Linnæus University
School of Computer Science, Physics and Mathematics

Degree project

Systematic Literature Review on Claims and Supporting Evidence for Self Adaptive Systems

Author: Tanvir Ahmad and M. Ashfaq Haider
Date: 2013-01-24
Subject: Computer Science
Level: Master
Course code: 5DV00E

Acknowledgements

We are heartily thankful to our supervisor Danny Weyns, whose encouragement, guidance, and full support enabled us to achieve completing the thesis, and understanding the domain. We would really like to show our gratitude to our parents, for support and encouragement to achieve this milestone. Finally we would like to thank all our friends and colleagues, who supported us during the completing of this thesis.

Abstract

Researchers and engineers have always benefited considerably from the results and guidelines presented by systematic literature reviews, conducted over the years in their respective domains of interest. Despite the large body of work on self-adaptation in recent years, we could not find systematic literature reviews that addresses the claims associated with self-adaptation and the sporting evidence for these claims in the field of self adaptive system (SAS). The domain of Self-adaptation enables the software systems to adapt to the needs of the external environment and internal dynamics. It targets different application domains under the influence of software engineering including embedded systems, e-commerce, web services, robotics, transportation etc. Also it claims to improve quality attributes of the system, which mainly includes reliability, performance and flexibility. The studies conducted, use various assessment methods to provide evidence for their research results.

We have investigated all these claims and supporting evidence for self-adaptive systems in software engineering fields. We performed a systematic literature review covering 19 leading software engineering venues, resulting in 181 studies used for data collection.

The study shows that the main claims of self-adaptation are the improvement in performance, reliability and flexibility within the system. On the other hand, the tradeoffs implied by self-adaptation have not received much attention. The evidence obtained generally from examples. We found few studies with industrial and academic collaboration however we could not find any exclusive industrial effort. From the researchers point of view, we highlighted that these areas required more consideration for future research. (i) explain research design in detail and explicitly define the limitations of the study. (iii) explicitly define the application domain. (iii) explicitly define the tradeoffs. (iv) data should be available publicly, (v) and finally more industry/academic collaboration effort would be required for obtaining high level of evidence.

Categories and Subject: {Software Engineering}{Software Design, software Architecture, Software Quality}

Keywords: Self-adaptive systems, adaptation, autonomic systems, quality concerns, systematic literature review

Contents

List of Figures

List of Tables

1 Introduction

In this chapter we start to explain the problem context and domain of the study. Then we give the some introduction about the self adaptive systems. And also discuss the contribution and objective of our study and in the last section we discuss the some related work about the field of self adaptive systems.

1.1 Problem context

Evidence based research and practice, initiated by the field of medicine, has been a source of advancement in several fields including the field of software engineering. This approach has proven to be successful, and has produced consistent positive results [31]. The goal of evidence based software engineering is described conclusively by [13]. Which is

" *To provide the means by which current best evidence from research can be integrated with practical experience and human values in the decision making process regarding the development and maintenance of software* [13]."

Evidence based research can be considered as a combination of best quality scientific studies in any domain on a specific topic or research question. Systematic literature review also referred as SLR is the main method of conducting such scientific studies [11]. The key features of an SLR are that, explicit and transparent methods should be used, it should follow a standard set of steps, it should be accountable, replicable and updateable, and should be relevant and useful. Nowadays the frequency of conducting an SLR has increased sufficiently in the field of software engineering [5]. However there is a need to put in effort to actually measure and clarify the resulted contribution and improvement of engineering complex software systems. Particularly over the last few years, we have many researchers who performed various studies in SAS e.g [19, 35, 30, 33, 29].

These studies summarize their achievements of the field. But no has performed the systematic literature review on claimed evidence for self-adaptation. As a result, there is no clear view on how self adaptation actually contributes to tackling the challenges of engineering and managing complex distributed software systems engineering. However, such insight is crucial for researchers and engineers.

We take Weyns et al. [32] work as pilot research that was presented in SEAMS 2012. The novelty of extension of this study is to apply the previous study on a broader area of software engineering fields, to see the assessment evidence in other venues of software engineering. Furthermore, a deeper analysis, additional to data and graphical representation is provided.

1.2 Self adaptive system

A fundamental definition of self-adaptive software was provided by DARPA Broad Agency Announcement on Self Adaptive Software (BAA-98-12) in December of 1997 [26]: *Self-adaptive software evaluates its own behavior and changes behavior when the evaluation indicates that it is not accomplishing what the software is intended to do, or when better functionality or performance is possible.* Another definition given by [8]: *Systems that are able to adjust their behaviour in response to their perception of the environment and the system itself.* According to theses definitions a self-adaptive system is able to operate with minimal human supervision and able to modify its behaviour according to changes in its environment. A self-adaptive system comprises two parts [32]: a managed system and a managing system. The managed system (application logic) is situated in an environment and provides some functionality to users. The managing system (adaptation logic) comprises the software to monitor the managed system and its environment and performs adaptations of the managed system when needed. Additional layers may exist that enable adaptations of underlying managing systems themselves.

Since its inception the nature functionality and complexity of software and systems have changed significantly, particularly during the advancements in the last decade. The increase in complexities makes system management a tedious task, which results in increased costs in terms of time and money. To cope with the challenge, there was a need to build a software system which had the ability to adapt at run time with the changing operation environment resources and user needs that may be unknown at design time. Such a system must be flexible, dependable, robust, recoverable, customizable, configurable and self-optimizing. The hallmarks of such complex or ultra-large-scale (ULS) systems are self-adaptation and self-organization [7] and we investigate only self-adaptation in engineering studies of self-adaptive systems. Self-adaptation has been accepted as one of the most sophisticated, and popular methods throughout the evolutionary computation community. It addresses the problems of evolutionary algorithms during the process of optimization. Self-adaptation brings capability to the system to adjust its behaviour in response to perception of the environment and ever changing internal dynamics. The study of self-adaptive systems has targeted several research areas including requirement engineering, software architecture, middleware and component-based development [7]. A decade in the past, systems that support self-adaptation were very rare in domains but today more and more systems have this requirement. Systems such as those in the e-commerce, distributed systems, fault-tolerant computing, embedded systems, multi-agent systems, artificial intelligence, robotics, knowledge based systems,mobile computing and ubiquitous computing [10, 7]. In this study we investigate in which domain most of the researchers and engineers are focused, what are the claimed concerns of these domains and which assessment approach used as supporting evidence to prove their claims when engineering self-adaptive software systems.

Self-adaptive software is expected to meet their needs in response to changes at run time. To achieve this goal, software should have certain characteristics, known as self-* properties [26, 16, 3]. One of the initial well known set of self-* properties, introduced by IBM, include eight properties [2]. Salehie and Tahvildari [26] presented with three levels of self-* properties and discuss the relationship between these characteristics and quality possible factors. They presents that self-adaptiveness and self-organizing are general properties at **general level**, which decompose large and primitive properties at two different levels. Our study only deal with self-adaptiveness. Although most of the data items of this survey can be applicable to the self-organizing property. However, that the amount of research dealing with self-organization of the system, a separate study review is required to adequately cover makes this new areas.

Major Level: At major level IBM [16] defines a set of four properties:

1. **Self-configuration:** related to maintainability,flexibility portability, and usability.

2. **Self-optimization:** (efficiency/performance) which is also called self-tuning or self-adjusting [28]

3. **Self-healing (reliability, availability):** linked to self-diagnosing [24] or self-repairing [9]. Self-diagnosing refers to diagnosing errors, faults and failures, while self-repairing focuses on recovery from them.

4. **Self-protection:** Security and it can also be linked to functionality.

Primitive Level: Self-awareness, self-monitoring, self-situated, and context-awareness are the underlying primitive properties [15, 25]. At this level primitive properties may also impact on maintainability, functionality, and portability.

The continuous effort of researchers and engineers particularly in the past one decade has resulted in a hugely grown body of work on engineering self-adaptive systems such as [8, 7]. Yuriy Brun explore the state-of-the-art in engineering self-adaptive systems and identify that the critical challenges for engineering of self-adaptive and self-managing software systems. He found that in designing self-adaptive systems must based on the feedback loops when engineering self-adaptive systems. Yuriy argues that, the ability of the system to respond to the environment in the form of self-adaptation to control his actions, is one of the most promising lines of research. Similarly Cheng et. al. identified essential views of self-adaptation: modelling dimensions, requirements, engineering, and assurances. In this study researchers also argue that, the engineering of self-adaptive software systems is a major challenge and feedback loops is a major property in self-adaptive systems. In our study we investigate the engineering studies and find out the evidence level when engineering self-adaptive software systems.

A key point in self-adaptive software is that its life cycle should not be stopped by the development and initial configuration. This cycle must be still in a suitable form to the system in order to evaluate the system and respond to changes at any time [26].

1.3 Contribution and Objectives

Our objective is to study and summarize existing research on engineering self-adaptive software systems. But also to shed light on the claimed benefits of self-adaptation and to what extent evidence exists for these claims. In particular, we aim to

1. identify the claimed benefits of self-adaptation,

2. identify the evidence for these claims.

To that end, we have performed a systematic literature review. In this study we used both automatic search with manual search of selected 19 main software engineering venues and select those studies that published in the time period 1st January 2000 to present.

The results of the systematic literature review will help us to identify the claimed benefits of self adaptation. We will assess the quality of the current research in engineering studies of self adaptive systems and could highlight the areas of lack quality such as research design and limitation. We collect the information for the claims made for self adaptations and level of evidence for these claims and identify the research areas for the future research. From the study, we derive conclusions concerning claims evidence in self-adaptive systems, and areas for future research such as software engineering design and embedded systems. All the material that was used for the study together with the extracted data is available at

http://homepage.lnu.se/staff/daweaa/SLR/CESAS/CE-SAS.htm.

1.4 Related Work

The importance of Systematic Literature Review in any research field is evident due to a number of reasons. It describes, how the proposed research is related to prior research. It helps in determination of relevance of addressed research problem and it justifies the proposed methodology. Also it discovers what knowledge exists related to the proposed research topic and increases the knowledge in the targeted research area finding gaps in published research, and generating new original ideas. Some studies have applies literature reviews in the field of self-adaptive systems, although the scope of such studies is currently limited. One such effort comes from the study on claims and supporting evidence for self-adaptive systems is the literature study by Weyns et al. [32]. The scope of this study was limited to SEAMS only, but it provides the guidelines for future studies. This thesis benefit for [32] and applies the research methods there described. It presents an overview of several key factors including quality attributes, focus area, application domains, claims and evidences. Our study is inspired mainly from Weyns et al. [27] which we referred to as a pilot study. We have taken most of the data items from the pilot study, because they proved to be successful for the purpose of the data collection and generated positive results in the acclaimed published paper. We have taken the quality attributes (i.e. reliability, availability, usability, performance and others) unchanged from the pilot study. We have also enhanced some of the data items, which include quality assessment criteria, software engineering fields and assessment methods. We have also introduced a new data item called Validation Settings, which is used to get information about the effort, whether it is a sole academic effort or sole industrial effort or a collaboration of industry and academics. This study differs from the pilot study in a way that we apply our study on broader area of software engineering field. We have also used statistical methods (i.e. regression analysis, correlation, standard deviation and mean) to analyze the resultant data after the data collection phase. Moreover, the SEAM study cannot be considered as a systematic literature review but as an informal literature survey [17]. The results of our study are generally a confirmation of results generated in the pilot study.

Patikirikorala et al. [23] performed a systematic survey on self-adaptive software systems using control engineering approaches. They investigate control methodologies in self-adaptive systems and also harvested a set of design patterns. However this survey did not investigate the evidence of self-adaptive systems. Moreover this survey is covering only 9 venues and not covering the prominent venues of software engineering fields. In our study we covering 19 venues including 4 main journals. Our study is an effort to provide a systematic overview of the current research in the field, and help engineers and researchers in the field to discover existing knowledge, and provide a base for generating new ideas and open ways for systematic literature reviews in future.

A. McCann [20] lists 10 criteria for assessing self-adaptive systems that including quality, cost, flexibility, strength, autonomy, feedback time and stability. Our study confirms that these criteria map to the main concerns of self-adaptation. However, the authors do not make any representations regarding proposed criteria of significance, with the exception of these studies is mainly looking at adaptation to improve performance [32]. Our study provides more detailed look at the importance of refining.

1.5 Report Structure

In chapter 2.1 we give an overview of the method we used in our study. We explain the main parts of the review protocol. We elaborate on the research questions, define search strategy and scope, check quality criteria. In chapter 3 summarize the data items that were collected. In chapter 4 we present the data extracted from the primary studies (i.e., the selected studies after filtering), and interpret these data answering the research questions. In chapter 5 we discusses limitations of our study. In Section 5.2 we discuss our personal reflection of the SLR. We conclude with a discussion of opportunities for future research in section 5.3.

2 Review Protocol

In this chapter we briefly discuss our review protocol. We explain here our whole research method including research questions, define research strategy, Inclusion and exclusion criteria. We also defined our sources searched, study selection process and at the end of chapter we defined quality assessment criteria that we used in our study.

2.1 Research Methodology

A (SLR) systematic literature review [17] is a well-defined approach to identify, evaluation and interpreting all relevant studies regarding a particular research question, topic area or phenomenon of interest. Usually, systematic reviews conduction is a three-step approach. Figure 2.1 shows the main steps composing the SLR process regarding the Planning, Execution, and Result Analysis [22].

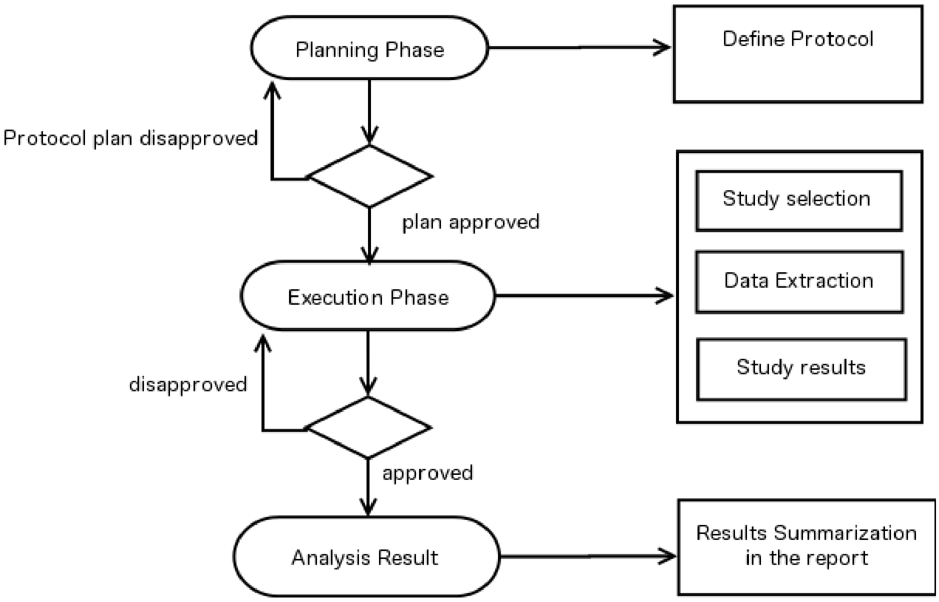


Figure 2.1: Overview of the systematic review process (adapted from [22]).

Two researchers are involved in this systematic literature review. In order to ensure a consistent planning process of our investigating systematic literature reviews and to reduce the possibility of researcher bias, we have developed a study review protocol.The main purpose of this protocol is to provide a common structure for our study review. In the planning Phase , the review protocol is defined, which includes the definition of research questions, the search strategy and scope, the data items that had to be collected, the approach for data analysis and presentation of the results. The research questions express the research topics of interest in this literature review. The scope of the review was based on the identification of the main workshops, conferences, and journals in the field. Also, we proposed our strategy for accessing the quality of studies that we considered in the review. Next, data items were identified and for each item a set of options were defined. The definition of data items was based on information derived from literature sources and from experiences with a preceding literature review [32]. For some of the data items, additional options were introduced during the review process, in particular for the fields of software engineering and validation setting. The protocol was cross-checked by an external reviewer and the feedback was used to make small improvements. Finally, the data derived from the primary studies was collected and summarized to answer the research questions.

The final report was checked by one independent researcher. His feedback was used to improve the description and correct minor issues.

2.2 Research Questions

We aimed at research questions meaningful not only to researcher but also to engineers. Thus, we first formulated the review goal through Goal-Question-Metric (GQM) perspectives (purpose, issue, object, viewpoint) [4]:

Purpose: Analyse and characterize

Issue: Claims and Evidence

Object: for self-adaptive software systems

Viewpoint: from a researcher's and engineer's viewpoint.

The general research question translates to three concrete research questions:

RQ1: What are the claims made for self-adaptation?

RQ2: What are the tradeoffs implied by self-adaptation?

RQ3: How much evidence is available for the claims and what are the types of evidence?

The goal of RQ1 is motivated by the need to get clear understanding of the claimed benefits of self adaptation and current state-of-the-art in self-adaptive systems. We are interested in identifying that how many quality attributes addressed in self adaptive systems and the domains of application in which self-adaptation has been applied. RQ1 is important for researchers and engineers to get an overview on quality attributes of the self-adaptive systems. RQ2 is to understand about what are the tradeoffs by applying self-adaptation.We investigate in this question which quality attributes have positive or negative influenced on self-adaptation, and whether or not the researchers clearly defined tradeoffs of software quality. With RQ3 we aim to investigate what assessment method have been used for evidence and how much evidence is available for applied methods. By evidence we mean evidence for adoption of methods of researcher that they used to prove their claims. In this research question we also investigate what assessment approaches can be useful for increasing the level of evidence. We also assessed the quality of the study. From the researcher and engineers point of view RQ3 give us an overview of evidence and what purposed method could be useful in practice.

2.3 Search Strategy

Our search strategy combines automatic with manual search. Our search comprised of three steps: In a first step we searched the studies by automatic method and in 2nd step manually we selected the studies that are relevant for self-adaptive systems, and then we filtered the engineering studies that have any evidence. We used the following search string in the first step:

> ((Title:adaptive OR Title:adaptation OR Title:self OR Title:autonomic
>
> OR Title:autonomous) OR
>
> (Abstract:adaptive OR Abstract:adaptation OR Abstract:self OR Abstract:autonomic OR Abstract:autonomous))

To ensure the validity of the search string we applied pilot searches on the set of studies from three specialized venues TAAS, ICAC, and SEAMS to ensure that the keywords provide the right scope of studies.

One researcher is responsible for the automatic search. Automated search performed by executing search string on five electronic data search engines, these are: IEEE Explore, ACM Digital library, Science Direct, Think Mind(TM) Digital library. Manual search refers to performed manually browsing Conference proceedings or Journals proceedings. All search were based on title and abstract.

We further refined the studies resulting from automatic search using a manual search step. The goal of this step is to identify the primary studies that are directly related to the research questions.Manual search was performed independently by both reviewers and disagreements were resolved by discussion. To that end, we defined the following inclusion/exclusion criteria

2.4 Inclusion Criteria

We limited our literature search over two dimensions: time (publication period) and space (publication venues).

- Studies which were published between 1st of January 2000 to present. We decided this broader period due to the reason that self-adaptive systems have become subject of active research around that time.

- The study must be related to the field of self-adaptive systems that must addressed adaptation logic in which after engineering self-adaptive software systems performs adaptations of the managed system when needed.

- We included only those studies that provide any evidence and assessment method to prove their claims. Assessment are validation approaches that may be in the form of example application, simulation, rigorous analysis, empirical, real world example or studies that provided some qualitative, textual, opinion-oriented evaluation. E.g. compare and contrast, oral discussion of advantages and disadvantages. Evidence may be in the form of toy example, observations, experiments, empirical or industrial evidence in Example application Authors describing an application and provide an example to assist in the description, but the example is "used to validate" or "evaluate" as far as the authors suggest [27].

2.5 Exclusion Criteria

- We excluded those studies which are related to theoretical such as new theory about self-adaptive system, discussion on new algorithm and do not provide any evidence about their theory because, we are only interested in those studies that have certain level of evidence.

- We also excluded tutorials, short papers, editorials etc. because these papers do not provide a reasonable information.

A study selected when it met the inclusion criterion and eliminated if it met any of the exclusion criterion.

2.6 Sources searched

To ensure a minimum level of quality of studies, we include the primary venues for publishing research results on self-adaptive systems, as well as the major conferences and journals on software engineering. The selected venues are listed in Table 1. The Rank is based on the evaluation published by the http://www.arc.gov.au/era/era_2010/archive/era_journal_list.htm. Australian Research Council. And those venues are not ranked by Australian Research Council, we put there n/a which means not applicable ranked.

Table 1: Searched venues

ID	Venue	Rank
ASE	International Conference on Automated Software Engineering	A
Adaptive	Adaptive and Self-adaptive Systems and Applications	n/a
DEAS	Design and Evolution of Autonomic Application Software	n/a
ICAC	International Conference on Autonomic Computing	B
ICSE	International Conference on Software Engineering	A
ICSM	International Conference on Software Maintenance	A
ISSTA	International Symposium on Software Testing and Analysis	A
SASO	Self-Adaptive and Self-Organizing Systems	n/a
SEAMS	Software Engineering for Adaptive and Self-Managing Systems	n/a
SefSAS	Software Engineering for Self-Adaptive Systems	n/a
WICSA	Working International Conference on Software Architecture	A
WOSS	Workshop on Self-Healing	n/a
WADS	Workshop on Architecting Dependable Systems	n/a
ISARCS	International Symposium on Architecting Critical Systems	n/a
FSE	Foundations of Software Engineering	A
JSS	Journal of Systems and Software	A
TAAS	Transactions on Autonomous and Adaptive Systems	n/a
TOSEM	Transactions on Software Engineering and Methodology	A*
TSE	Transactions on Software Engineering	A*

We have selected these venues because studies of self adaptation become popular with these venues (e.g. TAAS, ICAC, SASO and SEAMS). Most of the publications in self adaptive systems are presented in these specialized venues [6]. We also included the highly reputable software engineering venues (e.g. TOSEM, TSE, JSS, ICSE, FSE and ASE) because we aim at gaining solid information about claims and thus ignored lower-quality sources.

2.7 Study Selection Process

The main purpose of study selection process was to identify the relevant studies. The search string was used on the search engines to find the related studies from the selected sources in Table 1. Filtering of the studies selection process on each stage are described fully in figure 2.2.

At stage 0 all the venues were automatically searched by search string and downloaded all of the retrieved studies. After that at stage 1 both of researchers collaboratively manually checked the title , keywords and abstract of all the retrieved studies, discarding if any study did not meet the selection criteria. It may be possible that at stage 1 we also added all of those studies that were doubtful weather is included or excluded. Brereton et al. argue that abstracts might be too poor to rely on when selecting studies [5]. Thus at stage 2 we also decided about study inclusion based on the conclusions of studies and also as well as review the introduction, if needed. Then we obtained the full printing copies of all remaining studies. Both of the researchers individually taken the data extraction of each study and then both compare their results, a decision for each study was taken based on consensus.

By reading the details of the full text studies, more studies excluded, which were not relevant according to inclusion/exclusion criteria. Finally remaining studies were selected for data extraction that used in this study.

2.8 Quality assessment

Assessing the quality of the paper or its contribution is important for data synthesis and interpretation of results later on. As all studies were assessed through a quality check, To assess the quality, we collected a set of quality items as show on Table 2. These items are based on the assess method for research studies proposed in [12]. From the answers, a quality assessment score (max 12) is calculated by summing up the scores for all the questions for a study (scores for the various options are given between brackets).

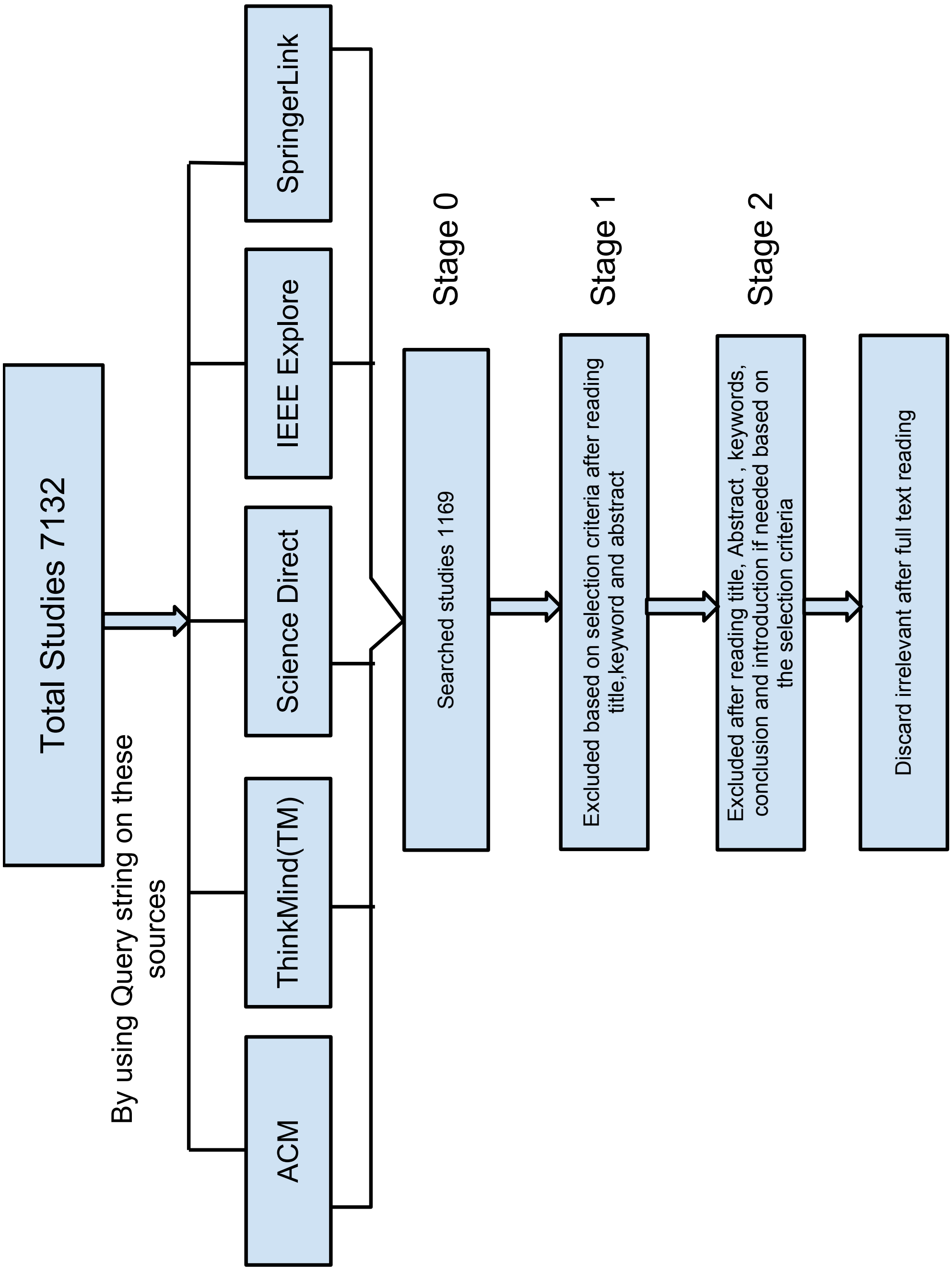

Total Studies 7132
By using Query string on these sources
SpringerLink
IEEE Explore
Science Direct
ThinkMind(TM)
ACM
Stage 0
Searched studies 1169
Stage 1
Excluded based on selection criteria after reading title,keyword and abstract
Stage 2
Excluded after reading title, Abstract , keywords, conclusion and introduction if needed based on the selection criteria
Discard irrelevant after full text reading

Figure 2.2: Study Selection

Table 2: Items to assess study quality

Quality item
1. Problem definition of the study. Options are: - (2) The authors provide an explicit problem description for the study. - (1) The authors provide a general problem description. - (0) There is no problem description.
2.1 Problem context of the study. Options are: - (1) If there is an explicit problem description for the research on the study, this problem description is supported by references. - (0.5) If there is a general problem description, this problem description is supported by references. - (0) There is no description of the problem context of the study.
2.2. Environment in which the study was carried out. Options are: - (1) The authors provide an explicit description of the environment in which this research was performed (e.g., lab setting, as part of a project, in collaboration with industry, etc.). - (0.5) The authors provide some general words about the environment in which this research was performed. - (0) There is no description of the environment.
3. Research design of the study refers to the way the study was organized. Options are: - (2) The authors explicitly describe the plan (different steps, timing, etc.) they have used to perform the research, or the way the research was organized. - (1) The authors provide some general words about the research plan or the way the research was organized. - (0) There is no description of how the research was planned/organized.
4. Contributions of the study refers to the study results. Options are: - (2) The authors explicitly list the contributions/results of the study. - (1) The authors provide some general words about the study results. - (0) There is no description of the research results.
5. Insights derived from the study. Options are: - (2) The authors explicitly list insights/lessons learned from the study. - (1) The authors provide some general words about insights/lessons learned from the study. - (0) There is no description of the insights derived from the study.
6. Limitations of the study. Options are: - (2) The authors explicitly list the limitations/problems with the study. - (1) The authors provide some general words about limitations/problems with the study. - (0) There is no description of the limitations of the study.

3 Data Items

In this chapter we start to presents the data items that we used in our study for review the papers. Next, we give a brief introduction for each data items.

Data items that we used to review each research paper. We will maintain the data in the forms, which will be specially prepared for recording the data extraction. Data extraction form should be the following data items shown in Table 3. We selected these data items except F15 from the SEAMS study Weyns et al. [32].

Table 3: Data collection form

Item ID	Field	Concern / research question
F1	Author(s)	Documentation
F2	Year	Documentation
F3	Title	Documentation
F4	Venue	Documentation
F5	Citation count	Documentation
F6	Quality score	RQ1-3
F7	Subject of the study	RQ1
F8	Application domain (if applicable)	RQ1
F9	Quality Concerns	RQ1-2
F10	Claimed benefits	RQ1-2
F11	Tradeoffs	RQ2
F12	Assessment / validation approach	RQ3
F13	Evidence level	RQ3
F14	Repeatability	RQ3
F15	Setting of validation.	RQ3

The data items author(s), year, title, venue, citation count (F1-F5) were used for documentation. Quality score (F6) is used for assessing the quality of study that described in section 6.

3.1 F7. Subject of the study

Its mean which of the software engineering field addressed in the study (F7). Software engineering can be divided into ten subdisciplines. They are: [1]

- Software requirements:
- Software design:
- Software construction:
- Software testing:
- Software maintenance:
- Software configuration management:
- Software engineering management:
- Software engineering process:
- Software engineering tools and methods:
- Software quality:
- other.

3.2 F8. Application Domains

Applications domain for which self-adaptation is used in the study. We started from the application domain taken from the study Weyns et al. [32]. After reviewing the studies we realized some more application domains. Possible application domains are:

- e-commerce(business application e.g. online book store system)
- web services
- embedded systems
- robotic systems
- traffic and transportation
- information systems
- other.

3.3 F9. Quality Concerns

Quality concerns related to self-adaptation. Formulation that are (positively or negatively) affected by self-adaptation in the study. The options (based on IEEE 9126 and ISO/IEC 25012) are:

- reliability (fault tolerance, recoverability): capability of software to maintain its level of performance under stated conditions for a stated period of time
- availability: the degree to which the software is in a functioning condition, i.e. capable to perform its intended functions
- usability (ease of learning, communicativeness): effort needed to use the system
- efficiency/performance (time behaviour, resource utilization): efficiency of the software by using the appropriate resources under stated conditions and in a specific context of use
- maintainability (analyzability, changeability, stability, testability): effort needed to make specified modifications.
- portability: ability of software to be transferred from one environment to another
- security: ability of the system to protect against misuse
- accuracy: the extent to which the software realizes the intended behavior in a specific context of use
- flexibility in use: capability of the software to provide quality in the widest range of contexts of use, incl. dealing with unanticipated change and uncertainty
- other.

3.4 F10. Claimed benefits of self-adaptation

Claimed benefits of self-adaptation can be one or more of the following (add the appropriate concerns selected in F9):

- preserving quality of the software (i.e., self-adaptation does not add quality to the system but maintains some quality attributes)

- improving quality of the software (i.e., self-adaptation add some quality to the system, i.e., it improves some quality attributes)
- assuring quality of the software (i.e., self-adaptation guarantees some quality attributes, typically by means of strong evidence or formal proof)
- improving other concerns

3.5 F11. Tradeoffs

Tradeoffs (F11) refers to the concerns of self-adaptation with a negative impact. This can be one or more of the following (add the appropriate concerns selected in F9):

- quality concerns that are negatively influenced due to self-adaptation (add the appropriate quality concerns selected in F10)
- other concerns that are negatively influenced due to self-adaptation (add the appropriate quality concerns selected in F10)

3.6 F12. Assessment / validation approach

Assessment / validation approach (F12) has the following options:(taken from the the study Weyns et al. [32])

- discussion (the authors provide a qualitative, textual, opinion-oriented evaluation; e.g., compare and contrast, oral discussion of advantages and disadvantages; the discussion may include the description of examples, but these examples must provide any evidence or results may be in the form of observations, toy example or industrial evidence etc.)
- example application (the authors use a concrete application or applications to assist the assessment of their work; the example application is either concretely realized, but not necessarily completely described in the study, or the application is used to validate or evaluate the work as far as the authors suggest).
- simulation (the authors access their work by the execution of a system with artificial data, using a model of the real world; the evaluation typically compares the work with a baseline approach by means of numerical results; this comparison may (or may not) use and existing benchmark)

- rigorous analysis (the authors employ a rigorous derivation and proof for assessment; rigorous analysis is typically based on formal methods)

- empirical study (in an empirical study, assessment is based on observation or experience. An empirical study includes clearly defined research questions/hypothesis that typically include statements about the proposed research approach in comparison with a baseline approach. The questions/hypothesis are tested with a suitable experiment. In the experiment, data is collected from both the proposed approach and the baseline approach. This data can be based on observation of the behaviour of humans or software. Statistical analysis of the collected data is used to test the hypotheses and provide evidence or not. A discussion of treats to validity is part of an empirical study. Concrete examples of empirical studies are case study, controlled laboratory experiment, and controlled experiment performed in industry setting.)

- experience from real examples (the results have been used on real-world examples, but not in the form of empirical studies; the evidence of its use is collected informally or formally)

3.7 F13. Evidence level

Evidence level is one of the following options:(taken from the the study Weyns et al. [32])

- evidence obtained from demonstration or application to simple/toy examples

- evidence obtained from expert opinions or observations

- evidence obtained from empirical studies (e.g., controlled lab experiments, causal case studies, etc.)

- industrial evidence

- other.

3.8 F14. Repeatability

Repeatability of the study is one of the following options:(taken from the the study Weyns et al. [32])

- study is not repeatable (no useful description of material is available to repeat the study)

- a partial description is available to repeat the study (may be described in the paper)
- the material to repeat the study is partially available (this typically includes links to material that is used to repeat the study)
- all the material is available to repeat the study (the study can be repeated with reasonable effort, probably in a different but similar setting)
- other.

3.9 F15. Validation setting

Validation setting is one of the following options:(taken from the the study [6])

- Academic effort
- Academic/industry collaboration
- Industrial effort
- other

4 Results Analysis

In this chapter, we first give the number of studies that included in the review and also discuss complete number of studies on each stage. Next, we presents the results of our research questions and discuss them briefly one by one. For each research question, we give a graphical overview and also we analyse of our data by statistically. At the end of each research question, we give a summary of that research question, in which we discuss that what we learned from this research question.

4.1 Included studies

Figure. 4.1 shows the number of studies that searched on each stage with respect to their venue. From the studies in total 7132 of 19 venues were searched and 1169 studies were selected after applying search string. We looked within the 1169 studies and selected 580 studies at 1st stage. At the stage 2nd we looked at 580 studies and resulted in selecting 290 studies for the primary review. And during the data extraction phase more studies were removed due to irrelevant content. This lead to 181 studies for the final review. Figure. 4.1 shows number of studies per venue that searched on each staged.

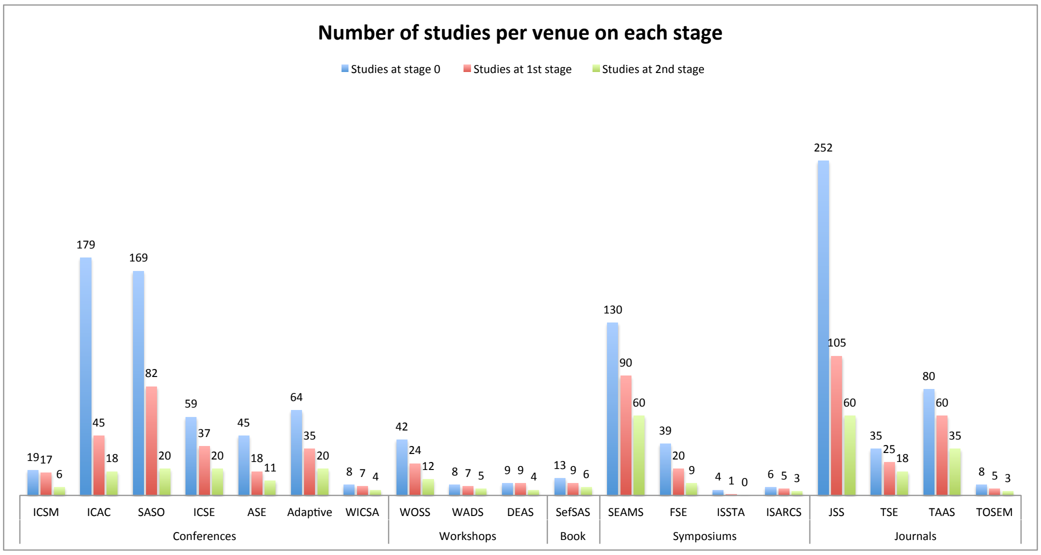
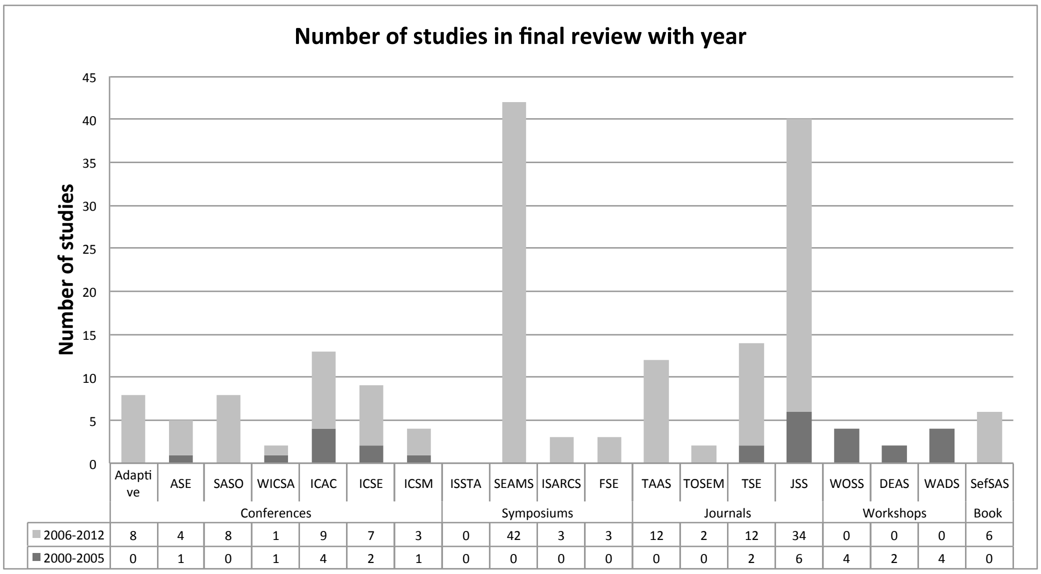


Figure 4.1: Number of studies per venue on each stage

Figure. 4.2 shows the number of included studies in final review. This figure give us an overview of the studies according to publication venue over time, which clearly shows that more than half of the studies presented in Journals and SEAMS which is 60,7% of total studies. We can see JSS and SEAMS are most prominent venues to publishing engineering studies of self adaptive systems. Most of the studies were published in SEAMS which are 23,02% while 22,09% studies appeared in JSS, 7,73% in TSE, 6,62% in TAAS and 1,1% in TOSEM. The top software engineering venues ICSM, ICAC, SASO, FSE, ICSE, Adaptive and ASE represent 27.62% of the studies.



	Adaptive	ASE	SASO	WICSA	ICAC	ICSE	ICSM	ISSTA	SEAMS	ISARCS	FSE	TAAS	TOSEM	TSE	JSS	WOSS	DEAS	WADS	SefSAS
	Conferences							Symposiums				Journals				Workshops			Book
2006-2012	8	4	8	1	9	7	3	0	42	3	3	12	2	12	34	0	0	0	6
2000-2005	0	1	0	1	4	2	1	0	0	0	0	0	0	2	6	4	2	4	0

Figure 4.2: Year wise total number of studies per venue that have evidence

This figure also shows increasing trend for publication research venues over year. Note that a significant increment can be seen after 2005 and specific attention in engineering of self adaptation studies increased during 2006 to 2012 years, which is obviously connected with the creation of ISARCS, SASO, Adaptive, ICAC, SEAMS, and TAAS around at that time.

We also looked at relative numbers. Between 2000 and 2005 on average 12% of the total number of studies focused on self- adaptation, however this number increased to 19% in the period between 2006 - 2012. Within the studies that focus on self-adaptation, between 2000 and 2005, on average, 8% engineering studies, while this number increased to 18% between 2006 and 2012.

We also applied the statistical analysis on the total number of studies with year and we find out that between 2000 and 2005 on average "2 to 7" , while between 2006 and 2012 it increase to on average "14 to 30" engineering studies of self adaptive studies were published per year. Its correlation is

0,93 which mean the number of studies increasing with respect to year, so number of studies have strong relationship with year.

We learn from this that most of the researchers published their research at the specialized venues(SEAMS, ICAC and TAAS etc), and not at the highly reputable software engineering venues susch as TOSEM , TSE and ICSE etc. It also be noted that none of study from ISSTA included in the final review. As we notice that SEAMS and JSS are most attractive venues for self-adaptive systems research. We also note that most of the empirical studies on self adaptive systems were presented in JSS. Moreover ICAC and TSE are the only venues that have high level of evidence "real world example". We also note that the evidence level of journals was high as compare to conferences and workshops.

4.2 RQ1: What are the claims made for self-adaptation?

In this section we discuss the results based on the Research Question(RQ)1 that we defined in section A.4.

Claims in self-adaptation is derived from these fields of data extracted form: software engineering field(F7), application domain(F8), quality Concerns(F9).

Figure. 4.3 give us an overview of software engineering fields. According to this figure , the software design and software quality were the most popular areas where self-adaptive systems developed using software engineering with supporting evidence of claims. They included 72 studies in software design and 27 in software quality. Together they account of around about two third of the total studies which is 77%. Software design was more important field as compared to software testing and software requirements. the reason behind it may be due to that software design has become at the beginning of software development life-cycle (SDLC). Architecture generation is the 1st step in software design and a good design can strongly effect the software development in term of quality concerns, therefore more researcher are focused on the software design field. We merge the architecture studies in the software design field.

Software quality was the second most prominent field in the software engineering fields. We noticed that the main focus of the researcher in software quality in the context of software engineering are on software functional quality and software structural quality. In functional software quality confirms the software design and with structural quality how it meets non-functional requirements such as robustness, efficiency, security etc. Remaining 23% studies were focused on other software engineering fields.

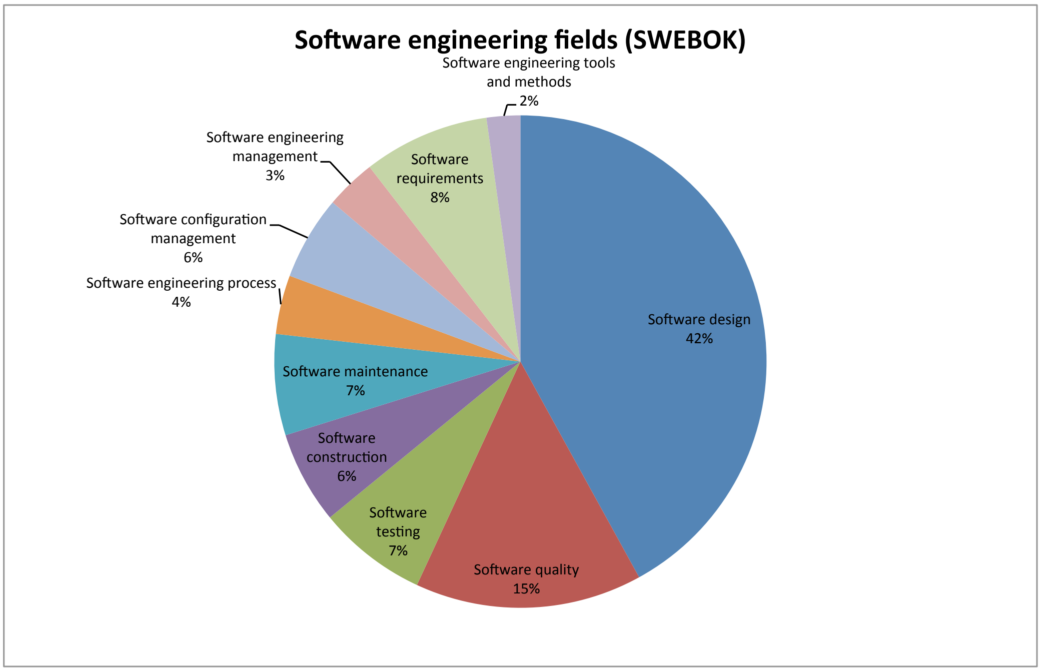
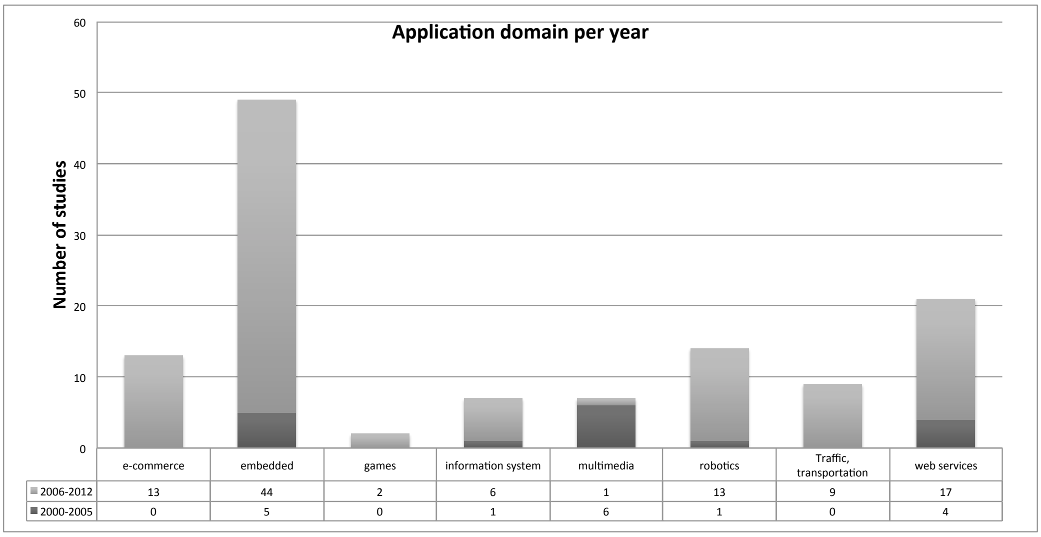


Figure 4.3: Software engineering subdisciplines

4.2.1 Classification of studies according to the software engineering fields and years

Not surprisingly, that a few studies about fields of the software engineering had been published untill 2005. Up to this date most of the studies were reported in software design and software quality, while few studies found in software construction and software requirements. Most of the studies on software engineering fields take the specific attention after 2005. We also note that there is little interest in Software Engineering tools and methods field since 2005, while growing interest in software testing.

We also looked at the application domains for which self-adaptation has been used. Figure. 4.4 shows the data extracted from the studies. Embedded systems account for 27,6% of the applications and this portion of application domain increased between 2006 and 2011. Web service is a very active area for self-adaptation and accounts of 11% of the studies. Robotics, which has always been an important domain in self-adaptation research, accounts for 7,7%. Remaining studies found in these application domain: e-commerce (6,6%), multimedia (3,8%), information system (3,8%), traffic, transportation (4,97%) and games (1,1%). We find out each study in only one domain. However a large number of studies do not explained any application domain which account for 33.2% of the total studies.



	e-commerce	embedded	games	information system	multimedia	robotics	Traffic, transportation	web services
■ 2006-2012	13	44	2	6	1	13	9	17
■ 2000-2005	0	5	0	1	6	1	0	4

Figure 4.4: Application domain per year

4.2.2 Statistical analysis of application domains.

On the bases of our previous absolute values we undertook regression analysis to find out the correlation between application domains and total number of primary studies each year. Results of our analysis are shown in the following Table 4 (where R^2 is Coefficient of Determination, R is Correlation and S.D. is Standard Derivation).

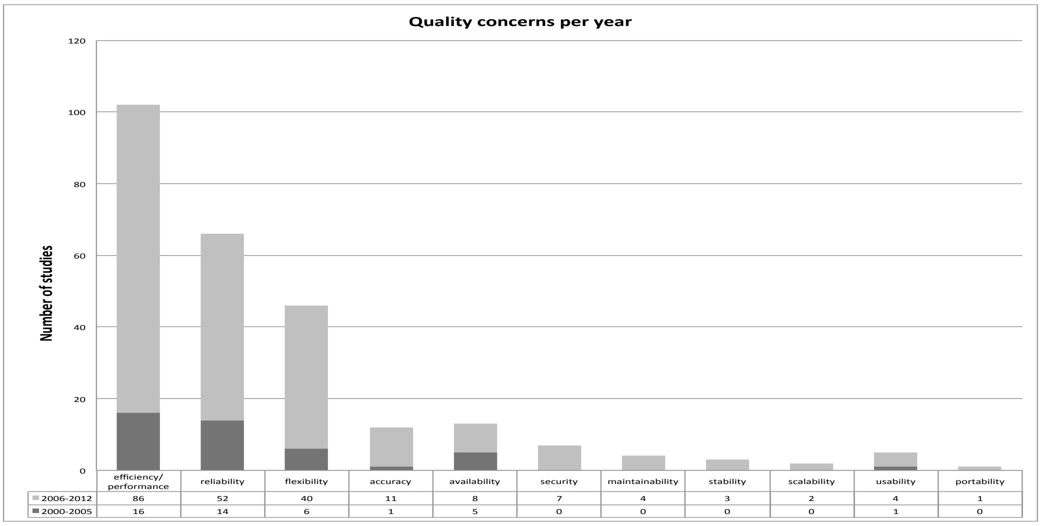
Table 4: Regression Analysis of application domains

Domains	Regression Eq.	R^2	R	Mean	S.D.
E-commerce	y = 0,1209x - 0,683	0,8	0,89	1	1,47
Embedded	y = 0,2395x + 0,4349	0,62	0,79	3,77	3,32
Games	y = 0,0205x - 0,1314	0,35	0,59	0,15	0,38
Information system	y = 0,0461x - 0,1028	0,58	0,76	0,54	0,66
Multimedia	y = -0,027x + 0,9148	0,14	-0,38	0,54	0,78
Robotics	y = 0,1132x - 0,4992	0,63	0,79	1,08	1,55
Traffic, transportation	y = 0,0673x - 0,244	0,38	0,62	0,69	1,18
Web services	y = 0,1087x + 0,1025	0,73	0,85	1,62	1,39

This table show us that e-commerce followed by web services have stronger relationship with a number of primary studies. The reason for this strong relationship that e-commerce started taking attention from 2008 and continuously it increasing up to 2012. The reason of weakness of robotics domain is between 2000 and 2007 only single study reported on robotics domain and it also takes the attention from 2008. Web service systems have gained an increased attention during the last five years. It has also been noted that the relationship of web services with a number of primary studies are stronger than embedded systems. These relationships also confirm the results of the SEAMS study [32] where we found that web based systems were the most prominent domain. It should be noted that multimedia domain have negative relationship with number of primary studies the reason for this specific attention for multimedia has significantly decreased after 2005. We also notice that there is little interest in games since 2006, while there is a growing interest in Information systems.

4.2.3 Classification of studies according to the quality concerns and year

Figure. 4.5 shows the distribution of types of quality concerns for which have been used in self adaptive systems.



	efficiency/ performance	reliability	flexibility	accuracy	availability	security	maintainability	stability	scalability	usability	portability
▨ 2006-2012	86	52	40	11	8	7	4	3	2	4	1
■ 2000-2005	16	14	6	1	5	0	0	0	0	1	0

Figure 4.5: Quality Concerns per year

The distribution of the concerns of self-adaptation in this study confirms the distribution we found in a previous study [32]. Top quality concerns of self-adaptation for which engineering is used are efficiency/performance

(56%), reliability (38%) and flexibility (24%). accuracy and availability also have minor attention which are 13% of the total studies. However under represented are stability, maintainability, scalability, portability, usability, and security that all together make up only 12% of the reported concerns. The remaining 5% of other reported concerns include complexity, engineering effort, stabilization, cost and dependability.

From Figure. 4.5 we can clearly see that there is no concern regarding security, maintainability, stability, scalability and portability between 2000 and 2005, we also notice that there is little interest in these concerns after 2005. Also there is a spectacular increase of interest in accuracy after 2005. We also noted that security and stability are the quality concerns that only managed in the embedded system domain. In case of web service domain, another interesting observation was only four quality concerns (efficiency/performance, reliability, flexibility and availability) were focused in this domain.

To confirm our results we undertook the statistical analysis by using regression analysis and standard derivation between the quality concerns and total number of primary studies in Table 5.

Table 5: Regression Analysis

Quality Concerns	Regression Eq.	R^2	R	Mean	S.D.
Performance	y = 0,5846x - 0,2933	0,93	0,96	7,85	6,61
Reliability	y = 0,2812x + 1,1622	0,79	0,89	5,08	3,45
Flexibility	y = 0,2941x - 0,633	0,58	0,79	3,54	4,20
Accuracy	y = 0,0639x + 0,0334	0,28	0,53	0,92	1,32
Availability	y = 0,033x + 0,5401	0,26	0,51	1,00	0,71
Security	y = 0,0376x + 0,0146	0,39	0,62	0,54	0,66
Maintainability	y = 0,0361x - 0,1943	0,39	0,62	0,31	0,63
Stability	y = 0,0276x - 0,1531	0,25	0,50	0,23	0,60
Scalability	y = 0,0177x - 0,0923	0,26	0,51	0,15	0,38
Usability	y = 0,0066x + 0,2928	0,02	0,14	0,38	0,51
Portability	y = -0,0006x + 0,086	0,00	-0,03	0,08	0,28

From Table 5 we can clearly see that performance has a strong relationship with the total number of studies followed by reliability and flexibility. Mean and S.D. shows that on average performance, reliability and flexibility are main concerns.

4.2.4 Comparison of correlation between application domain and quality concerns.

The aim of this comparison is to find out the relationship between application domain and quality concern, results of the correlation are shown in the following Table 6. We took the top three quality concerns (efficiency, reliability and flexibility) and found that the relationship of those concerned with the most prominent application domains(embedded, web services, robotics and e-commerce).

Table 6: Correlation comparison b/w most prominent domains and concerns

Application Domains	Efficiency/ Performance	Reliability	Flexibility
Embedded	0,92	0,67	0,69
Web services	0,82	0,83	0,60
Robotics	0,79	0,94	0,98
E-commerce	0,88	0,38	0,76

We can clearly see that efficiency/performance have strong relationship in most of the domains but the noticeable thing is that in the robotics domain there is a stronger relationship between flexibility then efficiency and reliability. We also notice that e-commerce domain also has the 2nd strongest relationship after the efficiency/performance but reliability has a weak relationship with e-commerce domain. These relationship also give us an observation about why flexibility was suddenly decreased in 2010 and 2012. We notice that flexibility starts increasing in 2008. One reasons of its increase is that studies of self adaptive systems start taking more attention after 2007, the second reason is that the researchers started giving the more attention in e-commerce and robotics domains from 2008 and our correlation relationship also shows that flexibility has a strong relationship with these fields. We also noticed that before 2008 there is only one study in robotics field that was published in 2005 [34] and we notice the flexibility was the main concern of that study. Similarly in 2010 the lack of robotics studies flexibility decreased, in 2010 there were two studies of robotics and flexibility was the main concern of the study [14] and similarly in 2012 there is only single study on robotics [21] and the main concern of that study was also flexibility. Before 2008 most of the studies focused on the embedded and web services domains and our correlation results also shows that efficiency/performance have strong relationship with embedded and reliability have strong relationship with web services. In 2010 most of the researcher focused on embedded domain in resulted efficiency/performance and reliability were the main concerns in that year and its inverse effect noticed on flexibility.

4.2.5 Summary of learned lessons from RQ1:

Most of the researcher are focused on the Software design field followed by the software quality field. Almost two thirds of studies lie in these two fields and the remaining one third of studies focus on all the other fields. In application domains we learn that by numbers most of the researchers focused on embedded system domain but our statistical shows that most focus of the researchers are web based systems(e-commerce, web services). We notice that efficiency/performance was the main concern in embedded system domain. Also we noticed that reliability was the main concerns in web services domain and flexibility was the main concern in the robotics domain. We have seen that in 2010 flexibility suddenly decreased for that we seen from the studies that most of the studies in that year were focused on embedded systems domain and lack of robotics studies flexibility decreased in 2010. In quality concerns we noted that most of the researchers focus on the self-optimization, self-healing and self-configuration and little attention was given to self-protection. Most of the researcher give more attention to efficiency/performance followed by reliability as compared to other quality concerns. We also note that most of the researchers focus on single concerns, 58% of the studies considered a single concern, 34% considered 2 concerns, the remaining 8% considered 3 or 4 concerns. The main focus of the researcher on the quality concerns and less attention on other concerns such as cost, memory usage, complexity, stabilization and engineering effort etc. Although other concern get some attention in some but still quality concern are the main concerns in that studies because in those studies there are more than one concerns.

4.3 RQ2: What are the tradeoffs implied by self-adaptation?

In this section we discuss the results based on the Research Question(RQ2) that we defined in section A.4.

To answer this question, we used data extracted from claimed benefits (F10), and tradeoffs (F11). Figure. 4.6 summarizes the claims versus the tradeoffs of self-adaptation. The figure clearly shows that most of the researchers focus on claimed benefits, while little attention is given to the implications of self-adaptation. We note that the main focus of the researchers on claimed to be positively influenced by self-adaptation which is 92% of the claimed quality attributes and the remaining 8% are negatively influenced. It is remarkable that efficiency/performance is almost the only reported quality attribute with a negative effect as a result of self-adaptation besides a few studies that report a negative impact on availability and flexibility.

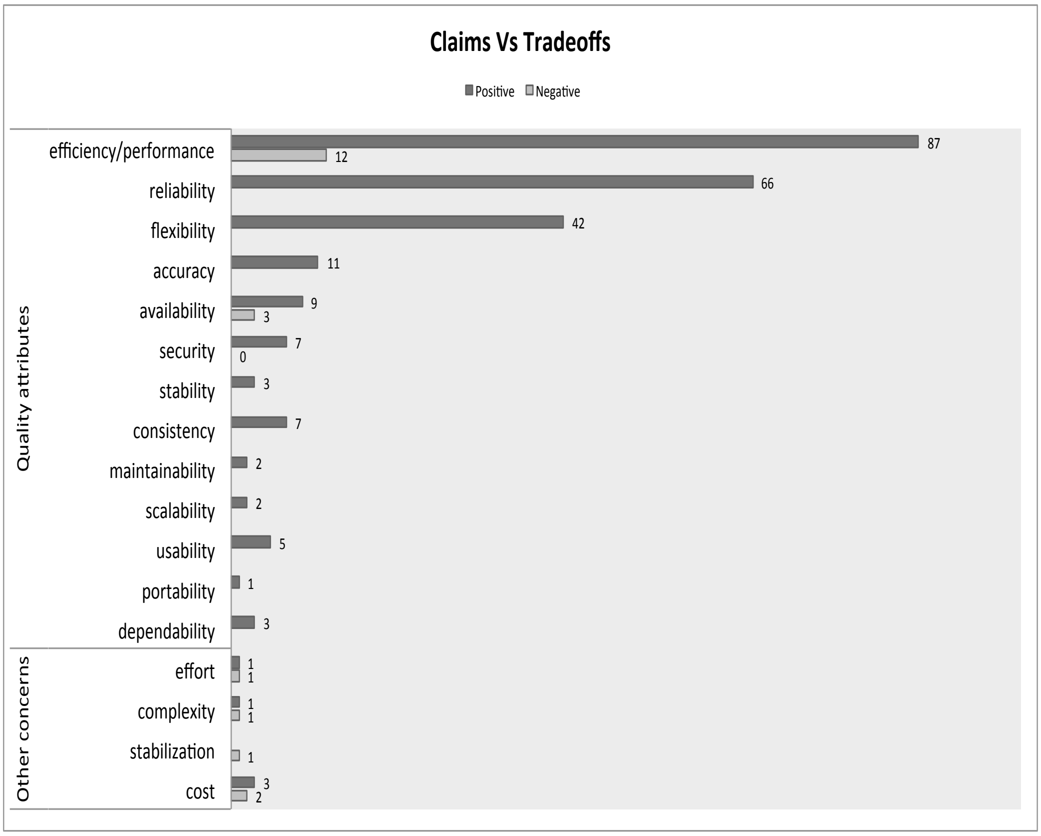


Figure 4.6: Calims and Tradeoffs

We notice that most of the dominant claims are recorded as improving the quality of the software which is 82,5% of the total claimed quality concerns, 6,8% claimed benefits recorded as assuring the quality of the software and remaining 3% preserving the quality of the software. Note that the ratio claims versus tradeoffs for the other concerns show an inverse image in stabilization and positive image in cost as for the quality attributes, notice that 50% of the other concerns are negatively affected and 50% positively.

4.3.1 Summary of learned lessons from RQ2:

Most of the researcher were focused on the improvement of the software quality attributes and less attention was given to negative influence. An interesting observation is that negative effect of efficiency/performance is mostly negatively influenced when quality attributes flexibility and reliability are improved. In contrast flexibility and availability are only negatively influenced when efficiency/performance is improved. We noted that dealing with tradeoffs were very weak, we recommended that researcher should be given more attention on tradeoffs of the software quality attributes.

4.4 RQ3: How much evidence is available for the claims and what are the types of evidence?

In this section we discuss the results based on the Research Question(RQ3) that we defined in section A.4.

To answer this question, we drew on data extracted from Quality score(F6), assessment approach (F12), evidence level (F13), repeatability (F14), citation count (F5) and validation setting (F15).

Figure. 4.7 shows a frequency analysis of score for each quality questions in table 2. All of the studies provided a rational for why the study was undertaken and also described the context of the study in which they have been conducted, and in most of the studies (57,4%) authors provide a general problem description, and 42,4% authors provide an explicit problem description for the research in the study(Q1-2.1). performed(Q2.2).

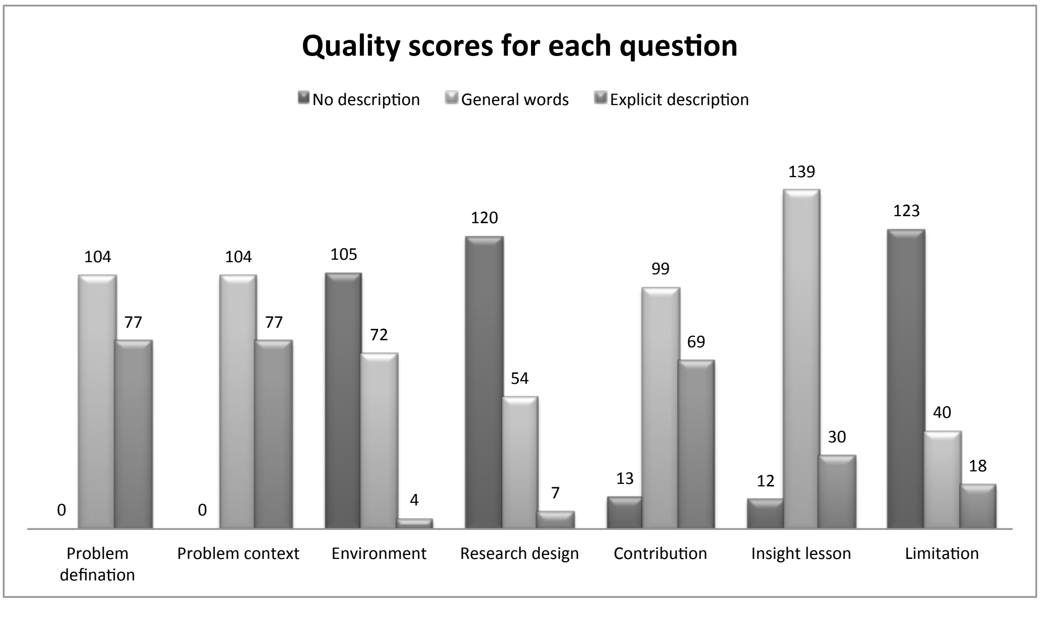


Figure 4.7: Quality score for each question

However 58% of studies do not describe environment in which the study was carried out, only 39,7% studies provide some general words about the environment and only in 4 studies (2,2%) authors provide an explicit description of the environment in which this research was Similarly 66,3% studies do not properly describe a research design in those studies there is no description of the research or how the study was organized. Only in 4 of the studies (2,2%), the authors explicitly describe the research plan, in 29.8% of the studies authors provide some general words about the research plan or the way the research was organized(Q3).

Most of the studies provide their contributions of the study, in 54,69% of the studies authors provide some general words about the results of the study and in 38% of the studies authors explicitly list the contributions/results of the study. Only in 13 (7,2%) studies there is no description of the research results(Q4). The most of the studies do not explicitly discuss the issues of insights/lessons, validity and reliability of their findings and only 16,6% of the studies provide explicitly learned/findings from the study(Q5). While 67.95% studies do not discuss the limitations of their approaches, 22% studies discuss their limitations in general words and only 10% studies explicitly list the limitations of the study.

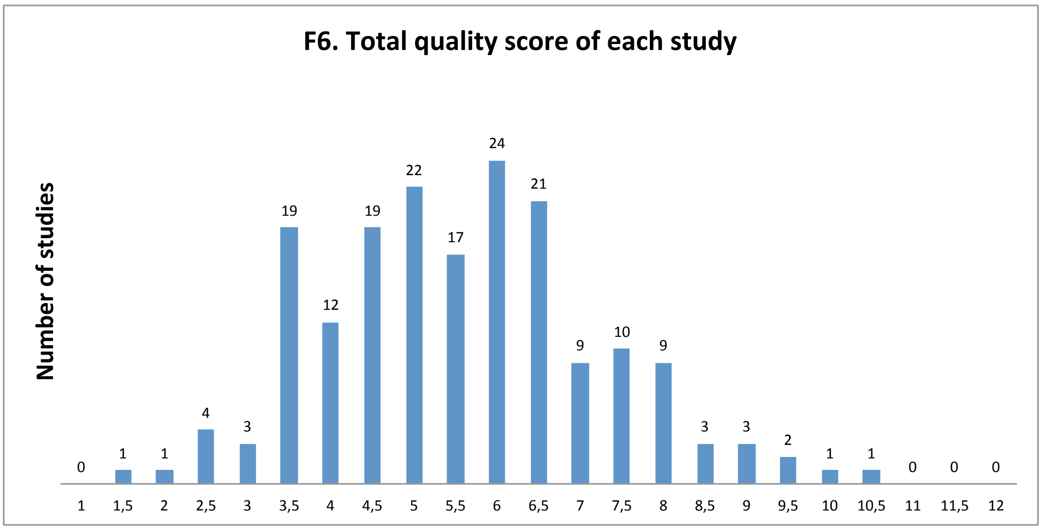


Figure 4.8: Total quality score

Figure. 4.8 , give us an overview of the distribution of total quality scores.The maximum total score is 12 and most of the studies received a score between 3,5 and 6,5. We summarizes the overall rating of the reviewed studies in four groups poor, fair, good and high. 15,4% of the studies scored between 0 and 3,5 and grouped it in poor quality, 52% of the studies lie between 4 and 6 and are considered fair studies,28% of the studies lie between 6,5 and 8,5 and are considered good quality and only 7 of the studies

(4%) scored 9 or higher on the overall quality rating and are considered high quality studies. We also undertook the statistic analysis of the quality score and number of studies in Table 7.

Table 7: Regression Analysis between number of studies and quality score on each forum

Venues	Regression Eq.	R^2	R	Mean	S.D.
Conferences	y = -0,2391x + 3,6848	0,1	-0,31	5,27	1,79
Symposia	y = -0,1937x + 3,3458	0,04	-0,21	5,48	1,37
Bookchapters	y = -0,0257x + 0,4279	0,03	-0,16	5,42	1,5
Workshops	y = -0,0791x + 0,9486	0,14	-0,37	4,5	1,11
Journals	y = -0,1265x + 3,7787	0,02	-0,13	6,03	1,73

This table tells us that all of the venues have negative relationship between quality score and number of studies. The correlation between quality score and number of studies of workshops have stronger then other venues, this means when the number of studies tends to decrease as the quality score increase which shows that the quality of workshops studies are low then other venues. Similarly the correlation relationship of Journals studies are weak then the other venues. This mean the number of studies are slowly decreased when quality of score increases and on average from 4 to 8 quality score of studies occurs in the journal studies. So we can say that the quality of journals studies are higher than the other venues. It should be noted that none of the studies got a full score or zero score on the overall quality assessment criteria, which means that on average in terms of quality the studies are neither perfect, nor are they completely flawed. We learn from this that overall quality of studies appears to have improved. In particular, the quality of journals are very good as compared to other venues. The highest quality of studies are presented in the following Table 8.

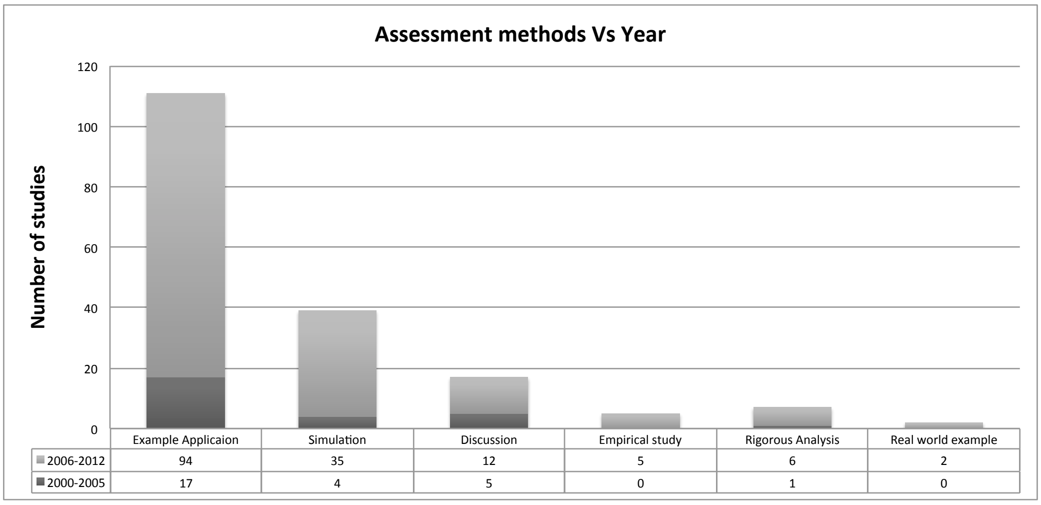
4.4.1 Classification of studies according to the assessment methods and year

Figure. 4.9 shows an overview of assessment methods with respect to years. Not surprisingly, example application accounts for the more than half of the studies making up 61% of the used assessments, simulation 21,5% and discussion 9.3% of the used approaches. Only a few (5) empirical studies listed in Table 12 and few(7) rigorous analysis studies are reported.Note that only 2 studies found used approach experience from real examples. It should be noted that the specific attention for rigorous analysis has significantly

Table 8: Studies with highest score

Year	Study Title	Quality Score
2008	Recommending Adaptive Changes for Framework Evolution	10,5
2011	Dynamic QoS Management and Optimization in Service-Based Systems	10
2011	Recommending Adaptive Changes for Framework Evolution	9,5
2012	Adam: Identifying defects in context-aware adaptation	9,5
2012	FORMS: Unifying Reference Model for Formal Specification of Distributed Self-Adaptive Systems	9
2010	Toward Physically-Adaptive Computing	9
2011	Model-based Self-Adaptive Resource Allocation in Virtualized Environments	9

decreased after 2007, while the inverse effect can be noticed for empirical studies after 2007.



	Example Applicaion	Simulation	Discussion	Empirical study	Rigorous Analysis	Real world example
▨ 2006-2012	94	35	12	5	6	2
■ 2000-2005	17	4	5	0	1	0

Figure 4.9: Total number of studies per year with assessment methods

4.4.2 Statistical analysis between assessment methods and number of primary studies on each year.

We undertook the regression analysis and standard derivation to find out the relationship between number of primary studies and assessment approaches in Table 9.

Table 9: Regression Analysis between assessment methods and number of primary studies

Assessment Methods	Regression Eq.	R^2	R	Mean	S.D.
Example Application	y = 0,6434x - 0,4199	0,93	0,96	8,54	7,26
Simulation	y = 0,2446x - 0,4051	0,79	0,89	3	3
Discussion	y = 0,0501x + 0,61	0,24	0,49	1,31	1,11
Empirical study	y = 0,046x - 0,2552	0,42	0,65	0,38	0,77
Rigorous Analysis	y = 0,0067x + 0,4451	0,01	0,08	0,54	0,88
Real world example	y = 0,0092x + 0,0251	0,07	0,27	0,15	0,38

From this table we can clearly see that example application has the strongest relationship with a number of the primary studies per year and it leads as a prominent assessment approach followed by simulation. We also undertook the statistical analysis of assessment approaches with top most prominent quality concerns and application domains.

Table 10: Correlation between assessment methods and quality concerns

Assessment Methods	Efficiency/Performance	Reliability	Flexibility
Example Application	0,93	0,89	0,75
Simulation	0,93	0,63	0,83
Discussion	0,74	0,25	0,18
Rigorous Analysis	0,74	0,3	0,16
Empirical study	0,88	0,64	0,63
Real world example	1	0	0

Table 10 shows us that Example Application have strong relationship between efficiency/performance and reliability. Similarly simulation have strong relationship with efficiency/performance and flexibility. Note that Empirical study have strongest relationship with efficiency/performance. We also note that in Real world example approach researchers only deal with efficiency/performance.

Table 11 shows that most of the researchers used example application as approach in e-commerce domain. We notice that in robotics domain researcher only used two assessment methods, example application and simulation. We also notice that rigorous analysis only used in embedded sys-

Table 11: Correlation between assessment methods and application domains

Assessment Methods	Embedded	Web services	Robotics	E-commerce
Example Application	0,72	0,66	0,77	0,95
Simulation	0,73	0,73	0,76	0,67
Discussion	0,43	-0,08	0	0,28
Rigorous Analysis	0,73	0,74	0	0
Empirical study	0,24	0	0	0
Real world example	0	0	0	0,68

tems and web services domains. Similarly real world example only deal in e-commerce domain.

Table 13 shows the distribution of evidence level. Given the used assessment methods, it is not surprising that most studies have a low evidence level and evidence only obtained from demonstrations or toy example(93%). This confirms previous studies, such as [32]. In Table 15 we list the studies with plus 100 citations counts because these studies are more likely to be used by the researchers in their research.

We list studies with the highest evidence level in Table 14.

In Table 15 we list the studies with plus 100 citations counts because these studies are more likely to be used by the researchers in their research. Figure. 4.10 shows the results for repeatability of the studies. Repeatability is considered as a foundation for quality research, as it allows to test and verify research results. It is not surprising that most of the studies(66%) provide no useful information to repeat the study. 30% of the studies provide a partial description and 4% provide partial material to repeat the study. On the other hand, there is no such study that provide all the necessary material to repeat the study.These results also confirm our previous results that we found in [32]. Note that after 2007 more than half of the studies provide material to repeat the study.

4.4.3 Statistical analysis of repeatability of study.

We undertook the relationship between number of repeatable studies and number of primary studies on each year as shown in Table 16

These relationship shows that not repeatable studies have a stronger relation with number of primary studies then partial description, the reason for this is because partial description in 2010 low reported as compared to 2009

Table 12: Empirical studies

Year	Study Title	Quality concerns
2012	Adam: Identifying defects in context-aware adaptation	reliability
2011	Recommending Adaptive Changes for Framework Evolution	efficiency/performance
2011	On the Performance of UML State Machine Interpretation at Runtime	efficiency/performance, flexibility
2008	An experimental study of adaptive testing for software reliability assessment	reliability
2008	Supporting Software Evolution Using Adaptive Change Propagation Heuristics	efficiency/performance

Table 13: Level of evidence

Level	Evidence	Absolute	Percentage
1	Demo example	168	92,8%
2	Expert opinion or observations	6	3,3%
3	Evidence obtained from empirical study	5	2,7%
4	Industrial evidence	2	1%

and 2011. It also should be noted that only few studies (listed in Table 17) provided partial material to repeat the study. The domains of these studies are: embedded, robotics, and Traffic, transportation. However on the other hand, there is no study in web service domain that provide partial material to repeat the study.

In validation setting we note that none of the study reported with industrial effort, however some studies 2,76% of the studies presented with the collaboration of academic and industrial effort. Most of the studies reported in academic effort as listed in Table 18.

Table 14: Studies with highest evidence level

Year	Study Title	Level
2011	An Autonomous Engine for Services Configuration and Deployment	4
2010	A Lightweight Transformational Approach to Support Large Scale Adaptive Changes	4
2011	Recommending Adaptive Changes for Framework Evolution	3
2011	On the Performance of UML State Machine Interpretation at Runtime	3
2008	An experimental study of adaptive testing for software reliability assessment	3
2008	Supporting Software Evolution Using Adaptive Change Propagation Heuristics	3

Table 15: Studies with Citation Count >100

Year	Study Title	Citation
2003	CARISMA: Context-Aware Reflective middleware System for Mobile Applications	427
2002	Model-based Adaptation for Self-Healing Systems	235
2006	Model-Based Development of Dynamically Adaptive Software	216
2004	Autonomous Adaptation to Dynamic Availability Using a Service-Oriented Component Model	168
2006	Autonomic Live Adaptation of Virtual Computational Environments in a Multi-Domain Infrastructure	137
2004	A Component Based Programming Framework for Autonomic Applications	129

Table 16: Correlation of repeatability

	Partial material	Partial description	Not repeatable
Correlation	0,53	0,89	0,94
Mean	0,54	4,23	9,15
S.D.	1,13	3,83	7,28

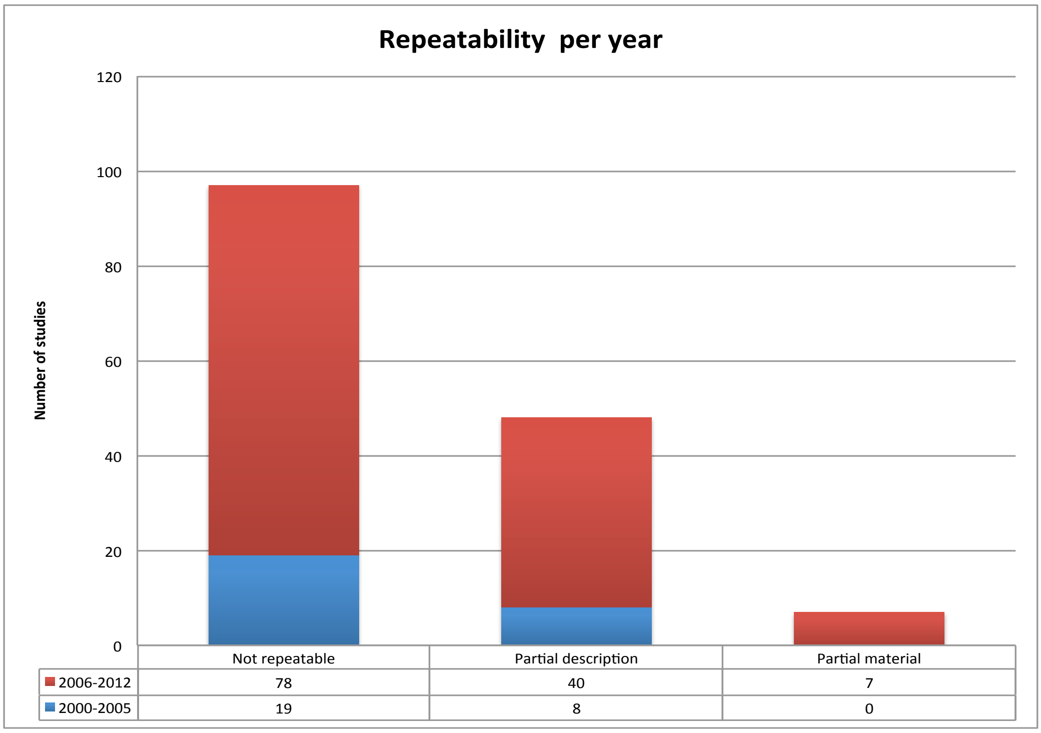


Figure 4.10: Repeatability of the studies per year

Table 17: Studies provided with partial material

Year	Study ID	Application domain
2010	On Decentralized Self-Adaptation: Lessons from the Trenches and Challenges for the Future	Traffic, Transportation
2009	Reinforcement Learning-Based Dynamic Adaptation Planning Method for Architecture-based Self-Managed Software	Robotics
2009	Policy-Based Architectural Adaptation Management: Robotics Domain Case Studies	Robotics
2009	Self-Repairing Systems Modeling and Verification using AGG	Embedded
2009	Using Filtered Cartesian Flattening and Microrebooting to Build Enterprise Applications with Self-adaptive Healing	Not identifiable
2008	Recommending Adaptive Changes for Framework Evolution	Not identifiable
2006	Symbolic Invariant Verification for Systems with Dynamic Structural Adaptation	Embedded

Table 18: Validation setting

	Studies	Percentage
Academic effort	176	97,24
Academic/industry collaboration	5	2,76
Industrial effort	0	0

5 Discussion

In this chapter, we first start to discuss our study limitations and the we presents our personal reflection of the study. Next, We conclude with a discussion of solved research questions and opportunities for future research.

5.1 Limitations of study

The methodology that we have adopted in this thesis has some limitations. Our study is limited to 19 major venues listed in Table 1. We believed that these venues are most prominent venues for software engineering studies for self adaptive systems. Another limitation is the potential biased of reviewers. We believe that literature which is included in our study has been filtered thoroughly in a comprehensive selection process that involved two reviewers who participated in cross-checking of search queries, and approving or rejecting one particular study and if both researcher not agreed then an external reviewer also involved for the consensus of the decision. In section 2, where we comprehensively discussed our research methodology, we believe it should minimize the threat of potential biased.

As there are other commonly used terms such as self-managing systems and autonomic computing or autonomous systems; these are strongly related to self-adaptive systems. In practice these terms are used interchangeably, however it is very difficult to draw a distinction between these terminologies and may be possible that we miss a study. We used only simple regression analysis on different data items for finding the correlation between them and in future we will analysis of our data with multi regression.

5.2 Personal reflections on the review

During our systematic review process many questions arose. As we followed the guidelines of Kitchenham et al. [17] for conducting the review. Kitchenham provides a set of guidelines for conducting a systematic literature review, which provides steps for formulating research questions to be answered to the review and developing a review protocol. Many researcher on different research areas of software engineering are also conducts the review by using these guidelines in their research [23, 13, 12]. By using guidelines of [17] we also develop our protocol that is available in Appendix A as sample for the future systematic literature review.

After developing the protocol the next step was searching the studies from the digital database libraries by running the search query string. Searching studies from different data sources was a very challenging task because every search engine has its own limitations. Every search engine has a different way of searching the studies by a search string. We noted that by running the search string on data sources we retrieved many irrelevant studies and rejected on the ground of topics. If more meaningful titles, keywords and the like are used then the precision of the searching process could be improved [18]. After the primary selection of studies we start data extraction on the forms and also save that data in electronic database excel.

After the data extraction we start to analyse the data by using graph representation and statistical analysis. We noted that in analysis phase there are also important that researcher must had some knowledge about statistical methods such as mean, regression analysis, correlation, standard derivation etc. If you don't have any knowledge about statistics then it should be possible that you could not be able to make appropriate relationship between the different data items. In our thesis when we analyse our data by using simple regression analysis and correlation then we learn lot of things about statistics analysis and found that how much importance of statistics in empirical studies. With statistics analysis we analyze our data in better way.

The most exciting and interesting part of our thesis is preparing for the results analyzing and explaining them. Although analysis part is very difficult because this is the part, in which the reviewers are most interested. This part will decide the fate of entire thesis. The most time consuming part is paper reading.

So based on our experience in this systematic literature review we are giving some suggestion for future students who wants to conduct SLR. 1) Formulate your research questions. 2) For searching the studies used automatic search as much as possible for that you must know how you can run the search query on different search engines . 3) Plan your time because without this you could not be able to finish it on time. 4) You should have some knowledge about statistics.

5.3 Conclusion

We performed a systematic literature review study to summarize existing research on engineering self-adaptive software systems and highlighted the claims that are made about self-adaptation and the available evidence for these claims. Our study shows that JSS and SEAMs are the most attractive venues for the research on self adaptive systems. We also noticed that the studies of embedded system domain are mostly addressed in the JSS, So we can say JSS is the most prominent venue for embedded systems studies. We solved our research questions that we defined in section A.4. As a result we learned from these research questions that, the most addressed claims of self adaptive systems are efficiency/performance, reliability, flexibility, accuracy and availability. Overall evidence level of the studies are very low due to the lack of attention on empirical studies and industrial evidence. Only few studies have been reported with the academic/industry collaboration, and no industrial effort is reported. We suggests on the behalf of our experience that: i). The researcher should be explained in detail their assessment approaches to increase the evidence level of the study. ii). More attention needed on tradeoffs from the self adaptive community. iii). More attention required from the academic/industry collaboration effort. iv). Detailed research design required from the researchers and need resources where data is publicly available for the future research.

5.4 Future Work

In future we analyse our data by using multiple regression analysis. We will also add more venues for the review. It should be interesting if we see the whole study for the architecture point of view and for this it may be possible that we will add some more fields for the architecture of self-adaptive systems. For this we will need to review again whole studies for the new data items. Because it should be interesting that peoples wants to know how many studies uses distinction loop, multiple loops or single loop. And currently we are working on this topic.

For the future students who wants to do this type of study there should some interesting areas in the self-adaptive systems such as Self Organizing systems and control loops in self-adaptive systems.

References

[1] Abran, A., Bourque, P., Dupuis, R. and Moore, J. W., eds [2001], *Guide to the Software Engineering Body of Knowledge - SWEBOK*, IEEE Press, Piscataway, NJ, USA.

[2] *Autonomic computing 8 elements* [2001].
URL: *http://www.research.ibm.com/autonomic/overview/elements.html.*

[3] Babaoglu, O., Jelasity, M., Montresor, A., Fetzer, C., Leonardi, S., Moorsel, A. v. and Steen, M. v. [2005], *Self-star Properties in Complex Information Systems: Conceptual and Practical Foundations (Lecture Notes in Computer Science)*, Springer-Verlag New York, Inc., Secaucus, NJ, USA.

[4] Basili, V. et al. [1994], Goal question metric approach, *in* 'Encyclopedia of Soft. Engineering'.

[5] Brereton, P. et al. [2007], 'Lessons from applying the systematic literature review process within the software engineering domain', *Journal on Systems and Software* **80**.

[6] Brun, Y. [Cape Town, 2010], Improving impact of self-adaptation and self-management research through evaluation methodology, *in* 'SEAMS'.

[7] Brun, Y., Marzo Serugendo, G., Gacek, C., Giese, H., Kienle, H., Litoiu, M., Müller, H., Pezzè, M. and Shaw, M. [2009], Software engineering for self-adaptive systems, Springer-Verlag, Berlin, Heidelberg, chapter Engineering Self-Adaptive Systems through Feedback Loops, pp. 48–70.

[8] Cheng, B. et al. [2009], Software engineering for self-adaptive systems: A research roadmap, *in* 'Software Engineering for Self-Adaptive Systems', Springer.

[9] de Lemos, R. and Fiadeiro, J. L. [2002], An architectural support for self-adaptive software for treating faults, *in* 'Proceedings of the first workshop on Self-healing systems', WOSS '02, ACM, New York, NY, USA, pp. 39–42.
URL: *http://doi.acm.org/10.1145/582128.582136*

[10] de Lemos, R. et al. [2012], Software engineering for self-adaptive systems: A second research roadmap, *in* 'Software Engineering for Self-Adaptive Systems II', LNCS, Springer.

[11] Dybå, T. and Dingsøyr, T. [2005], 'Evidence-based software engineering for practioners', *IEEE Software* **2**, 58–65.

[12] Dybå, T. and Dingsøyr, T. [2008], 'Empirical studies of agile software development: A systematic review', *Inf. Software Technology* **50**, 833–859.

[13] et. al., B. K. [2009], Systematic literature reviews in software engineering Ð a systematic literature review, Vol. 51.

[14] et al., F. S. [2010], Incremental online evolution and adaptation of neural networks for robot control in dynamic environments, *in* 'Adaptive', ThinkMind.

[15] Horn, P. [2001], 'Autonomic computing: IBM's Perspective on the State of Information Technology'.

[16] Kephart, J. and Chess, D. [2003], 'The vision of autonomic computing', *Computer* **36**(1).

[17] Kitchenham, B. and Charters, S. [2007], 'Guidelines for performing systematic literature reviews in software engineering', (EBSE 2007-001, Keele and Durham University).

[18] MacDonell, S. G. and Shepperd, M. J. [2007], Comparing local and global software effort estimation models – reflections on a systematic review, *in* 'Proceedings of the First International Symposium on Empirical Software Engineering and Measurement', ESEM '07, IEEE Computer Society, Washington, DC, USA, pp. 401–409.
URL: *http://dx.doi.org/10.1109/ESEM.2007.31*

[19] Magee, J. and Maibaum, T. [2006], Towards specification, modelling and analysis of fault tolerance in self managed systems, *in* 'Self-adaptation and Self-managing Systems', ACM.

[20] McCann, J. and Huebscher, M. [2004], Evaluation issues in autonomic computing, *in* 'Grid and Cooperative Computing', LNCS vol. 3252, Springer.

[21] Meister, E. and Gutenkunst, A. [2012], Self-adaptive framework for modular and self-reconfigurable robotic systems, *in* 'Adaptive', ThinkMind.

[22] Mian, P., Conte, T., Natali, A., Biolchini, J. and Travassos, G. [2005], A systematic review process for software engineering, *in* '3rd International Workshop Guidelines for Empirical Work', Vol. 1 of *Workshop Series on Empirical Software Engineering*, pp. 1–6.

[23] Patikirikorala., T. et al. [2012], 'Technical report: A systematic survey on the design of self-adaptive software systems using control engineering approaches', *Software Engineering for Adaptive and Self-Managing Systems* .

[24] Robertson, P. and Laddaga, R. [2005], Self-star properties in complex information systems, Springer-Verlag, Berlin, Heidelberg, chapter Model based diagnosis and contexts in self adaptive software, pp. 112–127.
URL: *http://dl.acm.org/citation.cfm?id=2167575.2167586*

[25] Salehie, M. and Tahvildari, L. [2005], Autonomic computing: emerging trends and open problems, *in* 'Proceedings of the 2005 workshop on Design and evolution of autonomic application software', DEAS '05, ACM, New York, NY, USA, pp. 1–7.
URL: *http://doi.acm.org/10.1145/1083063.1083082*

[26] Salehie, M. and Tahvildari, L. [2009], 'Self-adaptive software: Landscape and research challenges', *ACM TAAS* **4**.

[27] Shaw, M. [2003], Writing good software engineering research papers: minitutorial, *in* 'Proceedings of the 25th International Conference on Software Engineering', ICSE '03, IEEE Computer Society, Washington, DC, USA, pp. 726–736.
URL: *http://dl.acm.org/citation.cfm?id=776816.776925*

[28] Sterritt, R., Parashar, M., Tianfield, H. and Unland, R. [2005], 'A concise introduction to autonomic computing', *Adv. Eng. Inform.* **19**(3), 181–187.
URL: *http://dx.doi.org/10.1016/j.aei.2005.05.012*

[29] Tamura, G. et al. [2012], 'Towards practical runtime verification and validation of self-adaptive software systems', *Software Engineering for Self-Adaptive Systems II, Springer* .

[30] Vassev, E. and Hinchey, M. [2009], 'ASSL: A Software Engineering Approach to Autonomic Computing', *Computer* **42**(6), 90–93.

[31] Waters, P. [2007], *Evidence-based Practices Part 1: What is Evidence-based Practice?*, Southern Coast Addiction Technology Transfer Center.

[32] Weyns, D., Iftikhar, U., Malek, S. and Andersson, J. [2012], 'Claims and supporting evidence for self-adaptive systems', *Software Engineering for Adaptive and Self-Managing Systems* .

[33] Weyns, D., Malek, S. and Andersson, J. [2012], 'Forms: Unifying reference model for formal specification of distributed self-adaptive systems', *ACM Trans. Auton. Adapt. Syst.* .

[34] Weyns, D., Schelfthout, K. and Holvoet, T. [2005], Architectural design of a distributed application with autonomic quality requirements, *in* 'Proceedings of the 2005 workshop on Design and evolution of autonomic application software', DEAS '05, ACM, New York, NY, USA, pp. 1–7.
URL: *http://doi.acm.org/10.1145/1083063.1083076*

[35] Zhang, J. and Cheng, B. [2006], Model-based development of dynamically adaptive software, *in* '28th International Conference on Software Engineering', ACM.

Protocol for Systematic Literature Review on Claims and Evidence for Self-Adaptive Systems (Version 1.0—2012-05-14)

Tanvir Ahmad and Muhammad Ashfaq Haider

February 13, 2013

Abstract

In order to ensure a consistent planning process of our investigating systematic literature reviews and to reduce the possibility researcher bias, we have developed a study review protocol.The aim of this protocol to provide a common structure for our study review.

A Appendix

A.1 Protocol Overview

The protocol description is organized in 6 sections as follows. In section A.2 we discuss background and justification our study. Section A.3 gives a general overview of the research method. In section A.4 we explain the research questions we address. Section A.5 defines the material and methods. In section A.10 we describe the data items that need to be collected. Finally, we conclude with a brief description of data analysis in section A.11.

A.2 Background and Justification

Self-adaptation has been widely recognized as an effective approach to deal with the increasing complexity and dynamicity of modern software systems [32]. A self-adaptive system comprises two parts: the managed system that deals with the domain functionality and the managing system that deals with the adaptations of the managed system to achieve particular quality objectives. Self-adaptation endows a software system with the capability to adapt itself to internal dynamics and dynamics in the environment in order to achieve certain quality goals. Over the last decade, researchers and engineers have developed a huge body of work on engineering self-adaptive systems. However, it is not clear how the research results have actually contributed to improvements of engineering complex software systems. There are many investigator that performed systematic study over the last years, e.g systems [19, 35, 30, 33, 29].

These studies summarize their achievements of the field. But no body has been performed the claimed evidence on self-adaptation. As a result, there is no clear view on how self adaptation actually contributes to tackling the challenges of engineering and managing complex distributed software systems engineering. However, such insight is crucial for researchers and engineers. Our objective is to study and summarize existing research on engineering self-adaptive software systems and shed light on the claimed benefits of self-adaptation and to what extent evidence exists for these claims. Our particular aim will be on

1. identify the claimed benefits of self-adaptation,
2. identify the evidence for these claims.

To that end, we have performed a systematic literature review. From the study, we derive conclusions concerning claims evidence in self-adaptive systems, and areas for future research.

A.3 Overview Research Method

A systematic literature review [17] is a well-defined approach to identify, evaluate and interpret all relevant studies regarding a particular research question, topic area or phenomenon of interest. Usually, systematic reviews conduction is a three-step approach. Figure A.1 the main steps composing the SLR process are regarding the Planning, Execution, and Result Analysis [22].

Two researchers are involved in this literature study. In planning Phase , the review protocol is defined by the researchers, as described in this document. The review protocol includes the definition of research questions, the search strategy and scope, the data items that had to be collected, and the approach for data analysis and presentation of the results. After approval of the protocol review the researcher have to conduct the review in execution phase. Studies have to be selected based on the search criteria and data has to be collected as specified in the protocol defined in Phase 1. Next, the

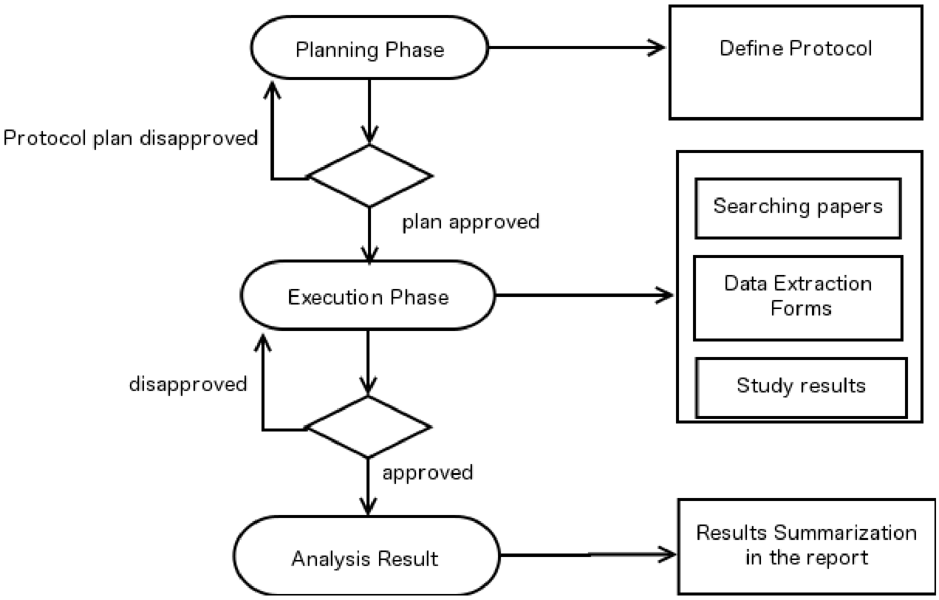


Figure A.1: Overview of the systematic review process (adapted from [22]).

data derived from the the primary studies has to be collacted and summarized to answer the research questions defined in the protocol. Finally, the review report has to be produced (Phase 3). The final report will be cross checked by an independent researcher. His feedback will be used to improve the description and correct minor issues.

A.4 Research Questions

We aimed at research questions meaningful not only to researcher but also to engineers. Thus, we first formulated the review goal through Goal-Question-Metric (GQM) perspectives (purpose, issue, object, viewpoint) [4]:

Purpose: Understand and characterize

Issue: the use of formal methods

Object: in self-adaptive software systems

Viewpoint: from a researcher and engineer viewpoint.

The general research question translates to four concrete research questions:

RQ1: What are the claims made for self-adaptation?

RQ2: What are the tradeoffs implied by self-adaptation?

RQ3: How much evidence is available for the claims and what are the types of evidence?

The goal of RQ1 is motivated by the need to get clear understanding of the expected usefulness of the current state-of-the-art in self-adaptive systems and claims of self-adaptation. RQ2 is to understand about what are the tradeoffs by applying self-adaptation.With RQ3 we aim to investigate what assessment method have been used for evidence and what is the level of evidence of the claimed associated with self-adaptation.

A.5 Material and Methods

The process used for this literature review is comprises a number of distinct phases.

- Study Design
  Subjects (Material)
  Procedure (Methods)
- Data Collection
- Result Analysis
- Report Writing

A.6 Study Design

Our study design is consists of subjects and procedure. Subject of our study is self-adaptive systems and procedure means how we collect the relevant studies such as searching strategy, inclusion and exclusion criteria etc. For Systematic review a well planned search strategy is very important to get the relevant study. We described our strategy from the following dimensions: search scope of inclusion and exclusion criteria, search method (i.e., automatic search or manual search), search strings, and electronic data sources used and venues.

A.6.1 Inclusion Criteria

We limited our literature search over two dimensions time (publication period) and space (publication venues).

- Studies which are published between 1st of January 2000 to 31st August 2012. We decided this broader period due to the reason that self-adaptive systems have become subject of active research around that time.
- Studies which are written in English language.
- The study must be related to the field of self-adaptive systems that must addressed adaptation logic in which after engineering self-adaptive software systems performs adaptations of the managed system when needed.
- We will included only those studies that provide any evidence and assessment method to prove their claims. Assessment are validation approaches that may be in the form of example application, simulation, rigorous analysis, empirical, real world example or studies that provided some qualitative, textual, opinion-oriented evaluation. E.g. compare and contrast, oral discussion of advantages and disadvantages. Evidence may be in the form of toy example, observations, experiments, empirical or industrial evidence in Example application Authors describing an application and provide an example to assist in the description, but the example is "used to validate" or "evaluate" as far as the authors suggest [27].
- Study comes from an acceptable source, which are listed below (Table 1). We aim at gaining solid information about claims and thus ignored lower-quality sources.
- We have defined a minimum criteria. that a search string from the list of terms, must occur at least once in the title of the paper, or in the abstract of the paper.

A.6.2 Exclusion Criteria

- We exclude the studies which are related to theoretical. Rational: We are not interested those papers that are only provides us a new theory about self-adaptation or any aspect of self-adaptation.
- We also exclude those papers that study is an Tutorial, short paper, editorial etc because these papers do not provide a reasonable information.

A study is selected when it met the inclusion criterion and eliminated if it met the exclusion criterion.

A.6.3 Search Method

We will perform both type of search methods manual search and automatic search. Automated search will be performed by executing search string on electronic data search engines such as IEEE Explore ,ACM Digital library etc. Manual search refers to performed manually browsing Conference proceedings or Journals proceedings. In the 1st stage we will almost perform automatic search but in the 2nd stage we will try to perform mostly manual search because manually we can avoid miss relevant material compared to automatic search.

A.6.4 Constructing Search Terms (Keywords)

The search for relevant studies will be begin from digital bibliography search engines through a search string and sources to search for relevant papers for the study review. Deciding search string for this review is challenging because there are lot of keywords used in the areas of software Engineering , Artificial Intelligence , Robotics and Autonomic systems. We have chosen following keywords related to self adaptive systems on the basis of research questions.

Keywords and Synonyms: adaptive , adaptation, self-* , autonomic,autonomous, adapt*,autono*

From these keywords we construct the following searching string which we will use in our 1st step.

> ((Title:adaptive OR Title:adaptation OR Title:self OR Title:autonomic OR Title:autonomous) OR
>
> (Abstract:adaptive OR Abstract:adaptation OR Abstract:self OR Abstract:autonomic OR Abstract:autonomous))

To ensure the validty of the search string we will apply pilot searches on the set of papers from JSS, SEAMS, WOSS and TAAS to ensure that the keywords provide the right scope of papers.

A.7 Venues

To ensure a minimum level of quality of studues, we include the primary venues for publishing research results on self-adaptive systems, as well as the major conferences and journals on software engineering. The selected venues are listed in Table 19. The Rank is based on the evaluation published by the Australian Research Council.[1]

[1] http://www.arc.gov.au/era/era_2010/archive/era_journal_list.htm

Table 19: Searched venues

ID	Venue	Rank
ASE	International Conference on Automated Software Engineering	A
Adaptive	Adaptive and Self-adaptive Systems and Applications	n/a
DEAS	Design and Evolution of Autonomic Application Software	n/a
ICAC	International Conference on Autonomic Computing	B
ICSE	International Conference on Software Engineering	A
ICSM	International Conference on Software Maintenance	A
ISSTA	International Symposium on Software Testing and Analysis	A
SASO	Self-Adaptive and Self-Organizing Systems	n/a
SEAMS	Software Engineering for Adaptive and Self-Managing Systems	n/a
ISARCS	International Symposium on Architecting Critical Systems	n/a
SefSAS	Software Engineering for Self-Adaptive Systems	n/a
WICSA	Working International Conference on Software Architecture	A
WOSS	Workshop on Self-Healing	n/a
WADS	Workshop on Architecting Dependable Systems	n/a
FSE	Foundations of Software Engineering	A
JSS	Journal of Systems and Software	A
TAAS	Transactions on Autonomous and Adaptive Systems	n/a
TOSEM	Transactions on Software Engineering and Methodology	A*
TSE	Transactions on Software Engineering	A*

A.8 Resources to be searched

- IEEE Explore
- ACM Digital library
- ThinkMind(TM) Digital library

- ScienceDirect
- SpringerLink
- Google scholar (scholar.google.com)

A.9 Quality assessment

Assessing the quality of the paper or its contribution is important for data synthesis and interpretation of results later on. As all studies were assessed through a quality check, To assess the quality, we collected a set of quality items as show on Table 2. These items are based on the assess method for research studies proposed in [12]. From the answers, a quality assessment score (max 12) is calculated by summing up the scores for all the questions for a study (scores for the various options are given between brackets). We adopted the quality assessment questions (Table 3)that used in SEAMS study Weyns et al. [32].

A.10 Data Collection Items

Data items that we will use to reviewing each research paper. We will maintain the data in the forms, which will be specially prepared for recording the data extraction. Data extraction form should be the following data items shown in Table 21. We selected these data items except F15 from the SEAMS study Weyns et al. [32].

The data items author(s), year, title, venue, citation count (F1-F5) were used for documentation. Quality score (F6) is used for assessing the quality of study.
Software engineering field of the study (F7). Software engineering can be divided into ten subdisciplines. They are: [1]

- Software requirements:
- Software design:
- Software construction:
- Software testing:
- Software maintenance:
- Software configuration management:
- Software engineering management:

- Software engineering process:
- Software engineering tools and methods:
- Software quality:
- other.

Applications domain (F8) for which self-adaptation is used in the study. We started from the application domain taken from the study Weyns et al. [32]. Possible application domains are:

- web services
- embedded systems
- robotic systems
- traffic and transportation
- other.

Quality concerns related to self-adaptation. Formulation that are (positively or negatively) affected by self-adaptation in the study. The options (based on IEEE 9126 and ISO/IEC 25012) are:

- reliability (fault tolerance, recoverability): capability of software to maintain its level of performance under stated conditions for a stated period of time
- availability: the degree to which the software is in a functioning condition, i.e. capable to perform its intended functions
- usability (ease of learning, communicativeness): effort needed to use the system
- efficiency/performance (time behaviour, resource utilization): efficiency of the software by using the appropriate resources under stated conditions and in a specific context of use
- maintainability (analyzability, changeability, stability, testability): effort needed to make specified modifications.
- portability: ability of software to be transferred from one environment to another
- security: ability of the system to protect against misuse

- accuracy: the extent to which the software realizes the intended behavior in a specific context of use

- flexibility in use: capability of the software to provide quality in the widest range of contexts of use, incl. dealing with unanticipated change and uncertainty

- other.

Claimed benefits of self-adaptation (F10) can be one or more of the following (add the appropriate concerns selected in F9):

- preserving quality of the software (i.e., self-adaptation does not add quality to the system but maintains some quality attributes)

- improving quality of the software (i.e., self-adaptation add some quality to the system, i.e., it improves some quality attributes)

- assuring quality of the software (i.e., self-adaptation guarantees some quality attributes, typically by means of strong evidence or formal proof)

- improving other concerns

Tradeoffs (F11) refers to the concerns of self-adaptation with a negative impact. This can be one or more of the following (add the appropriate concerns selected in F9):

- quality concerns that are negatively influenced due to self-adaptation (add the appropriate quality concerns selected in F10)

- other concerns that are negatively influenced due to self-adaptation (add the appropriate quality concerns selected in F10)

noindent Assessment / validation approach (F12) has the following options:(taken from the the study Weyns et al. [32])

- discussion (the authors provide a qualitative, textual, opinion-oriented evaluation; e.g., compare and contrast, oral discussion of advantages and disadvantages; the discussion may include the description of examples, but these examples must provide any evidence or results may be in the form of observations, toy example or industrial evidence etc.)

- example application (the authors use a concrete application or applications to assist the assessment of their work; the example application is either concretely realized, but not necessarily completely described in the study, or the application is used to validate or evaluate the work as far as the authors suggest).
- simulation (the authors access their work by the execution of a system with artificial data, using a model of the real world; the evaluation typically compares the work with a baseline approach by means of numerical results; this comparison may (or may not) use and existing benchmark)
- rigorous analysis (the authors employ a rigorous derivation and proof for assessment; rigorous analysis is typically based on formal methods)
- empirical study (in an empirical study, assessment is based on observation or experience. An empirical study includes clearly defined research questions/hypothesis that typically include statements about the proposed research approach in comparison with a baseline approach. The questions/hypothesis are tested with a suitable experiment. In the experiment, data is collected from both the proposed approach and the baseline approach. This data can be based on observation of the behavior of humans or software. Statistical analysis of the collected data is used to test the hypotheses and provide evidence or not. A discussion of treats to validity is part of an empirical study. Concrete examples of empirical studies are case study, controlled laboratory experiment, and controlled experiment performed in industry setting.)
- experience from real examples (the results have been used on real-world examples, but not in the form of empirical studies; the evidence of its use is collected informally or formally)

Evidence level (F14) is one of the following options:(taken from the the study Weyns et al. [32])

- evidence obtained from demonstration or application to simple/toy examples
- evidence obtained from expert opinions or observations
- evidence obtained from empirical studies (e.g., controlled lab experiments, causal case studies, etc.)
- industrial evidence

- other.

Repeatability (F15) of the study is one of the following options:(taken from the the study Weyns et al. [32])

- study is not repeatable (no useful description of material is available to repeat the study)
- a partial description is available to repeat the study (may be described in the paper)
- the material to repeat the study is partially available (this typically includes links to material that is used to repeat the study)
- all the material is available to repeat the study (the study can be repeated with reasonable effort, probably in a different but similar setting)
- other.

Validation setting. (F16) is one of the following options:(taken from the the study [6])

- Academic effort
- Academic/industry collaboration
- Industrial effort
- other

A.11 Result Analysis

The data derived from the the studies will be collated and summarized to answer the research questions. The synthesis includes the following:

1. Listing of findings,
2. Reaching consensus among reviewers in case of conflicting opinions,
3. Analysis of findings, and
4. Answering research questions and interpretation of the results.

Based on the synthesis, conclusions and recommendations for future research in the field will be derived, and limitations of the review have to be identified. Finally, the results of the review are presented in a review report.

A.12 Validation of the Protocol

Significant changes to the Protocol now have been made in accordance with all the feedbacks and comments from the reviewer. This version (1), will be used for review. If we should need any further changes we will update this protocol and change the version number accordingly.Every change to the protocol will be recorded and the protocol updated accordingly.

Table 20: Items to assess study quality

1.Rationale for why the study was undertaken. Options are:

- (2) The authors provide an explicit problem description for the research in the study.
- (1) The authors provide a general problem description.
- (0) There is no problem description.

2. Context of the study. Options are:

- (1) If there is an explicit problem description for the research on the study, this problem description is supported by references to other work.
- (0.5) If there is a general problem description, this problem description is supported by references to other work.
- (0) There is no description of the context of the study.

For both options, reference may be just citations to other work, or there may be a description of the work related to the problem description.

3. Environment in which the study was carried out. Options are:

- (1) The authors provide an explicit description of the environment in which this research was performed. Examples are lab setting, as part of a project, in collaboration with industry, in collaboration with students, etc.
- (0.5) The authors provide some general words about the environment in which this research was performed.
- (0) There is no description of the environment.

4. Research design of the study refers to the way the study was organized. Options are:

- (2) The authors explicitly describe the research plan (different steps, timing, etc.) they have used to do the research, or the way the research was organized.
- (1) The authors provide some general words about the research plan or the way the research was organized.
- (0) There is no description of the research was organized.

5. Contributions of the study refers to the study results. Options are:

- (2) The authors explicitly list the contributions/results of the study.
- (1) The authors provide some general words about the results of the study.
- (0) There is no description of the research results.

6. Insights derived from the study. Options are:

- (2) The authors explicitly list the insights/lessons learned/findings from the study.
- (1) The authors provide some general words about the insights/lessons learned/findings from the study.
- (0) There is no description of the insights derived from the study.

7. Limitations of the study. Options are:

- (2) The authors explicitly list the limitations of/problems with the study.
- (1) The authors provide some general words about limitations of/problems with the study.
- (0) There is no description of the limitations of the study.

Table 21: Data collection form

Item ID	Field	Concern / research question
F1	Author(s)	Documentation
F2	Year	Documentation
F3	Title	Documentation
F4	Venue	Documentation
F5	Citation count	Documentation
F6	Quality score	RQ1-3
F8	Subject of the paper	RQ1
F7	Application domain (if applicable)	RQ1
F9	Quality Concerns	RQ1-2
F10	Claimed benefits	RQ1-2
F11	Tradeoffs	RQ2
F12	Assessment / validation approach	RQ3
F13	Evidence level	RQ3
F14	Repeatability	RQ3
F15	Software engineering field (if applicable)	RQ1
F16	Setting of validation.	RQ3

B Appendix

B.1 List of studies

ID	Year	Title	Author
adaptive-02	2010	An Overall Process for Self-Adaptive Pervasive Systems	A. Bucchiarone et al.
adaptive-03	2011	Adaptive Mobile Web Applications Through Fine-Grained Progressive Enhancement	H.Desruelle et al.
adaptive-05	2010	Incremental Online Evolution and Adaptation of Neural Networks for Robot Control in Dynamic Environments	F.Schlachter et al.
adaptive-06	2010	Adaptability Support in Time and Space Partitioned Aerospace Systems	Joao Craveiro and Jose Rufino
adaptive-08	2010	Efficiency Testing of Self-adapting Systems by Learning of Event Sequences	J. Hudson et al.
adaptive-12	2012	Using Role-Based Composition to Support Unanticipated, Dynamic Adaptation - Smart Application Grids	C. Piechnick et al.
adaptive-10	2012	A QoS Optimization Model for Service Composition	S. De Gyves Avila and K. Djemame
adaptive-11	2012	Self-Adaptive Framework for Modular and Self-Reconfigurable Robotic Systems	E. Meister et al.
ASE-01	2000	A Declarative Approach for Designing and Developing Adaptive Components	P.Boinot et al.
ASE-08	2010	PLASMA: A Plan-based Layered Architecture for Software Model-driven Adaptation	H. Tajalli et al.
ASE-10	2011	Self-Adaptive Software Meets Control Theory: A Preliminary Approach Supporting Reliability Requirements	A.Filieri et al
ASE-12	2007	Behavioral Adaptation of Component Compositions based on Process Algebra Encodings	R. Mateescu et al.
ASE-22	2006	An Automated Formal Approach to Managing Dynamic Reconfiguration	I.Warren et al.
DEAS-01	2005	Architectural Design of a Distributed Application with Autonomic Quality Requirements	D.Weyns et al.

ID	Year	Title	Author
DEAS-02	2005	From Product Lines to Self-Managed Systems: An Architecture-Based Runtime Reconfiguration Framework	M.Kim et al.
FSE-01	2010	FUSION: a framework for engineering self-tuning self-adaptive software systems	A.Elkhodary et al.
FSE-06	2011	Taming Uncertainty in Self-Adaptive Software	N. Esfahani et al.
FSE-08	2008	Model-Based Fault Detection in Context-Aware Adaptive Applications	M.Sama et al
ICAC-01	2006	Autonomic Live Adaptation of Virtual Computational Environmentsina Multi Domain-Infrastructure	P.Ruth et al.
ICAC-02	2004	A Component Based Programming Framework for Autonomic Applications	Hua Liu and Manish Parashar
ICAC-03	2005	Design and Evaluation of an Autonomic Workflow Engine	T.Heinis et al.
ICAC-07	2008	Automatic Configuration of an Autonomic Controller: An Experimental Study with Zero-Configuration Policies	T.Heinis and C.Pautasso
ICAC-10	2009	Self-Correlating Predictive Information Tracking for Large-Scale Production Systems	Y. Zhao et al.
ICAC-12	2010	FORMS: a FOrmal Reference Model for Self-adaptation	D.Weyns et al.
ICAC-16	2011	SILENCE: Distributed Adaptive Sampling for Sensor-based Autonomic Systems	E.Kyung Lee et al.
ICAC-18	2005	Towards Autonomic Virtual Applications in the In-VIGO System	Jing Xu et al.
ICAC-19	2006	Resource Management in the Autonomic Service-Oriented Architecture	J. Almeida et al.
ICAC-26	2005	Automated and Adaptive Threshold Setting: Enabling Technology for Autonomy and Self-Management	D.Breitgand et al.
ICAC-27	2007	Autonomic Reactive Systems via Online Learning	Sanjit A. Seshia
ICAC-28	2008	Multi-Level Intrusion Detection System (ML-IDS)	Y. Al-Nashif et al.

ID	Year	Title	Author
ICAC-29	2008	Generating Adaptation Policies for Multi-Tier Applications in Consolidated Server Environments	G. Jung et al.
ICSE-01	2003	Using Process Technology to Control and Coordinate Software Adaptation	G.Valetto and G.Kaiser
ICSE-03	2009	Using Quantitative Analysis to Implement Autonomic IT Systems	Radu Calinescu and Marta Kwiatkowska
ICSE-04	2009	Model Evolution by Run-Time Parameter Adaptation	I.Epifani et al.
ICSE-05	2009	Taming Dynamically Adaptive Systems Using Models and Aspects	B.Morin et al.
ICSE-08	2004	Autonomous Adaptation to Dynamic Availability Using a Service-Oriented Component Model	H.Cervantes and R.S.Hall
ICSE-10	2006	Symbolic Invariant Verification for Systems with Dynamic Structural Adaptation	B.Becker et al.
ICSE-15	2008	Recommending Adaptive Changes for Framework Evolution	B.Dagenais and Martin P. Robillard
ICSE-17	2006	Model-Based Development of Dynamically Adaptive Software	Ji Zhang and Betty H.C. Cheng
ICSE-20	2012	Application of Self-Adaptive Techniques to Federated Authorization Models	Christopher Bailey
JSS-01	2000	Kendra: adaptive Internet system	J.A. McCann et al.
JSS-02	2006	Developing adaptive systems with synchronized architectures q	T.Seceleanu and D.Garlan
JSS-03	2009	Autonomous mobile agent routing for efficient server resource allocation	v.Baousis et al.
JSS-04	2008	Using ontologies and Web services for content adaptation in Ubiquitous Computing	M.Forte et al.
JSS-05	2008	The architecture of an event correlation service for adaptive middleware-based applications	Y.Liu et al.
JSS-06	2011	Self-adapting workflow reconfiguration	R. Baird et al.
JSS-07	2011	A context-aware reflective middleware framework for distributed real-time and embedded systems	Shengpu Liu and Liang Cheng

ID	Year	Title	Author
JSS-09	2001	Synchronization and flow adaptation schemes for reliable multiple-stream transmission in multimedia presentations	C. M. Huang et al.
JSS-10	2010	Multi-layer faults in the architectures of mobile, context-aware adaptive applications	M. Sama et al.
JSS-11	2006	Monitoring techniques for an online neuro-adaptive controller	Y.Liu et al.
JSS-113	2007	A comprehensive approach to model and use context for adapting applications in pervasive environments	T.Chaari et al.
JSS-117	2008	An architectural approach to the correct and automatic assembly of evolving component-based systems	P. Pelliccione et al.
JSS-12	2008	An experimental study of adaptive testing for software reliability assessment	K.Y. Cai et al.
JSS-162	2005	Adaptive video transcoding and streaming over wireless channels	Z. Lei and D. Georganas
JSS-165	2005	An agent based adaptive bandwidth allocation scheme for multimedia applications	S.S. Manvi and P. Venkataram
JSS-198	2007	COCOA: COnversation-based service COmposition in pervAsive computing environments with QoS support	S.B. Mokhtar et al.
JSS-20	2004	An adaptable vertical partitioning method in distributed systems	J.H Son and M. Ho Kim
JSS-21	2006	Using temporal logic to specify adaptive program semantics	Ji Zhang and Betty H.C. Cheng
JSS-22	2006	On the efficiency and performance evaluation of the bandwidth clustering scheme for adaptive and reliable resource allocation	C. X. Mavromoustakis and Helen D. Karatza
JSS-24	2011	Dynamic deployment and quality adaptation for mobile augmented reality applications	T.Verbelen et al.
JSS-25	2009	Quality attribute tradeoff through adaptive architectures at runtime	J. Yang et al.
JSS-27	2009	Autonomic QoS control in enterprise Grid environments using online simulation	R. Nou et al.
JSS-28	2011	Toward architecture-based context-aware deployment and adaptation	N. Gui et al.

ID	Year	Title	Author
JSS-29	2010	Automated assembly of Internet-scale software systems involving autonomous agents	W. Jiao et al.
JSS-30	2008	A language for high-level description of adaptive web systems	S.Mohtasham and A. Ghorbani
JSS-31	2000	Dynamic adaptation of sharing granularity in DSM systems	A. Itzkovitz et al.
JSS-32	2008	A self-stabilizing autonomic recoverer for eventual Byzantine software	O. Brukman et al.
JSS-33	2011	Ontology driven bees foraging approach based self adaptive online recommendation system	V. Mohanraj et al.
JSS-89	2008	Virtualization-based autonomic resource management for multi-tier Web applications in shared data center	X. Wang et al.
JSS-46	2012	A Self-adaptive hierarchical monitoring mechanism for Clouds	G. Katsaros et al.
JSS-45	2012	QoS and energy management with Petri nets: A self-adaptive framework	D. Perez-Palacin et al.
JSS-44	2012	A development framework and methodology for self-adapting applications in ubiquitous computing environments	S. Hallsteinsen et al.
JSS-43	2012	Achieving dynamic adaptation via management and interpretation of runtime models	M. Amoui et al.
JSS-42	2012	Adaptive application offloading using distributed abstract class graphs in mobile environments	E. Abebe and C. Ryan
JSS-41	2012	Analysing monitoring and switching problems for adaptive systems	M. Salifu et al.
JSS-40	2012	An evaluation of multi-model self-managing control schemes for adaptive performance management of software systems	T. Patikirikorala et al.
JSS-50	2012	An adaptive model-free resource and power management approach for multi-tier cloud environments	X. Wang et al.
JSS-51	2012	A feedback-based decentralised coordination model for distributed open real-time systems	L. Nogueira et al.

ID	Year	Title	Author
JSS-47	2012	Stitch: A language for architecture-based self-adaptation	S. Wen Cheng and D. Garlan
JSS-49	2012	Adam: Identifying defects in context-aware adaptation	Chang Xu et al.
SASO-03	2011	Controlling the learning dynamics of interacting self-adapting systems	N. Rosemann et al.
SASO-05	2010	Runtime Behavior Monitoring and Self-Adaptation in Service-Oriented Systems	H. Psaier et al.
SASO-13	2010	Self Adaptation of Cooperation in Multi-Agent Content Sharing Systems	S.M. Allen et al.
SASO-16	2009	Study of Self-adaptation Mechanisms in a Swarm of Logistic Agents	R. Charrier et al.
SASO-17	2010	Toward Physically-Adaptive Computing	Kenneth M. Zick and John P. Hayes
SASO-18	2011	Towards Self-Adaptation in Real-time, Networked Systems: Efficient Solving of System Constraints for Automotive Embedded Systems	M. Zeller et al.
SASO-19	2008	A Simulator for Self-Adaptive Energy Demand Management	Y. Guo et al.
SASO-20	2009	A Completely Evolvable Genotype-Phenotype Mapping for Evolutionary Robotics	L. Konig and H. Schmeck
SEAMS-01	2009	The Design of a Self-healing Composition Cycle for Web Services	K.S. May Chan and Judith Bishop
SEAMS-02	2009	Evaluating the Effectiveness of the Rainbow Self-Adaptive System	Shang-Wen Cheng et al.
SEAMS-03	2009	Architecture-Driven Self-Adaptation and Self-Management in Robotics Systems	G. Edwards et al.
SEAMS-06	2009	A Formal Model for Self-Adaptive and Self-Healing Organizations	R. Haesevoets et al.
SEAMS-07	2009	Reinforcement Learning-Based Dynamic Adaptation Planning Method for Architecture-based Self-Managed Software	Dongsun Kim and Sooyong Park
SEAMS-08	2009	On Exploiting Decentralized Bio-inspired Self-organization Algorithms to Develop Real Systems	E. di Nitto et al.

ID	Year	Title	Author
SEAMS-09	2009	Engineering Adaptive Requirements	A. Qureshi and A. Perini
SEAMS-10	2009	Using Dynamic Workflows for Coordinating Self-adaptation of Software Systems	da Silva and R. de Lemos
SEAMS-16	2007	Mixed-Mode Adaptation in Distributed Systems: A Case Study	K. N. Biyani et al.
SEAMS-17	2007	Designing Run-Time Fault-Tolerance Using Dynamic Updates	Ali Ebnenasir
SEAMS-18	2007	Digitally Evolving Models for Dynamically Adaptive Systems	J. Goldsby et al.
SEAMS-19	2007	Implementing Adaptive Performance Management in Server Applications	Yan Liu and Ian Gorton
SEAMS-23	2008	Modeling Collaborations with Dynamic Structural Adaptation in Mechatronic UML	M. Hirsch et al.
SEAMS-24	2008	Towards Goal-Oriented Development of Self-Adaptive Systems	M. Morandini ET AL.
SEAMS-25	2008	Monitoring Multi-tier Clustered Systems With Invariant Metric Relationships	M. A. Munawar et al.
SEAMS-27	2008	From Goals To Components: A Combined Approach To Self-Management	D. Sykes et al.
SEAMS-28	2006	Towards Specification, Modelling and Analysis of Fault Tolerance in Self Managed Systems	Jeff Magee and Tom Maibaum
SEAMS-29	2010	Starfish: Policy Driven Self-Management in Wireless Sensor Networks	T. Bourdenas
SEAMS-33	2010	Towards Pro-active Adaptation with Confidence Augmenting Service Monitoring with Online Testing	A. Metzger et al
SEAMS-35	2010	Adaptation and Abstract Runtime Models	Thomas Vogel and Holger Giese
SEAMS-36	2010	On Decentralized Self-Adaptation: Lessons from the Trenches and Challenges for the Future	D. Weyns et al.
SEAMS-37	2011	Model-based Self-Adaptive Resource Allocation in Virtualized Environments	N. Huber et al.
SEAMS-38	2011	FlashMob: Distributed Adaptive Self-Assembly	D. Sykes et al

ID	Year	Title	Author
SEAMS-39	2011	GRAF: Graph-based Runtime Adaptation Framework	M. Derakhshanmanesh et al.
SEAMS-40	2011	Pairwise Testing of Dynamic Composite Services	A.Kattepur et al.
SEAMS-41	2011	Dynamic Plans for Integration Testing of Self-adaptive Software Systems	da Silva and R. de Lemos
SEAMS-43	2011	A Self-Adaptive Deployment Framework for Service-Oriented Systems	S.Burg and E. Dolstra
SEAMS-44	2011	Inflation and Deflation of Self-Adaptive Applications	W. Moore and R. Childers
SEAMS-45	2006	Verifying the Adaptation Behavior of Embedded Systems	K. Schneider et al.
SEAMS-47	2009	Reflecting on Self-Adaptive Software Systems	J. Andersson et al.
SEAMS-48	2009	StarMX: A Framework for Developing Self-Managing Java-based Systems	R. Asadollahi et al.
SEAMS-49	2009	Behavioural Self-Adaptation of Services in Ubiquitous Computing Environments	J. Camara et al.
SEAMS-50	2006	Experience and Prospects for Various Control Strategies for Self-Replicating Multi-Agent Systems	J.-P. Briot et al.
SEAMS-56	2008	Scheduling Time-bounded Dynamic Software Adaptation	S. Fritsch et al.
SEAMS-58	2008	Endogenous Versus Exogenous Self-Management	D.Weyns et al.
SEAMS-59	2011	Application of Software Health Management Techniques	N. Mahadevan et al.
SEAMS-61	2011	On the Performance of UML State Machine Interpretation at Runtime	E. Hofig et al.
SEAMS-73	2012	Evaluation of Resilience in Self-Adaptive Systems Using Probabilistic Model-Checking	J CÃ¡mara and R de Lemos
SEAMS-74	2012	Timing Constraints for Runtime Adaptation in Real-Time, Networked Embedded Systems	M. Zeller and C. Prehofer
SEAMS-72	2012	Reliability-Driven Dynamic Binding via Feedback Control	A. Filieri et al.

ID	Year	Title	Author
SEAMS-71	2012	Traffic Routing for Evaluating Self-Adaptation	J. Wuttke et al.
SEAMS-70	2012	Dynamic Self-Adaptation for Distributed Service-Oriented Transactions	H. Gomaa and K. Hashimoto
ISARCS-01	2010	Reactive model-based control of reconfiguration in the Fractal component-based model	Gwenael Delaval, Eric Rutten
ISARCS-02	2010	Using QoS-Contracts to Drive Architecture-Centric Self-adaptation	F. Chauvel et al.
ISARCS-03	2010	A Self-healing Component Sandbox for Untrustworthy Third Party Code Execution	Kiev Gama and Didier Donsez
SefSAS-04	2009	Improving Architecture-Based Self-Adaptation Through Resource Prediction	Shang-Wen Cheng et al.
SefSAS-05	2009	Policy-Based Architectural Adaptation Management: Robotics Domain Case Studies	John C. Georgas and Richard N. Taylor
SefSAS-06	2009	A Case Study in Goal-Driven Architectural Adaptation	W. Heaven et al.
SefSAS-09	2009	Using Architecture Models to Support the Generation and Operation of Component-based Adaptive Systems	Nelly Bencomo and Gordon Blair
SefSAS-11	2009	Using Filtered Cartesian Flattening and Microrebooting to Build Enterprise Applications with Self-adaptive Healing	J. White et al.
SefSAS-12	2009	Modeling of Context-Aware Self-Adaptive Applications in Ubiquitous and Service-Oriented Environments	K. Geihs et al.
TAAS-02	2011	An Adaptive and Predictive Approach for Autonomic Multirate Multicast Networks	N. XIONG et al.
TAAS-04	2011	AutoHome: An Autonomic Management Framework for Pervasive Home Applications	J. BOURCIER et al.
TAAS-05	2008	A New Class of Nature-Inspired Algorithms for Self-Adaptive Peer-to-Peer Computing	STEVEN Y. KO and I.GUPTA
TAAS-06	2009	M-AID: An Adaptive Middleware Built Upon Anomaly Detectors for Intrusion Detection and Rational Response	Z.ZHANG
TAAS-07	2008	An Adaptive Automatically Tuning Intrusion Detection System	ZHENWEI YU and J. P. TSAI

ID	Year	Title	Author
TAAS-09	2011	Cooperative Evolution of Services in Ubiquitous Computing Environments	D.TACCONI
TAAS-12	2007	Adaptive Correctness Monitoring for Wireless Sensor Networks Using Hierarchical Distributed Run-Time Invariant Checking	D.HERBERT et al.
TAAS-15	2012	A Formalized, Taxonomy-Driven Approach to Cross-Layer Application Adaptation	R. POPESCU et al
TAAS-19	2012	3PC: System Support for Adaptive Peer-to-Peer Pervasive Computing	M. HANDTE et al.
TAAS-18	2012	Adaptare: Supporting Automatic and Dependable Adaptation in Dynamic Environments	M. DIXIT et al.
TAAS-17	2012	FORMS: Unifying Reference Model for Formal Specification of Distributed Self-Adaptive Systems	D.Weyns et al.
TAAS-16	2012	Reflective Pervasive Systems	N. SERBEDZIJA et al.
TOSEM-01	2011	Recommending Adaptive Changes for Framework Evolution	B.Dagenais and Martin P. Robillard
TOSEM-02	2009	Programming Pervasive and Mobile Computing Applications:the TOTA Approach	Marco Mamei and Franco Zambonelli
TSE-01	2008	Model-Based Adaptation of Behavioral Mismatching Components	C.Canal et al.
TSE-02	2001	Dynamic Adaptation and Deployment of Distributed Components In Hadas	Israel Ben-Shaul et al.
TSE-03	2006	The SATIN Component System A Metamodel for Engineering Adaptable Mobile Systems	S.Zachariadis et al.
TSE-06	2003	CARISMA: Context-Aware Reflective mIddleware System for Mobile Applications	L.Capra et al.
TSE-07	2008	Dynamic QoS Adaptation for Mobile Middleware	S.N.Chuang et al
TSE-30	2011	An Autonomous Engine for Services Configuration and Deployment	F.Cuadrado et al.
TSE-31	2011	MOSES: a Framework for QoS Driven Runtime Adaptation of Service-oriented Systems	V.Cardellini et al.

ID	Year	Title	Author
TSE-34	2010	Context-Aware Adaptive Applications: Fault Patterns and Their Automated Identification	M.Sama et al.
TSE-36	2011	Dynamic QoS Management and Optimization in Service-Based Systems	R.Calinescu et al
TSE-38	2011	Adaptation of Service Protocols using Process Algebra and On-the-Fly Reduction Techniques	R.Mateescu et al.
TSE-11	2012	Self-management of Adaptable Component-based Applications	L. Rosa et al.
TSE-13	2012	A Decentralized Self-Adaptation Mechanism For Service-Based Applications in The Cloud	Vivek Nallur, Rami Bahsoon
TSE-12	2012	Domain-Specific Service Selection for Composite Services	O. Moser et al.
TSE-10	2012	Adaptation of Service Protocols Using Process Algebra and On-the-Fly Reduction Techniques	R. Mateescu et al.
WADS-01	2005	Specifying Adaptation Semantics	Ji Zhang and Betty H.C. Cheng
WADS-02	2005	Architectural Runtime Configuration Management in Support of Dependable Self-Adaptive Software	J. C. Georgas et al.
WADS-04	2004	Adding Safeness to Dynamic Adaptation Techniques	J. Zhang et al.
WADS-05	2003	A Framework for Using Component Redundancy for self-Optimising and self-Healing Component Based Systems	Ada Diaconescu and John Murphy
WICSA-02	2004	An Architecture for Coordinating Multiple Self-Management Systems	S. Wen Cheng et al.
WICSA-03	2009	Self-Repairing Systems Modeling and Verification using AGG	A.Bucchiarone et al.
WOSS-09	2002	Self-Adapting Concurrency: The DMonA Architecture	Sam Michiels and Lieven Desmet
WOSS-10	2004	Resource-based Approach to Feature Interaction in Adaptive Software	J.Bisbal and B.H.C. Cheng
WOSS-11	2004	Towards a Knowledge-Based Approach to Architectural Adaptation Management	J. C. Georgas and Richard N. Taylor

ID	**Year**	**Title**	**Author**
WOSS-13	2002	Model-based Adaptation for Self-Healing Systems	D.Garlan and B.Schmerl
ICSM-01	2002	Using Service Specific Proxies to Migrate Web Services to the Wireless Web: an Illustrative Case Study of Adaptive Maintenance	Warren Harrison
ICSM-02	2008	Characterizing Maintainability Concerns in Autonomic Element Design	Q. Zhu et al.
ICSM-05	2008	Supporting Software Evolution Using Adaptive Change Propagation Heuristics	Haroon Malik and Ahmed E. Hassan
ICSM-06	2010	A Lightweight Transformational Approach to Support Large Scale Adaptive Changes	M. L. Collard et al.

Linnæus University
School of Computer Science, Physics and Mathematics

SE-391 82 Kalmar / SE-351 95 Växjö
Tel +46 (0)772-28 80 00
dfm@lnu.se
Lnu.se/dfm