# Claims and Supporting Evidence for Self-Adaptive Systems: A Literature Study

Danny Weyns[1], M. Usman Iftikhar[1], Sam Malek[2], Jesper Andersson[1]
[1]*Linnaeus University, Växjö, Sweden*
[2]*George Mason University, Fairfax VA, USA*
*Contact: danny.weyns@lnu.se*

***Abstract*—Despite the vast body of work on self-adaption, no systematic study has been performed on the claims associated with self-adaptation and the evidence that exists for these claims. As such an insight is crucial for researchers and engineers, we performed a literature study of the research results from SEAMS since 2006 and the associated Dagstuhl seminar in 2008. The study shows that the primary claims of self-adaptation are improved flexibility, reliability, and performance of the system. On the other hand, the tradeoffs implied by self-adaptation have not received much attention. Evidence is obtained from basic examples, or simply lacking. Few systematic empirical studies have been performed, and no industrial evidence is reported. From the study, we offer the following recommendations to move the field forward: to improve evaluation, researchers should make their assessment methods, tools and data publicly available; to deal with poor discussion of limitations, conferences/workshops should require an explicit section on limitations in engineering papers; to improve poor treatment of tradeoffs, this aspect should be an explicit subject of reviews; and finally, to enhance industrial validation, the best academy-industry efforts could be formally recognized by the community.**[1]

## I. Introduction

Self-adaptation is widely recognized as a key approach to tackle some of the hard challenges of engineering and managing complex distributed software systems [1], [2], [3]. 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 goals. Examples are a system that heals itself when certain errors occur, or a system that optimizes its performance under changing conditions.

Different loosely connected communities have studied self-adaptation. Prominent examples are the SEAMS community (International Symposium on Software Engineering for Adaptive and Self-Managing Systems, http://www.self-adaptive.org/), the ICAC community (International Conference on Autonomic Computing, http://www.autonomic-conference.org/), and the SASO community (Self-Adaptive and Self-Organizing Systems, http://www.saso-conference.org/). The co-existence of different communities has led to the exploration of the field from different perspectives. However, it has also led to fragmentation of the field and the development of different vocabularies. In this paper, we take a broad perspective on what is considered as a self-adaptive system. In particular, we consider a self-adaptive system as a system consisting of two parts: a *managed system* and a *managing system*. The managed system is situated in an environment and provides some functionality to users. The managing system comprises the software to monitor the managed system and its environment and performs adaptations of the managed system when needed. Note that the software of the managed system and the software of the managing system may be clearly partitioned in separate modules or layers, or the software of the two parts may be interwoven.

### A. Problem Description and Objectives

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. Several survey articles have been written over the last years, e.g. [4], [5], [6], [7]. These articles summarize achievements of the field and outline challenges for future work. However, to the best of our knowledge, no systematic study has been performed on the claims associated with self-adaptation and the evidence that exists for these claims. As a result, there is no clear view on how self-adaptation actually contributes to tackling the challenges of engineering and managing complex software systems. However, such an insight is crucial for researchers and engineers. Our objective is to study and summarize existing research related to engineering self-adaptive software systems and shed light on what the claimed benefits are of self-adaption and to what extent evidence exists for these benefits. In particular, we aim to:

1) identify the focus of current research in engineering self-adaptive systems, providing context for the study,
2) assess the quality of current research studies,
3) understand the claims made for self-adaptation and the available evidence for these claims, and
4) identify limitations of current approaches and outline potential areas for future work in the field, helping to derive conclusions from the study.

[1]This research is supported by grant FP7-PEOPLE-2011-CIG 303791, and grants CCF-0820060 and CCF-1217503 from NSF and N11AP20025 from DARPA. The authors are grateful to Masuma Naqvi and the 45 Master students at Linnaeus University that have participated in this study.

From the study, we offer recommendations to move the field forward. We focus in this study on the research results of SEAMS since 2006 and the Dagstuhl seminar in 2008 (http://www.self-adaptive.org/). As such, the study will cover a representative sample of the literature, which allows deriving recommendations to move the field forward. However, a complete systematic literature review [8] is required to confirm whether the findings are conclusive for the whole field.

### B. Paper Overview

In Section 2 we discuss the method we used in our study. We explain the research questions we address, discuss the criteria to assess the quality of studies, and summarize the data items that were collected. In section 3 we present the data extracted from the studies, and interpret this data answering the research questions. Section 4 discusses limitations of our study, compares our findings with related studies, and presents recommendations to move our field forward. Finally, we wrap up and conclude in section 5.

## II. Research Method

Our study comprises the primary steps of a systematic literature review [8], which is a well-defined approach to identify, evaluate and interpret all relevant studies regarding a particular research question, topic area or phenomenon of interest. The major difference in comparison to a systematic literature review is that we restricted the scope of the study upfront, i.e, we studied the papers published at SEAMS between 2006 and 2011 and the papers of the 2008 SefSAS book (www.springer.com/computer/swe/book/978-3-642-02160-2). Since our primary goal is to understand the claims and supporting evidence of self-adaptation, we excluded papers about theoretical aspects, as well as surveys and roadmap papers. We also excluded short papers of 1 or 2 pages. Overall, we studied 96 papers out of a total of 124 papers. Figure 1 shows an overview of the three-phased method we applied in the study.

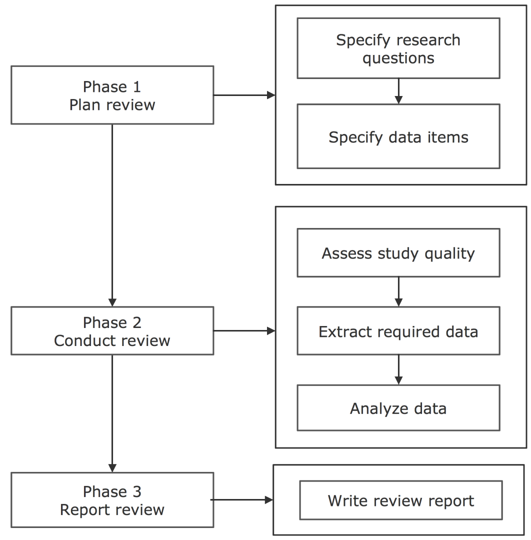


Figure 1. Overview of the review process (adapted from [9]).

In total four researchers that are active in the field of self-adaptive systems and 45 master students[2] were involved in the review. The review planning (Phase 1) was performed by the researchers. In particular, two researchers defined initial research questions, together with the data items that had to be collected. Two other researchers checked the questions and data items. The feedback was used to make small adaptations of questions and data items. Subsequently, the researchers supported by the master students conducted the review (Phase 2). Each study was assigned to two researchers and two groups of two students. For each study, we organized a 20 minutes discussion session where all assigned reviewers came together with the data they collected from the study. During these discussions sessions, a decision for each data item was taken, based on consensus. The data was then added to a database for subsequent analysis, which was performed by the researchers. Finally, the group of researchers prepared this final report (Phase 3). All the material that was used for the study is available at http://homepage.lnu.se/staff/daweaa/SLR-SEAMS.htm.

In the remainder of this section, we present the research questions for the study, and summarize the quality assessment criteria and data items collected for the study. We also briefly explain how we performed data analysis.

### A. Research Questions

We formulated the goal of the study through Goal-Question-Metric (GQM) perspectives (purpose, issue, object, viewpoint) [10]. The general goal of our study is:

*Purpose:* Understand and characterize
*Issue:* the claims and supporting evidence
*Object:* of self-adaptive software systems
*Viewpoint:* from a researchers and engineers viewpoint.

The general research question translates to several concrete research questions. In detail, our study covers the following research questions:

**RQ1:** What is the focus of research in self-adaptive software systems?
**RQ2:** What are the claimed benefits of self-adaptation and what are the tradeoffs implied by self-adaptation?
**RQ3:** What assessment methods have been used and how much evidence is available?
**RQ4:** What are the limitations of the existing approaches?

[2]Students involved were last year master students in software engineering that took preparatory courses on software quality and self-adaptive systems. To be allowed to participate in the study, the students had to pass an exam in which they demonstrated solid knowledge in the area of self-adaptive systems and sufficient understanding of the data items used in the study.

**RQ5:** What are interesting areas for future research in the field?

RQ1 is motivated by the need to get insight in the research trends in self-adaptive systems, providing context for the study. RQ2 and RQ3 are the central research questions in this study. The goal of RQ4 and RQ5 is to help deriving conclusions from the study.

### B. Quality Criteria

All studies are assessed through a quality check, which is an inherent part of a thorough literature study. Checking the quality of the studies is important for data synthesis and interpretation of results later on. To assess the quality, we collected a set of quality items as show on Table I. These items are based on the assess method for research studies proposed in [11]. 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).

### C. Data Items

For each study, the data items shown in Table II were collected. Six reviewers read each paper in detail and extracted the data in a form. This data was used during a discussion to resolve conflicts and reach consensus for the data items.

The data items author, year, title, and keywords (F1-F4) were used for documentation purposes. Citation count (F5) is the number of references provided by Google Scholar in December 2011. Quality score (F6) is the sum of the scores for the quality criteria described in section II-B.

Category of the study (F7) has the following options:

- theory (study of a new theory about any aspects of self-adaptation)
- engineering (study about any aspects of the development, operation, or maintenance of self-adaptive software systems)
- application (study about a particular application in which self-adaptation is applied)
- empirical study (systematical empirical inquiry, including case study, controlled laboratory experiment, and controlled experiment performed in industry setting)
- survey (study of a collection of data about any aspects of self-adaptation, typically, by reviewing literature in the field of self-adaptation)
- other.

Subject of the study (F8) refers to the general theme of the study. Options are:

- concepts (study about new ideas/abstractions of self-adaptation)
- models (study of a new model or models that describe any aspects of self-adaptation)
- behavior (study about any aspects of the behavior of self-adaptive systems)
- requirements
- architecture
- framework (study about a software framework, i.e. software providing generic functionality that can be selectively changed or extended to provide application specific software)
- implementation
- testing
- verification
- other.

Concrete focus (F9) refers to the specific subject of the research. Options are:

- one or more of the activities of self-adaptation (i.e. monitoring of the managed system or the environment, analysis of the collected data, planning of adaptation, execution of adaptation)

Table I
ITEMS TO ASSESS STUDY QUALITY

| Quality item |
|---|
| 1. Problem definition of the study. Options are:<br>(2) The authors provide an explicit problem description for the study.<br>(1) The authors provide a general problem description.<br>(0) There is no problem description. |
| 2. Problem context of the study. Options are:<br>(1) If there is an explicit problem description for the research on the study, this problem description is supported by references.<br>(0.5) If there is a general problem description, this problem description is supported by references.<br>(0) There is no description of the problem context of the study. |
| 3. Environment in which the study was carried out. Options are:<br>(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.).<br>(0.5) The authors provide some general words about the environment in which this research was performed.<br>(0) There is no description of the environment. |
| 4. Research design of the study refers to the way the study was organized. Options are:<br>(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.<br>(1) The authors provide some general words about the research plan or the way the research was organized.<br>(0) There is no description of how the research was planned/organized. |
| 5. Contributions of the study refers to the study results. Options are:<br>(2) The authors explicitly list the contributions/results of the study.<br>(1) The authors provide some general words about the study results.<br>(0) There is no description of the research results. |
| 6. Insights derived from the study. Options are:<br>(2) The authors explicitly list insights/lessons learned from the study.<br>(1) The authors provide some general words about insights/lessons learned from the study.<br>(0) There is no description of the insights derived from the study. |
| 7. Limitations of the study. Options are:<br>(2) The authors explicitly list the limitations/problems with the study.<br>(1) The authors provide some general words about limitations/problems with the study.<br>(0) There is no description of the limitations of the study. |

Table II
DATA COLLECTION FORM

| Item ID | Field | Concern / research question |
|---|---|---|
| F1 | Author(s) | Documentation |
| F2 | Year | Documentation |
| F3 | Title | Documentation |
| F4 | Keywords | Documentation |
| F5 | Citation count | RQ3 |
| F6 | Quality Score | RQ3 |
| F7 | Category of the study | RQ1 |
| F8 | Subject of the study | RQ1 |
| F9 | Concrete focus of the study | RQ1 |
| F10 | Application domain | RQ1 |
| F11 | Disciplined split | RQ1 |
| F12 | Concerns | RQ2 |
| F13 | Claimed benefits | RQ2 |
| F14 | Tradeoffs | RQ2 |
| F15 | Assessment approach | RQ3 |
| F16 | Evidence level | RQ3 |
| F17 | Repeatability | RQ3 |
| F18 | Findings | RQ4 |
| F19 | Limitations | RQ4 |
| F20 | Challenges | RQ5 |

- runtime models to support adaptation (i.e., models that are reified at runtime to support self-adaption)
- reflection/meta-level computation (i.e., software that observes and modifies its own structure or behavior)
- multiple MAPE/control loops (including decentralization, coordination, etc.)
- other.

Applications domains (F10) for which self-adaption is used (in case the study is about a particular application) or applied for evaluation (in case the applications are used for illustration, evaluation, etc.). Possible application domains are: parallel computing (grid, parallel computing, cloud computing etc.), service-based systems (webservices, business applications, e-commerce, etc.), client-server systems, embedded systems, robotics, traffic and transportation, and other.

Disciplined split (F11) is one of:

- the managing system is clearly separated from the managed system; the latter comprises the application logic that provides the system's domain functionality
- the managing system is clearly separated from the managed system; the latter comprises resources/hardware - resources can be CPU cycles, memory, bandwidth, etc.; hardware can be an engine, a cooling installation, etc.
- managing and managed system are mixed; the software of both systems is partially or completely interwoven
- not applicable.

Concerns of self-adaption (F12) refers to what is of interest and influenced by self-adaption (positively or negatively). Options are:

(1) Quality concerns related to self-adaptation, i.e., quality attributes that are influenced 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 behavior, 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.

(2) Other concerns related to self-adaptation, i.e., any other concerns that are influenced by self-adaptation in the study. Examples are quality of the engineering artifacts, costs, benefits, etc.

Claimed benefits of self-adaptation (F13) can be one or more of the following (refers to concerns selected in F12):

- preserving quality of the software (i.e., quality attributes that are maintained by self-adaptation)
- improving quality of the software (i.e., quality attributes that are improved or added to the system by self-adaptation)
- assuring quality of the software (i.e., quality attributes that are guaranteed by self-adaptation, typically supported by strong evidence or formal proof)
- other concerns that are positively influenced by self-adaptation.

Tradeoffs (F14) refers to the concerns of self-adaptation with a negative impact (as selected in F12). This can be one or more of the following:

- quality concerns that are negatively influenced by self-adaptation
- other concerns that are negatively influenced by self-adaptation

Assessment/validation (F15) has the following options:

- discussion (a qualitative, textual, opinion-oriented evaluation; e.g., comparison, oral discussion of advantages

and disadvantages; the discussion may include the description of examples that may not have been realized)
- example application (assessment based on a concrete application or applications; the example application is concretely realized, but not necessarily completely described in the study)
- simulation (execution of a system with artificial data using a model of the real world; includes typically a comparison of the proposed approach with a baseline/benchmark)
- rigorous analysis (rigorous derivation and proof; rigorous analysis is typically based on formal methods)
- empirical study (systematical empirical inquiry to provide evidence for or against any particular theory or hypothesis by means of accurate analysis of data obtained from observation or experience)
- 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)
- none

Evidence level (F16) is one of the following options:

- no evidence
- evidence obtained from demonstration or application to simple/toy examples
- evidence obtained from expert opinions or observations
- evidence obtained from empirical studies
- industrial evidence
- other

Repeatability (F17) has the following options:

- 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 can be 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.

F18, F19, and F20 refer to the study as a whole. Findings (F18) briefly lists (i) the stated contributions of the study, and (ii) the stated insights/lessons learned derived from the study. Limitations (F19) enumerates the stated shortcomings or problems with the study, and Challenges (F20) lists open problems identified by the authors for future research.

### D. Data Analysis

The data derived from the the studies was collated and summarized to answer the research questions. The synthesis included the following:

1) listing of findings,
2) reaching consensus among reviewers in case of conflicting opinions,
3) analysis of findings,
4) answering research questions and interpretation of the results.

Based on the synthesis, we derived conclusions and recommendations for future research in the field. Finally, we reflected on the validity of the review.

## III. RESULTS

We now discuss the study results based on the research questions we defined for the study.

*RQ1: What is the focus of research in self-adaptive systems?*

Research focus is derived from the following concerns: category of the study (F7), subject (F8), concrete focus (F9), application domain (F10), and disciplined split (F11).

Overall, from the 124 studies, 90 studies focused on engineering and 6 studies on applications. Only these 96 studies, which make up 77 % of the total number of papers, were considered in the rest of the review. The remaining papers are theory studies (12), surveys (5), roadmap papers (2), and short papers (10).

Figure 2 shows the frequency of the subjects of the studies. Not surprisingly, architecture (26 %) together with models (24 %) account for half of the studies, as the roots of the field lay in architecture and runtime architectural models. Note that the specific attention for architecture has significantly decreased after 2008, while the inverse effect can be noticed for models. We also notice that there is little interest in formal verification since 2008, while there is a growing interest in frameworks, requirements, and testing.

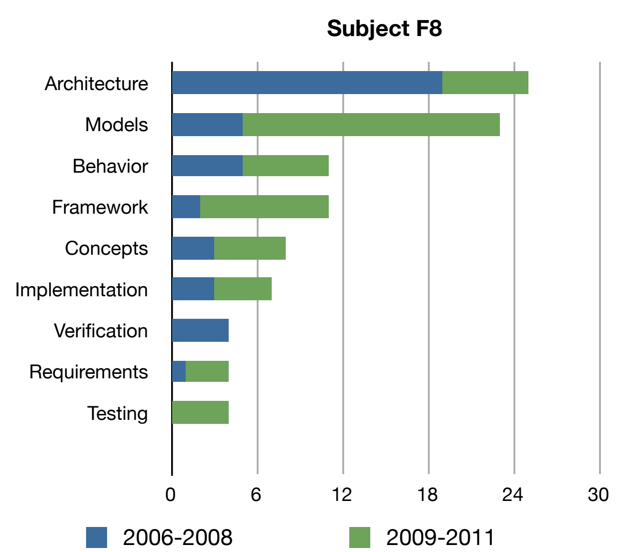


Figure 2. Subjects of the studies (number of studies).

Figure 3 zooms in on the concrete focus of the studies. 56 % of the studies focus on one or more activities of self-adaptation (monitoring, analyzing, planning, execution), and 18 % focus on runtime models. We notice a spectacular increase of interest in runtime models after 2008 (factor of 8); the connection with the organization of the models@runtime workshop seems evident[3]. 10 % of the studies focus on multiple control loops, and 4 % focus on reflection.

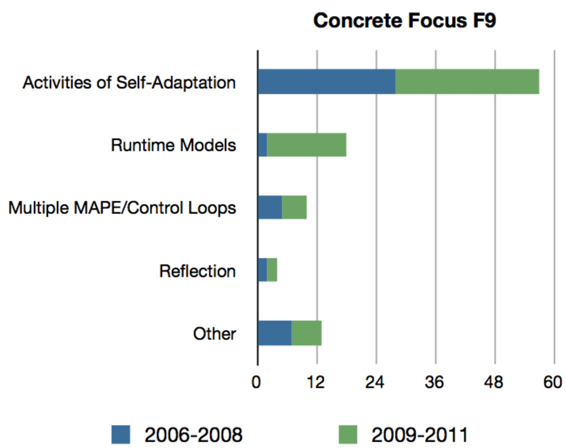


Figure 3. Concrete focus of the studies (number of studies).

Figure 4 shows the data extracted from the application domains for which self-adaptation has been used (F10). Our particular interest was in course grained families of systems. Service-based systems account for 34 % of the applications, and this portion is increasing. Dynamic service composition is a very active area for self-adaptation. Robotics, which has always been an important domain in self-adaptation research, accounts for 15 %.

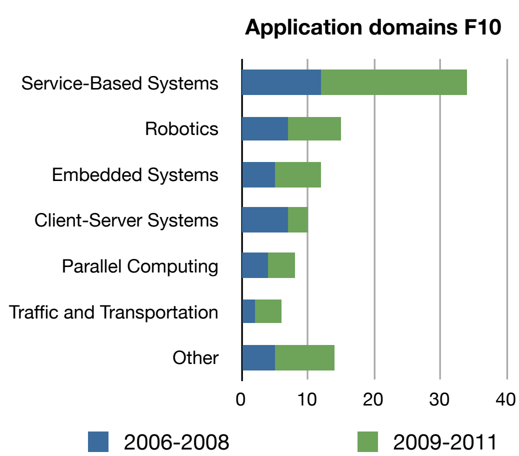


Figure 4. Application domains (number of studies).

We also looked at the separation of concerns between managed and managing system. The data extracted from disciplined split (F11) shows that 70 % of the studies make a strict separation between the managed system and the managing system. 56 % of the studies consider managed systems comprising of software, the remaining 14 % consider managed systems comprising of resources or hardware. In 30 % of the studies, the managed system and the managing system are interwoven. Whereas a strict separation of concerns may indicate disciplined engineering practice, a number of authors deviate from this principle due to the invasive nature of particular types of self-adaptation, and the use of self-organizing, agent-based approaches.

[3]http://www.comp.lancs.ac.uk/∼bencomo/MRT/

*RQ2: What are the claimed benefits of self-adaptation and what are the tradeoffs implied by self-adaptation?*

To answer this question, we used data extracted from concerns (F12), claimed benefits (F13), and tradeoffs (F14).

We assessed all the concerns that are of interest for self-adaption in the studies. Overall, we found that 94 % of the concerns were quality attributes. Top quality attributes are flexibility (31 %), reliability (29 %), and efficiency/performance (27 %). Particularly underrepresented are availabilty, accuracy, usability, and security that all together make up only 12 % of the reported concerns. The remaining 6 % other reported concerns include complexity, engineering effort, and budget.

Figure 5 summarizes the claims versus the tradeoffs of self-adaption. The figure clearly demonstrates that the papers mainly report on claimed benefits, while little attention is given to the implications of self-adaptation. 39 % of the studies considered a single concern, 47 % considered 2 concerns, the remaining 14 % considered 3 or 4 concerns. 85 % of the considered quality attributes are claimed to be positively influenced by self-adaption and the remaining 15 % 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 accuracy). We also assessed the type of claims that were made for the quality attributes and found that the dominant claim is improvement of software quality attributes (88 %). The remaining 12 % is equally distributed over preserving and assuring quality attributes. Note that the ratio claims versus tradeoffs for the other concerns show an inverse image as for the quality attributes, i.e., 70 % of the other concerns are negatively affected and 30 % positively.

*RQ3: What assessment methods have been used and how much evidence is available?*

To answer this question, we drew on data extracted from citation count (F5), quality scores (F6), assessment approach (F15), evidence level (F16), and repeatability (F17).

Citation counts shows an average of 16 citations for SEAMS papers from 2006 and 2008, and an average of 8 citations for 2007, 2009, and 2010. The SefSAS book has 4 papers with +30 citations. As it is well-known that it is difficult to derive conclusions from the number of citations

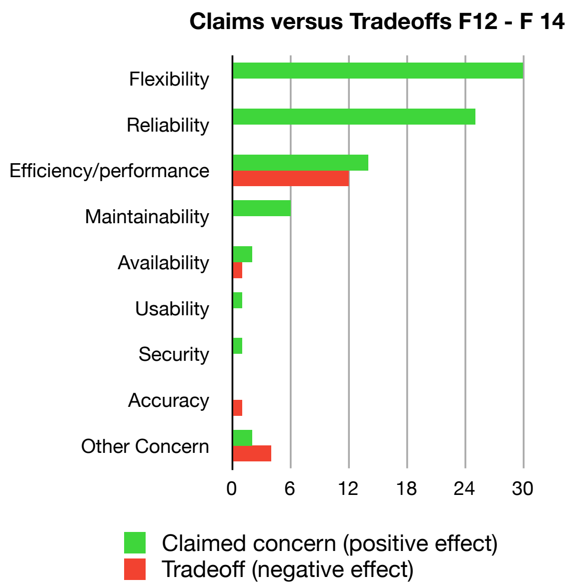


Figure 5. Claims versus tradeoffs of self-adaptation. Bars show the relative share (%) of the concerns w.r.t. to the total number of reported concerns.

of studies, we provide this information for the interest of the reader. Table III shows the studies with plus 30 citations, which might be an indication of high quality of the studies.

Table III
STUDIES WITH +30 CITATIONS.

| Citations | Study | Year |
|---|---|---|
| 168 | Software Engineering for Self-Adaptive Systems: A Research Roadmap | 2009 |
| 81 | Architecture-based Self-Adaptation in the Presence of Multiple Objectives | 2006 |
| 61 | Engineering Self-Adaptive Systems through Feedback Loops | 2009 |
| 51 | MUSIC: Middleware Support for Self-Adaptation in Ubiquitous and Service-Oriented Environments | 2009 |
| 48 | From Goals To Components: A Combined Approach To Self-Management | 2008 |
| 46 | Towards Goal-Oriented Development of Self-Adaptive Systems | 2008 |
| 34 | Modeling Dimensions of Self-Adaptive Software Systems | 2009 |
| 32 | Runtime Adaptation in a Service-Oriented Component Model | 2008 |

Figure 6 shows the frequency of the quality scores in percentages. Some important insights can be derived if we look at the scores of the individual quality properties. Problem definition and description of contributions have a very good score (only 3 % of the studies do not provide a problem description and 3 % do not provide a description of contributions) . However, the scores of the description of research design and limitations are very low (85 % do not describe research design and 66 % do not provide a description of limitations). These aspects can be given more attention when reporting on research results of self-adaptation. Note that the average of the total quality scores is 5.40 out of the maximum total of 12, which means that on average in terms of quality the studies are neither perfect, nor are they completely flawed.[4] Table III shows the studies with a quality score of plus 8.


Figure 6. Quality scores (percentages of studies).

Table IV
STUDIES WITH +8 QUALITY SCORE IN CHRONOLOGICAL ORDER.

| Study | Year |
|---|---|
| Architecture-based Self-Adaptation in the Presence of Multiple Objectives | 2006 |
| The WSDM of Autonomic Computing: Experiences in Implementing Autonomic Web Services | 2007 |
| From Goals To Components: A Combined Approach To Self-Management | 2008 |
| Endogenous Versus Exogenous Self-Management | 2008 |
| Live Goals for Adaptive Service Compositions | 2010 |
| Towards Pro-active Adaptation with Confidence Augmenting Service Monitoring with Online Testing | 2010 |
| Model-based Self-Adaptive Resource Allocation in Virtualized Environments | 2011 |

Figure 7 shows the assessment methods used in the studies. Example application and discussion make up 90 % of the used assessments. Only a few (quasi) empirical studies[5] and no assessment on real applications are reported.

[4]The total quality score of a study is the sum of the quality scores for all the quality items, as described in Table I.

[5]With empirical studies we refer to systematic investigations, including surveys (investigation of a topic based on data derived from interviews or questionnaires from a representative sample of the target population), case studies (investigation of a topic by observational investigation and statistical analysis of collected data), and experiments (a rigorous and controlled investigation in which subjects are assigned to different treatments at random and the effects are determined based on statistical analysis of collected data).

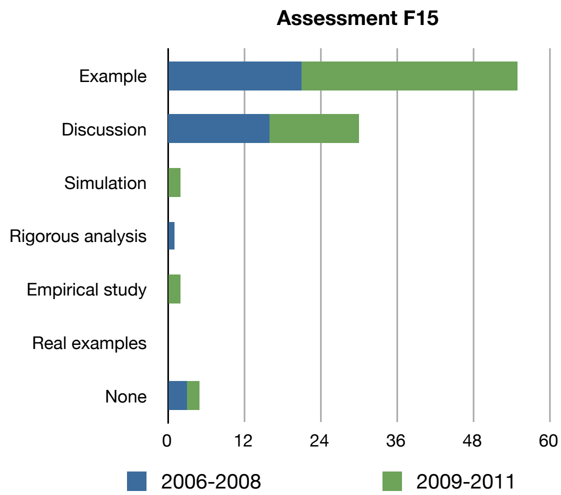


Figure 7. Assessment methods (number of studies).

Given the used assessment methods, it is not surprising that most studies have a low evidence level. The data collected for evidence level (F16) shows that evidence is only obtained from simple/toy examples (51 %) and discussion (49 %). On the other hand, there is no study that shows industrial evidence.

Repeatability is considered as a foundation for quality research, as it allows to test and verify research results. Figure 8 shows the results for repeatability of the studies. About 39 % of the reported studies provide no useful information to repeat the study. 45 % of the studies provide a partial description and 14 % provide partial material to repeat the study. Only 2 studies make all the necessary material available to repeat the study. Based on these results, we conclude that it would be beneficial for the community to make tools and results used for evaluations publicly available to facilitate cross validation and comparison across different studies.

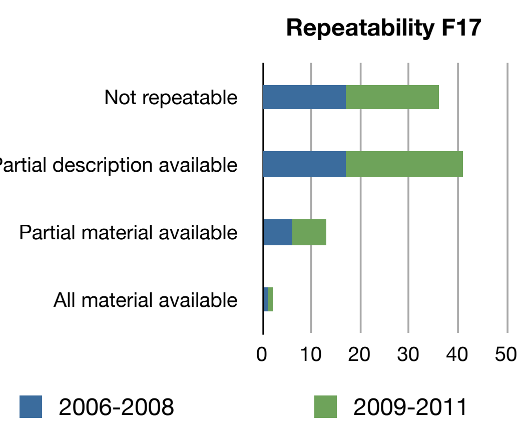


Figure 8. Repeatability (number of studies).

*RQ4: What are limitations of the existing approaches?*

Research limitations are derived from the data items: findings (F18), and limitations (F19). We identified the following recurring themes in the reported findings:

- *Technology:* existing technology can be exploited to support self-adaptation, e.g., aspects, reflection, standard interfaces (15 studies)
- *Complexity:* realizing self-adaptation is complex, e.g., implementing control loops, goal models, debugging, etc. (5 studies)
- *Model vs. Implementation:* building a solution may reveal a mismatch between the architectural models and the actual implementation (5 studies)
- *Invasiveness:* the managed system needs to be build/prepared to support self-adaptation (3 studies)

Themes identified in the reported limitations are:

- *Domain-specific solutions:* solutions that are only applicable to a given domain/application (13 studies)
- *Simplifying Assumptions:* simplifications of the system or software on which the approach relies on (11 studies)
- *Performance:* the overhead of the adaptations, inefficiency of the decision making, etc. (7 studies)
- *Manual effort:* the amount of manual effort and cost required for using the approach (4 studies)
- *Evaluation:* the lack of sufficient evidence that the approach really works (3 studies)
- *Accuracy:* the lack of adequate accuracy of the approach in achieving the objectives (2 studies)
- *Guarantees:* the absence of proofs or guarantees about reliability, safety, etc. (2 studies)

These themes confirm several limitations we derived from earlier analysis, including simplifying assumptions and limited assessment. It is important to note that the majority of the studies (66 %) report no limitations.

*RQ5: What are interesting areas for future research?*

Areas for future research are derived from data item challenges (F20). We identified the following themes in the collected data:

- *Improve experiments/evaluation:* proper evaluation and experimentation of a research was difficult and could be improved (52 studies)
- *Enhance technique/theory:* the authors identify a challenge or the need for enhancing the foundations of the approach/research (52 studies)
- *Develop tool support:* development of tool support is necessary for the approach to be practical for use (25 studies)
- *Formalize and refine concepts:* there is a need for better describing or rigorously specifying concepts, techniques, languages, etc. (23 studies)
- *Revisit assumptions:* some of the assumptions were oversimplifying and need to be revisited in the future (19 studies)

- *Alternative techniques/implementation:* possible alternative approaches are identified to solving the problem (10 studies)
- *Standardize:* take a concept and make it into a standard or incorporate standards into a given solution (2 studies)

Reported challenges confirm clearly the need to improve foundations and assessment. Some complementary opportunities for future research based on the collected data that are not reported by the studies as challenges are the study of requirements for self-adaptive systems and multiple control loops, and the application of self-adaptation to deal with other quality attributes such as security, usability and accuracy.

## IV. Discussion of the Results

We start with discussing study limitations. Then we compare our study results with findings presented in a number of related papers. We conclude with some concrete recommendations to move the field forward.

### A. Limitations of the Study

One threat to internal validity with the study is the involvement of students as reviewers, which may compromise the relevance of the results. We have anticipated this issue by paring two groups of students with two established researchers in the field to evaluate each study. During the interactive discussion sessions, we noticed that the students progressively improved their evaluation skills. In the second half of the review process, a majority of the students showed advanced competence in performing reviews.

Another threat to internal validity is the existence of bias of the reviewers. To reduce bias, we have opted to let 6 reviewers perform the initial reviews of each paper independently. Then the results were compared and discussed in case of conflicts. Bias in the design of the study and the analysis of results was kept low by involving four researchers, of which two effectively performed the reviews, the other two researchers provided feedback and supported analysis.

As we have only looked at studies of SEAMS, there is a threat to external validity with respect to the generalization of the conclusions of the study. In particular, as SEAMS was a workshop until 2010, this may explain the lack of thorough evaluation of many studies. In this respect, we do not make any claims concerning the results for the whole field. Nevertheless, as we studied a significant part of the literature on self-adaptive systems, the results are definitively valid, though not conclusive. Therefore, a complete systematic review [8] is needed, an effort we plan to start in the near future.

### B. Related Work

We limit the discussion to a number of recognized papers in the field that discuss related aspects, and an interesting survey paper that was recently presented at SEAMS.

Autonomic computing, introduced by IBM [2], is put forward as a way of reducing the total cost of ownership of complex IT systems by means of self-management. Four basic concepts are distinguished to realize self-management: self-configuration (related to runtime maintenance and flexibility), self-optimization (efficiency/performance), self-healing (reliability, availability), and self-protection (security). The results of our study show that the SEAMS community has paid little attention to reducing the costs of IT systems. Furthermore, the focus has be on the first three concepts; self-protection has received very little attention.

In line with [2], [12] argues that the primary driver for self-management is the increasing complexity of IT. The authors list the key underlying quality attributes that relate to self-management: reliability, efficiency, maintainability, usability, functionality, and portability. However, no further insight is provided on rational for these attributes nor the extent to which they are effectively studied or applied. [13] lists 10 evaluation criteria for self-adaptive systems, which include quality of service, cost, robustness, autonomy, reaction time, and stability. However, the authors make no statements about the relevance of their proposed criteria, except that the research is mainly looking at using adaptation to improve performance. Our study provides a more fine grained view on the relevance of the main concerns of self-adaptation considered by the SEAMS community.

Dobson et al. [4] point to the increase in complexity of developing autonomic/self-adaptive system, but also *the enormous labor, complexity and costs savings in the longer term*. These findings are confirmed in our study, although only by a limited number of studies. The authors also state that *decentralization is the sine qua non of autonomic systems* and argue for more research on decentralized control in self-adaptive systems. Although, limited attention has been given to decentralized of self-adaptation so far in the SEAMS community, the reported challenges give confidence that more research is on its way. This trend was also confirmed at the 2nd Dagstuhl seminar on engineering self-adaptive systems (http://www.self-adaptive.org/).

[14] argues that there is a mismatch between the expected impact of self-adaptation research (incl. by industry) and the perceived impact of such research in premier software engineering publications venues. One of the main causes is the used methods to evaluate self-adaptation research. The author provides recommendations to improve the impact of self-adaptation research by providing evaluation techniques that allow comparison between self-adaptive software and conventional software techniques. Our study offers complementary recommendations to move the field forward.

### C. Recommendations

Based on the insights derived from our study, we offer the following recommendations to move our field forward:

- To be able to evaluate better, we need the community to make their methods, tools, and data available, and even better use a couple of studies as examples, Znn.com[6] is one excellent candidate,
- To address the limitation and assumption issues, conferences/workshops in this field should require a limitation and assumption section for every paper, very similar to how the empirical software engineering papers are expected to have by default a threats to validity section,
- To address the weak treatment of tradeoffs implied by self-adaptation, a discussion of tradeoffs should be an explicit aspect of reviews, and finally
- To stimulate academy-industry collaborations, such efforts could be given special recognition at conferences, e.g., by awarding the best studies, similar as the SPL community (splc.net/fame.html). Such recognition could feed discussions about what constitutes excellence and success in self-adaptation.

## V. Conclusions

The objective of this literature study was to summarize existing research on engineering self-adaptive software systems and shed light on the claimed benefits and provided evidence of self-adaptation. The study shows that researchers clearly explain problem and contributions of their studies, but the description of the research design and limitations can be improved. The main focus of the research is on architecture and models, with particular attention on the activities of self-adaptation and runtime models. The dominant application domain for applying and evaluating self-adaption research is service-based systems, followed by robotics. The primary claims of self-adaptation are improved flexibility, reliability, and performance of the system. The main reported tradeoff implied by self-adaptation is performance overhead, but tradeoffs have not received sufficient attention. Evidence is often obtained from applying the research to simple examples. Only a few systematic empirical studies have been undertaken, and no industrial evidence is reported. Some areas that deserve further research are requirements, verification, testing, multiple control loops, and the application of self-adaptation for other quality attributes, such as usability and security. From our study, we derived the following recommendations to move the field forward: researchers should make their assessment methods, tools and data publicly available; conferences/workshops should require an explicit section on limitations in engineering papers; discussion of tradeoffs should be an explicit aspect of reviews; and successful academy-industry collaborations could be formally recognized by the community.

Over the past 20 years, several researchers in the wider software engineering community have repeated the need for systematic evaluation [15]. This systematic study confirms that this observation also applies to the SEAMS community. We hope that the results of this study and the proposed recommendations can be useful for our community as a step towards improvement of the research in our field.

[6] http://rainbow.self-adapt.org/benchmark

## References


[1] P. Oreizy *et al.*, "An architecture-based approach to self-adaptive software," *IEEE Intelligent Systems*, 14(3), 1999.

[2] J. Kephart and D. Chess, "The vision of autonomic computing," *Computer*, vol. 36, no. 1, 2003.

[3] J. Kramer and J. Magee, "Self-Managed Systems: an Architectural Challenge," *Future of Software Engineering, ICSE*, 2007.

[4] S. Dobson *et al.*, "A survey of autonomic communications," *ACM TAAS*, vol. 1, 2006.

[5] M. C. Huebscher and J. A. McCann, "A survey of autonomic computing: degrees, models, and applications," *ACM Computer Surveys*, vol. 40, 2008.

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

[7] B. H. Cheng *et al.*, "Software engineering for self-adaptive systems: A research roadmap," in *Software Engineering for Self-Adaptive Systems*. LNCS vol. 5525, Springer, 2009.

[8] B. Kitchenham and S. Charters, "Guidelines for performing systematic literature reviews in software engineering," Keele and Durham University, Tech. Rep. EBSE 2007-001, 2007.

[9] P. Brereton *et al.*, "Lessons from applying the systematic literature review process within the software engineering domain," *Journal on Systems and Software*, vol. 80, pp. 571–583, 2007.

[10] V. R. Basili, G. Caldiera, and H. D. Rombach, "The goal question metric approach," in *Encyclopedia of Software Engineering*. Wiley, 1994.

[11] T. Dybå and T. Dingsøyr, "Empirical studies of agile software development: A systematic review," *Inf. Software Technology*, vol. 50, pp. 833–859, 2008.

[12] M. Salehie and L. Tahvildari, "Autonomic computing: emerging trends and open problems," in *Design and evolution of autonomic application software (DEAS'05)*, 2005.

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

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

[15] C. Zannier *et al.*, "On the success of empirical studies in the international conference on software engineering," in *ICSE*, 2006.