)	9	2	B1 (Qualis) B (ERA) 16 (h5-index)	14th
46	International Conference on Software Engineering and Knowledge Engineering (SEKE)	53	6	B1 (Qualis) B (ERA) 11 (h5-index)	28th
47	International Conference on Software Engineering and Service Science (ICSESS)	26	7	13 (h5-index)	7th
48	International Conference on Software Engineering Research and Practice (SERP)	18	2	B2 (Qualis)	14th
49	International Conference on Software Engineering Research, Management & Applications (SERA)	17	3	B3 (Qualis) C (ERA) 10 (h5-index)	14th
50	International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/ Distributed Computing (SNPD)	14	4	C (ERA)	17th
51	International Conference on Software Maintenance and Evolution (ICSME)	13	12	N/ A	31st
52	International Conferences on Pervasive Patterns and Applications (PATTERNS)	6	3	N/ A	8th
53	International Joint Conference on Artificial Intelligence (IJCAI)	5	3	A1 (Qualis) A (ERA) 43 (h5-index)	25th
54	International Requirements Engineering Conference (RE)	12	3	23 (h5-index)	24th
55	International Software Metrics Symposium (swmetrics)	6	3	N/ A	11th
56	International Symposium on Service-Oriented System Engineering (SOSE)	5	4	B2 (Qualis)	N/ A
57	International Symposium on the Foundations of Software Engineering (FSE)	14	6	A (ERA)	24th
58	International Symposium on Theoretical Aspects of Software Engineering (TASE)	8	2	B3 (Qualis) C (ERA)	10th
59	international working conference on source code analysis and manipulation (SCAM)	20	1	B3 (Qualis) C (ERA) 13 (h5-index)	16th
60	International Workshop on Database and Expert Systems Applications (DEXA)	35	5	B1 (Qualis) B (ERA) 15 (h5-index)	26th

61	International Workshop on Security in Information Systems (WOSIS)	1	0	C (ERA)	17th
62	Panhellenic Conference on Informatics (PCI)	8	4	9 (h5-index)	20th
63	SIGPLAN-SIGSOFT workshop on Program analysis for software tools and engineering (PASTE)	3	1	B (ERA)	11th
64	Software Engineering Workshop (SEW)	13	6	B2 (Qualis) C (ERA)	36th
65	The ACM SIGPLAN conference on Systems, Programming, Languages and Applications: Software for Humanity (SPLASH)	1	1	12 (h5-index)	30th
66	The IEEE International Conference on Information Reuse and Integration (IRI)	30	7	B2 (Qualis) 11 (h5-index)	17th
67	The International Conference on Program Comprehension (ICPC)	27	4	A2 (Qualis) C (ERA) 22 (h5-index)	23th
68	The International Conference on the Quality of Information and Communications Technology (QUATIC)	12	5	B4 (Qualis)	10th

Table E: Researchers

Name	Country	Home Page	Name	Country	Home Page	Name	Country	Home Page
Yann-Gaël Guéhéneuc	Canada - France	🏠	Hong Zhu	UK	🏠	Douglas C. Schmidt	USA	🏠
Marko Zanoni	Italy	🏠	Nikolaos Tsantalis	Canada	🏠	Ian Bayley	UK	🏠
Jing Dong	USA	🏠	Giuliano (Giulio) Antoniol	Canada	🏠	Eduardo B. Fernández	USA	🏠
Ghulam Rasool	Pakistan	🏠	Giuseppe Di Lucca	Italy	🏠	Francesca Arcelli Fontana	Italy	🏠
Apostolos Ampatzoglou	Greece - Netherlands	🏠	Yajing Zhao	USA	🏠	Stefano Maggioni	Italy	🏠
Zhang Cheng	China	🏠	Patrick Mäder	Germany	🏠			

Table F: Data Extracted for the Evaluation Phase

Param. Name	Papers	Paper Info*
Yann-Gaël Guéhéneuc	An empirical study of the relationships between design pattern roles and class change proneness	Existed
	Instantiating and detecting design patterns: Putting bits and pieces together	Existed
	Using design patterns and constraints to automate the detection and correction of inter-class design defects	Existed
	Do Design Patterns Impact Software Quality Positively?	Existed
	Evaluating the impact of design pattern and anti-pattern dependencies on changes and faults	Existed
	Mining the relationship between anti-patterns dependencies and fault-proneness	Existed
	Domain matters: bringing further evidence of the relationships among anti-patterns, application domains, and quality-related metrics in java mobile apps	Existed
	A Study on the Relation between Antipatterns and the Cost of Class Unit Testing	Existed
	Ptidej: A flexible reverse engineering tool suite	Existed
	Identification of design motifs with pattern matching algorithms	Existed
	Evaluating the use of design patterns during program comprehension–experimental setting	Existed
	DEQUALITE: building design-based software quality models	Existed
	Design Pattern Application: Pure-Generative Approach vs. Conservative-Generative Approach	Existed
	A Taxonomy and a First Study of Design Pattern Defects	Existed
	Playing roles in design patterns: An empirical descriptive and analytic study	Existed
	DeMIMA: A Multilayered Approach for Design Pattern Identification	Existed
	An exploratory study of the impact of anti-patterns on class change-and fault-proneness	Existed
	From a domain analysis to the specification and detection of code and design smells	Existed
	P-MARt: Pattern-like Micro Architecture Repository	Existed
	Improving design-pattern identification: a new approach and an exploratory study	Existed
	An empirical study on the efficiency of different design pattern representations in UML class diagrams	Existed
	Meta-modeling design patterns: Application to pattern detection and code synthesis	Included
	Using explanations for design-patterns identification	Included
	A domain analysis to specify design defects and generate detection algorithms	Included
	Ptidej: Promoting patterns with patterns	Included

	Design patterns: A round-trip	Included
	Correction of high-level design defects with refactorings	Included
	A reverse engineering tool for precise class diagrams	Included
Marko Zanoni	On applying machine learning techniques for design pattern detection	Existed
	DPB: A benchmark for design pattern detection tools	Existed
	Using design pattern clues to improve the precision of design pattern detection tools	Existed
	A tool for design pattern detection and software architecture reconstruction	Existed
	Pattern detection for conceptual schema recovery in data-intensive systems	Existed
Jing Dong	UML extensions for design pattern compositions	Existed
	Design pattern detection by template matching	Existed
	Pattern-based design evolution using graph transformation	Existed
	A pattern-based approach to structural design composition	Existed
	DP-Miner: Design pattern discovery using matrix	Existed
	Ensuring structure and behavior correctness in design composition	Existed
	A model transformation approach for design pattern evolutions	Existed
	Classification of design pattern traits	Existed
	Design pattern evolutions in QVT	Existed
	A behavioral analysis and verification approach to pattern-based design composition	Existed
	Visualizing design patterns in their applications and compositions	Existed
	Extending UML to Visualize Design Patterns in Class Diagrams	Existed
	Model Checking Security Pattern Compositions	Existed
	Evolving design patterns based on model transformation	Existed
	Automated verification of security pattern compositions	Existed
	Commutability of design pattern instantiation and integration	Existed
	A Matrix-Based Approach to Recovering Design Patterns	Existed
	A Classification of Design Pattern Evolutions	Existed
	Verifying Behavioral Correctness of Design Pattern Implementation	Existed
	A formal framework for design component contracts	Existed
	XSLT-based evolutions and analyses of design patterns	Existed
	On instantiation and integration commutability of design pattern	Included
	Service oriented evolutions and analyses of design patterns	Included
	A Transformational Approach to Structural Design Assessment and Change	Included
Ghulam Rasool	Design pattern recovery based on annotations	Existed
	Towards A Catalogue of Design Patterns Variants	Existed
	Flexible design pattern detection based on feature types	Existed
Apostolos Ampatzoglou	A methodology to assess the impact of design patterns on software quality	Existed
	An empirical investigation on the reusability of design patterns and software packages	Existed
	Building and mining a repository of design pattern instances: Practical and research benefits	Existed
	The Effect of GoF Design Patterns on Stability: A Case Study	Existed
	Evaluation of object-oriented design patterns in game development	Existed
	An Empirical Study on Design Pattern Usage on Open-Source Software	Included
	An empirical investigation on the impact of design pattern application on computer game defects	Included
	Design Pattern Alternatives: What to do when a GoF pattern fails	Included
	Investigating the use of object-oriented design patterns in open-source software: A case study	Included
	A comparative study on the effectiveness of patterns in software libraries and standalone applications	Included
Zhang Cheng	Using a follow-on survey to investigate why use of the visitor, singleton and facade design patterns is controversial	Existed
	Merged Request: A New Design Pattern for Enhancing the Performance of Concurrent Access	included
Hong Zhu	Formal specification of the variants and behavioural features of design patterns	Existed
	Formalizing Design Patterns in Predicate Logic	Existed
	An algebra of design patterns	Existed
	On the Composition of Design Patterns	Existed
	Tool Support for Design Pattern Recognition at Model Level	Existed
	A formal language for the expression of pattern compositions	Existed
	Specifying Behavioral Features of Design Patterns in First Order Logic	Existed
	On the Composability of Design Patterns	Existed
Nikolaos	Design pattern detection using similarity scoring	Existed

Tsantalis	An empirical study on students' ability to comprehend design patterns	Existed
	Architectural Risk Analysis of Software Systems Based on Security Patterns	Existed
	A Novel Approach to Automated Design Pattern Detection	Included
Giuliano (Giulio) Antoniol	Using metrics to identify design patterns in object-oriented software	Existed
	Object-oriented design patterns recovery	Existed
	An empirical study of the relationships between design pattern roles and class change proneness	Duplicate
	DeMIMA: A Multilayered Approach for Design Pattern Identification	Duplicate
	Inference of object-oriented design patterns	Existed
	Playing roles in design patterns: An empirical descriptive and analytic study	Duplicate
	An exploratory study of the impact of anti-patterns on class change-and fault-proneness	Duplicate
	A Study on the Relation between Antipatterns and the Cost of Class Unit Testing	Duplicate
	Linguistic anti-patterns: what they are and how developers perceive them	Existed
	Sub-graph mining: identifying micro-architectures in evolving object-oriented software	Existed
Giuseppe Di Lucca	Recovering Interaction Design Patterns in Web Applications	Existed
	Integrating Model Driven and Model Checking to Mine Design Patterns	Existed
	Declarative design pattern-based development using aspect oriented programming	Existed
	An Aspect-Oriented Framework for Flexible Design Pattern-based Development	Existed
	Design pattern detection using a DSL-driven graph matching approach	Existed
	Mining Design Patterns in Object-Oriented Systems by a Model-Driven Approach	Existed
	A model-driven graph-matching approach for design pattern detection	Existed
	Improving Design Patterns Modularity Using Aspect Orientation	Existed
	Improving design pattern quality using aspect orientation	Existed
	Model-driven detection of Design Patterns	Existed
Yajing Zhao	Design pattern detection by template matching	Duplicate
	DP-Miner: Design pattern discovery using matrix	Duplicate
	Model Checking Security Pattern Compositions	Duplicate
	Automated verification of security pattern compositions	Duplicate
	A Matrix-Based Approach to Recovering Design Patterns	Duplicate
	Verifying Behavioral Correctness of Design Pattern Implementation	Duplicate
	XSLT-based evolutions and analyses of design patterns	Duplicate
	On instantiation and integration commutability of design pattern	Duplicate
Patrick Mäder	Design pattern recovery based on annotations	Duplicate
	Flexible design pattern detection based on feature types	Duplicate
	Pattern-based auto-completion of UML modeling activities	Existed
	Variability points and design Pattern Usage in architectural tactics	Included
Douglas C. Schmidt	Leader/followers-a design pattern for efficient multi-threaded event demultiplexing and dispatching	Existed
	Virtual component: a design pattern for memory-constrained embedded applications	Existed
	Applying patterns to improve the performance of fault-tolerant CORBA	Included
Ian Bayley	Formal specification of the variants and behavioural features of design patterns	Duplicate
	Formalizing Design Patterns in Predicate Logic	Duplicate
	On the Composition of Design Patterns	Duplicate
	Tool Support for Design Pattern Recognition at Model Level	Duplicate
	A formal language for the expression of pattern compositions	Duplicate
	Specifying Behavioral Features of Design Patterns in First Order Logic	Duplicate
	On the Composability of Design Patterns	Duplicate
Eduardo B. Fernández	Enterprise security pattern: a new type of security pattern	Existed
	Enterprise security pattern: a model-driven architecture instance	Existed
	Security patterns and requirements for Internet-based applications	Included
	Analysis of Application of Security Patterns to Build Secure Systems	Included
Francesca Arcelli Fontana	A tool for design pattern detection and software architecture reconstruction	Duplicate
	DPB: A benchmark for design pattern detection tools	Duplicate
	Using design pattern clues to improve the precision of design pattern detection tools	Duplicate
	On applying machine learning techniques for design pattern detection	Duplicate
	Metrics and antipatterns for software quality evaluation	Included
Stefano Maggioni	Understanding the relevance of micro-structures for design patterns detection	Existed
	Using design pattern clues to improve the precision of design pattern detection tools	Duplicate
	Metrics and antipatterns for software quality evaluation	Duplicate

[*] Existed: papers in the overlap between the QGS and the set of included papers.
Included: new found papers which should be included.
Duplicate: Papers which are considered before (common between two authors in the table).

REFERENCES

Aljasser, K.: Implementing design patterns as parametric aspects using ParaAJ: The case of the singleton, observer, and decorator design patterns. Computer Languages, Systems & Structures 45, 1-15 (2016)

Ampatzoglou, A., Charalampidou, S., Stamelos, I.: Research state of the art on GoF design patterns: A mapping study. Journal of Systems and Software 86(7), 1945-1964 (2013)

Ampatzoglou, A., Kritikos, A., Kakarontzas, G., Stamelos, I.: An empirical investigation on the reusability of design patterns and software packages. Journal of Systems and Software 84(12), 2265-2283 (2011)

Ampatzoglou, A., Chatzigeorgiou, A., Charalampidou, S., Avgeriou, P.: The Effect of GoF Design Patterns on Stability: A Case Study. IEEE Transactions on Software Engineering 41(8), 781-802 (2015)

Ampatzoglou, A., Michou, O., Stamelos, I.: Building and mining a repository of design pattern instances: Practical and research benefits. Entertainment Computing 4(2), 131-142 (2013)

Arksey, O'Malley, L.: Scoping studies: towards a methodological framework. International Journal of Social Research Methodology 8(1), 19-32 (2005)

Bai, J., Luo, H., Qin, F.: Design pattern modeling and extraction for CAD models. Advances in Engineering 93, 30-43 (2016)

Biolchini J., Mian, P. G., Natali, A. C., Travassos, G. H.: Systematic review in software engineering. Technical report ES67905, University of Rio de Janeiro (2005)

Blewitt, A., Bundy, A., Stark, I.: Automatic verification of Java design patterns. In: Proceedings of 16th Annual International Conference on Automated Software Engineering, pp. 324-327. IEEE, California (2001)

Bouassida, N., Ben-Abdallah, H., Issaoui, I.: Evaluation of an automated multi-phase approach for patterns discovery. International Journal of Software Engineering and Knowledge Engineering 23(10), 1367-1398 (2013)

Budgen, D., Turner, M., Brereton, P., Kitchenham, B.: Using mapping studies in software engineering. In: Proceedings of the 20th Annual Meeting of the Psychology of Programming Interest Group, pp. 195-204. Lancaster University, Lancaster (2008)

C2 Wiki: Design Patterns, http://c2.com/cgi/wiki?DesignPatterns (accessed September 17, 2016)

Cacho, N., Sant'anna, C., Figueiredo, E., Dantas, F., Garcia, A., Batista, T.: Blending design patterns with aspects: A quantitative study. Journal of Systems and Software 98, 117-139 (2014)

Dong, J.: UML Extensions for Design Pattern Compositions. Journal of Object Technology 1(5), 151-163 (2002)

Dong, J., Alencar, P., Cowan, D., Yang, S.: Composing pattern-based components and verifying correctness. Journal of Systems and Software 80(11), 1755-1769 (2007)

Dong, J., Yang, S., Zhang, K.: Visualizing design patterns in their applications and compositions. IEEE Transactions on Software Engineering 33(7), 433-453 (2007)

Dong, J., Alencar, P. S., Cowan, D. D.: A behavioral analysis and verification approach to pattern-based design composition. Software and Systems Modeling 3(4), 262-272 (2004)

Dyba, T., Kitchenham, B. A., Jørgensen, M.: Evidence-Based Software Engineering for Practitioners. IEEE Software 22(1), 58-65 (2005)

Elish, M. O., Mohammed, M. A.: Quantitative analysis of fault density in design patterns: An empirical study. Information and Software Technology 66, 58-72 (2015)

Febrero, F., Calero, C., Moraga, M.: A Systematic Mapping Study of Software Reliability Modeling. Information and Software Technology 56(8), 839-849 (2014)

Felizardo, K. R., Mendes, E., Kalinowski, M., Souza, E. F., Vijaykumar, N. L.: Using Forward Snowballing to update Systematic Reviews in Software Engineering. International Symposium on Empirical Software Engineering and Measurement (ESEM), (2016)

Fernandez, E. B., La Red, D. L., Peláez, J. I.: A conceptual approach to secure electronic elections based on patterns. Government Information Quarterly 30(1), 64-73 (2013)

Fortuin, H.: A Modern, Compact Implementation of the Parameterized Factory Design Pattern. Journal of Object Technology 9(1), 57-63 (2010)

Gaitani, M. A. G., Zafeiris, V. E., Diamantidis, N., Giakoumakis, E. A.: Automated refactoring to the Null Object design pattern. Information and Software Technology 59(C), 33-52 (2015)

Gamma, E., Helms, R., Johnson, R., Vlissides, J.: Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley, USA (1995)

Griffith, I., Izurieta, C.: Design pattern decay: the case for class grime. In: Proceedings of the 8th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement, pp. 1-4. ACM, Torino (2014)

Hafiz, M., Adamczyk, P., Johnson, R.: Organizing Security Patterns. IEEE Software 24(4), 52-60 (2007)

Hafiz, M.: A pattern language for developing privacy enhancing technologies. Software: Practice and Experience 43(7), 769-787 (2013).

Halkidis, S. T., Chatzigeorgiou, A., Stephanides, G.: A qualitative analysis of software security patterns. Computers & Security 25(5), 379-392 (2006)

Hamza, H. S.: Improving analysis patterns reuse: an ontological approach. In Proceedings of Ontologies as Software Engineering Artifacts Workshop, pp. 24-28. OOPSLA, Vancouver (2004)

Hasheminejad, S. M. H., Jalili, S.: Design patterns selection: An automatic two-phase method. Journal of Systems and Software 85, 408-424 (2012)

Jaafar, F., Guéhéneuc, Y. G., Hamel, S., Khomh, F., Zulkernine, M.: Evaluating the impact of design pattern and anti-pattern dependencies on changes and faults. Empirical Software Engineering 21(3), 896-931 (2016)

Jalali, S., Wohlin, C.: Systematic literature studies: database searches vs. backward snowballing. In: Proceedings of the ACM-IEEE international symposium on Empirical software engineering and measurement, pp. 29-38. ACM, Lund (2012)

Kerievsky, J.: Refactoring to patterns. Addison-Wesley Professional, USA (2005)

Kienzle, J., Romanovsky, A.: Framework based on design patterns for providing persistence in object-oriented programming languages. IEEE Proceedings-Software 149(3), 77-85 (2002).

Kim, D., Shen, W.: Evaluating pattern conformance of UML models: a divide-and-conquer approach and case studies. Software Quality Journal 16(3), 329-359 (2008)

Kim, S. K., Carrington, D.: A formalism to describe design patterns based on role concepts. Formal Aspects of Computing 21, 397-420 (2009)

Kitchenham, B. A.: Procedures for performing systematic reviews. Technical Report TR/SE-0401 and NICTA Technical Report 0400011T.1, Keele University (2004)

Kitchenham, B. A., Pretorius, R., Budgen, D., Brereton, O. P., Turner, M., Niazi, M., Linkman, S.: Systematic literature reviews in software engineering-A tertiary study. Information and Software Technology 52(8), 792-805 (2010)

Kitchenham, B. A., Budgen, D., Brereton, O.P.: Using Mapping Studies as the Basis for Further Research—A Participant- Observer Case Study. Information and Software Technology, special section from EASE 53(4), 638-651 (2011)

Kitchenham, B. A., Charters, S.: Guidelines for performing Systematic Literature Reviews in Software Engineering. Version 2.3, EBSE Technical Report EBSE-2007-01, Keele University and University of Durham (2007)

Lano, K., Kolahdouz-Rahimi, S.: Model-transformation design patterns. IEEE Transaction on Software Engineering 40(12), 1224-1259 (2014)

Li, Z., Paris, A., Peng, L.: A systematic mapping study on technical debt and its management. Journal of Systems and Software 101, 193-220 (2015).

Lyon, D., Castellanos, F.: The Parametric Singleton Design Pattern. Journal of Object and Technology 6(3), 13-23 (2007)

Massingill, B. L., Mattson, T. G., Sanders, B. A.: Parallel programming with a pattern language. International Journal on Software Tools for Technology Transfer 3(2), 217-234 (2001).

Najari, A., Dubois, S., Barth, M., Sonntag, M.: From Altshuller to Alexander: Towards a Bridge between Architects and Engineers. Procedia CIRP 39, 119-124 (2016).

Nilsson, E.: Design patterns for user interface for mobile applications. Advances in Engineering Software 40(12), 1318-1328 (2009)

Noyons, C.M.E., Buter, K.R., van Raan, F.J.A., Schwechheimer, H., Winterhager, M., Weingart, P.: The Role of Europe in World-Wide Science and Technology. Monitoring and Evaluation in a Context of Global Competition, (2000).

Pati, T., Hill, J.: A survey report of enhancements to the visitor software design pattern. Software: Practice and Experience 44(6), 699-733 (2012)

Pauwels, S., Hübscher, C., Bargas-Avila, J., Opwis, K.: Building an interaction design pattern language: A case study. Computers in Human Behavior 26(3), 452-463 (2010)

Petersen, K., Vakkalanka, S., Kuzniarz, L.: Guidelines for conducting systematic mapping studies in software engineering: An update. Information and Software Technology 64, 1-18 (2015)

Petersen, K., Feldt, R., Mujtaba, S., Mattsson, M.: Systematic mapping studies in software engineering. In: 12th International Conference on Evaluation and Assessment in Software Engineering, pp. 68-77. University of Bari, Bari (2008)

Porras, G. C., Guéhéneuc, Y. G.: An empirical study on the efficiency of different design pattern representations in UML class diagrams. Empirical Software Engineering 15(5), 493-522 (2010)

Riaz, M., Breaux, T., Williams, L.: How have we evaluated software pattern application? A systematic mapping study of research design practices. Information and Software Technology 65, 14-38 (2015)

Rossi, B., Russo, B.: Evolution of design patterns: a replication study. In: Proceedings of the 8th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement, pp. 38-42. ACM, Torino (2014)

Sarun, I., Weenawadee, M.: Retrieving model for design patterns. ECTI Transactions on Computer and Information Technology 3(1), 51-55 (2007)

Scanniello, G., Gravino, C., Risi, M., Tortora, G., Dodero, G.: Documenting design-pattern instances: a family of experiments on source-code comprehensibility. ACM Transactions on Software Engineering and Methodology 24(3), 1-35 (2015)

Schanz, T., Izurieta, C.: Object oriented design pattern decay: a taxonomy. In: Proceedings of the 2010 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement, pp. 1-8. ACM, Bolzano-Bozen (2010)

SCImago Journal and Country Rank. http://www.scimagojr.com (accessed August 31, 2016)

Tofan, D., Galster, M., Avgeriou, P., Schuitema, W.: Past and future of software architectural decisions–A systematic mapping study. Information and Software Technology 56(8), 850-872 (2014).

Tsai, W. T., Wu, B., Jin, Z., Huang, Y., Li, W.: Ontology patterns for service-oriented software development. Software: Practice and Experience 43(7), 867-883 (2013)

Ujhelyi, Z., Szőke, G., Horváth, A., Csiszár, N. I., Vidács, L., Varró, D., et al.: Performance comparison of query-based techniques for anti-pattern detection. Information and Software Technology 65, 147-165 (2015)

Uzunov, A. V., Fernandez, E. B., Falkner, K.: ASE: a comprehensive pattern-driven security methodology for distributed systems. Computer Standards & Interfaces 41, 112-137 (2015)

Van-Hilst, M., Fernandez, E. B., Braz, F.: A multi-dimensional classification for users of security patterns. Journal of Research and Practice in Information Technology 41(2), 89-98 (2009)

Wen-Jin, L., Ju-long, P., Kang-Jian, W.: Research on Detecting Design Pattern Variants from Source Code Based on Constraints. International Journal of Hybrid Information Technology 8(5), 63-72 (2015)

Yu, D., Zhang, Y., Chen, Z.: A comprehensive approach to the recovery of design pattern instances based on sub-patterns and method signatures. Journal of Systems and Software 103, 1-16 (2015)

Zanoni, M., Fontana, F. A., Stella, F.: On applying machine learning techniques for design pattern detection. Journal of Systems and Software 103, 102-117 (2015)

Zdun, U.: Systematic pattern selection using pattern language grammars and design space analysis. Software-Practice and Experience 37(9), 983-1016 (2007)

Zhang, C., Budgen, D.: What Do We Know about the Effectiveness of Software Design Patterns? IEEE Transactions on Software Engineering 38(5), 1213-1231 (2012)

Zhang, C., Budgen, D.: A survey of experienced user perceptions about software design patterns. Information and Software Technology 55(5), 822-835 (2013)

Zhang, H., Babar, M.A., Tell, P.: Identifying relevant studies in software engineering. Information and Software Technology 53(6), 625-637 (2011)

Zhao, L., et al.: A pattern language for designing e-business architecture. Journal of Systems and Software 81(8), 1272-1287 (2008)

Zhu, H., Bayley, I.: An algebra of design patterns. ACM Transactions on Software Engineering and Methodology 22(3), 1-35 (2013)

Abbas Rasoolzadegan has received his B.Sc. degree in Software Engineering from Aeronautical University, Tehran, Iran, in 2004. He has also received M.Sc. and Ph.D. degrees in Software Engineering from Amirkabir University of Technology, Tehran, Iran, respectively in 2007 and 2013. He is currently an Assistant Professor with the Computer Engineering Department of Ferdowsi University of Mashhad. His main research focus is on software quality engineering, especially in terms of design patterns and refactoring. For more details see the SQLab homepage: http://sqlab.um.ac.ir.

Bahareh Bafandeh Mayvan is currently Ph.D. candidate in software engineering at Ferdowsi University of Mashhad (FUM). She received her B.Sc. degree in computer engineering from FUM in 2010 and her M.Sc. degree in computer engineering from University of Tehran in 2014. Her current research efforts include software quality engineering, graph theory, design patterns, and image processing.

Zahra Ghavidel Yazdi is master student in software engineering at Ferdowsi University of Mashhad (FUM). She received her B.Sc. degree in Computer Engineering from FUM and she is currently working on software quality engineering at Software Quality Lab (SQL) in FUM.