8-2022

Self-adaptive systems: A systematic literature review across categories and domains

Terence WONG

Markus WAGNER

Christoph TREUDE
Singapore Management University, ctreude@smu.edu.sg

Citation

WONG, Terence; WAGNER, Markus; and TREUDE, Christoph. Self-adaptive systems: A systematic literature review across categories and domains. (2022). *Information and Software Technology*. 148, 1-25.
Available at: https://ink.library.smu.edu.sg/sis_research/8767

Self-Adaptive Systems: A Systematic Literature Review Across Categories and Domains

Terence Wong[a], Markus Wagner[a], Christoph Treude[b]

[a]*School of Computer Science, The University of Adelaide, Australia*
[b]*School of Computing and Information Systems, The University of Melbourne, Australia*

Abstract

Context: Championed by IBM's vision of autonomic computing paper in 2003, the autonomic computing research field has seen increased research activity over the last 20 years. Several conferences (SEAMS, SASO, ICAC) and workshops (SISSY) have been established and have contributed to the autonomic computing knowledge base in search of a new kind of system – a self-adaptive system (SAS). These systems are characterized by being context-aware and can act on that awareness. The actions carried out could be on the system or on the context (or environment). The underlying goal of a SAS is the sustained achievement of its goals despite changes in its environment.

Objective: Despite a number of literature reviews on specific aspects of SASs ranging from their requirements to quality attributes, we lack a systematic understanding of the current state of the art.

Method: This paper contributes a systematic literature review into self-adaptive systems using the dblp computer science bibliography as a database. We filtered the records systematically in successive steps to arrive at 293 relevant papers. Each paper was critically analyzed and categorized into an attribute matrix. This matrix consisted of five categories, with each category having multiple attributes. The attributes of each paper, along with the summary of its contents formed the basis of the literature review that spanned 30 years (1990-2020).

Results: We characterize the maturation process of the research area from theoretical papers over practical implementations to more holistic and generic

Email address: `terence.wong@adelaide.edu.au`, `markus.wagner@adelaide.edu.au`, `christoph.treude@unimelb.edu.au` ()

approaches, frameworks, and exemplars, applied to areas such as networking, web services, and robotics, with much of the recent work focusing on IoT and IaaS.

Conclusion: While there is an ebb and flow of application domains, domains like bio-inspired approaches, security, and cyber physical systems showed promise to grow heading into the 2020s.

Keywords: Self-Adaptive Systems, Literature Review

1. Introduction

Information systems and the devices that contain them are growing more complex and more pervasive in society. This is due to the increased quantity of and increased demands on these systems. In the past, these systems operated standalone in isolation from other devices with a narrowly prescribed function (e.g., early mobile phones or PCs). In the current age, these systems have become more distributed (e.g., cloud servers, sensor networks) and more complex (e.g., current mobile phones or PCs). Users expect these systems to be always connected and highly integrated with minimal downtime.

As the nature of information systems has rapidly evolved, the original in-isolation requirements of these systems are outdated. It is no longer beneficial to specify a system's behavior at design time because there are overwhelmingly more potential system states than can be designed for. This is due to the increased functionality and connectedness of modern information systems. Hence, it is desirable for a system to be able to adapt its behavior at run time to changes in its context (or environment) to ensure the continual achievement of its goals. Such a system is called a self-adaptive system (SAS) [1, 2].

In the grand challenge presented by IBM [2], SASs were to provide self-management properties such as self-configuration, self-optimization, self-healing, and self-protection. This challenge led to the establishment of the International Conference on Autonomic Computing (ICAC) as well as establishing foundational theory on SASs [3]. Self-adaptive approaches range from static, reactive, parametric solutions to dynamic, proactive, structural solutions. The former approaches are based in predetermined plans and configurations while the latter approaches commonly leverage the power of AI/ML [4].

While systematic reviews have been conducted to characterize the state of the art of specific aspects of self-adaptive systems such as requirements [5, 6], claims and evidence [7], quality attributes [8], and machine learning in the context of SAS [9], there is a lack of systematic work spanning and providing an overview of all aspects of self-adaptive systems. In this work, we contribute a systematic literature review into self-adaptive systems which categorizes papers into five categories (Technological, Methodology, Perspective, Analytical, Empirical) and summarizes trends and developments across time, categories, and related attributes.

Our review characterizes the development of a research area over 30 years, from theoretical and model based papers in the 1990s and practical implementations and frameworks in the early 2000s to the ramping up period featuring more holistic and generic approaches, which were forthwith extended to frameworks and exemplars. While web services were prevalent in the field as application domain for much of the foundational years, Internet of Things (IoT) and Infrastructure as a Service (IaaS) have dominated self-adaptive systems in the last five years.

To the best of our knowledge, this is the first systematic literature review on self-adaptive systems produced after the ramping up of the research field that is not limited to a specific aspect of the field, such as the aforementioned reviews. Our main contribution is the cataloging of research on self-adaptive systems and the organization of this catalog according to paper categories, application domains, and additional attributes specific to each paper category. This catalog serves multiple purposes: (i) summarizing the state of the art for practitioners by providing a multi-faceted overview of previous and current work, (ii) identifying current trends and gaps for researchers by organizing content over time and exposing common and uncommon attribute combinations, and (iii) highlighting areas with high potential impact to guide educators in assessing which skills will be particularly relevant for future decision makers.

We find that the current state of the art in self-adaptive systems is focused on developing methodologies and technology in the area of cloud-based services, such as IoT and IaaS. Even though research on self-adaptive systems tends to be diverse, empirical and analytical research is currently playing a smaller role, as are other application domains. Perhaps encouragingly, many approaches are evaluated using real-world case studies, with less reliance on simulations. The importance of self-adaptive systems is rapidly growing in areas such as bio-inspired approaches, security, and cyber physical systems.

Going forward, we expect to see a shift towards empirical studies as the research field continues to mature, with industrial case studies in a diverse range of application domains.

The remainder of this paper is structured as follows: Section 2 presents background and highlights findings from related reviews. Section 3 presents our methodology before we provide a chronological overview of research on self-adaptive systems in Section 4. Section 5 discusses the limitations and threats to validity of this work, before Section 6 concludes this paper.

2. Background and Related Reviews

In this section, we define key terminology in the area of self-adaptive systems in the context of existing literature reviews which often focus on specific aspects of self-adaptive systems.

Autonomic computing is a self-managing computing model named after, and patterned on, the human body's autonomic nervous system. It deals with the design and the construction of computing systems that possess inherent self-managing capabilities [10]. The term gained popularity in the early 2000s as a result of IBM's autonomic computing initiative. Seminal articles by Ganek, Kephart and others [2, 11, 12] describe the fundamental characteristics of autonomic systems, a framework for how systems will evolve to become more self-managing, and the key role for open industry standards needed to support autonomic behavior in heterogeneous system environments. A first consideration of research challenges in the field of autonomic computing was published by Kephart in 2005 [13], with a focus on autonomic element challenges, autonomic system challenges, and human-computer challenges. A survey published in 2008 found autonomicity to be not well defined, leading to different systems adhering to different degrees of autonomicity [14].

A **self-adaptive system** is a closed-loop system with a feedback loop aiming to adjust itself to changes during its operation [15]. In one of the few review articles that span the entire field of self-adaptive software, Salehie and Tahvildari present a landscape of research in self-adaptive software by highlighting relevant disciplines and prominent research projects [15]. Since the publication of their review in 2009, other review articles have focused on specific aspects of self-adaptive systems, ranging from engineering approaches for self-adaptive systems [16] and requirements modeling and analysis for self-adaptive systems [5, 6] to machine learning in the context of self-adaptive systems [9] and quality attributes that are frequently addressed [8].

A **self-healing system** is a system that is capable of performing a reconfiguration step in order to recover from a permanent fault [17]. In an early review of work on self-healing systems, Ghosh et al. [18] surveyed research in this field and proposed a strategy of synthesis and classification. This was followed by surveys by Psaier and Dustdar [19] and Schneider et al. [20]. A special case of self-healing systems are **self-protecting systems**, defined as autonomic systems capable of detecting and mitigating security threats at runtime. Yuan and Malek provided a review of work in this area [21].

Finally, Weyns et al. [7] provided a review on an aspect that is orthogonal to the types of self-adaptive systems, instead focusing on researchers' claims and supporting evidence in this field. They recommend researchers to make their assessment methods, tools and data publicly available and to improve discussion of limitations [7].

Contrary to these reviews and more in line with the 2009 review by Salehie and Tahvildari [15], we do not limit ourselves to a particular aspect of self-adaptive systems or a particular aspect of published research in the area, instead aiming to provide a high-level overview across categories and application domains.

3. Methodology

This section outlines the methodology followed in our systematic review, detailing the steps recommended by Kitchenham et al. [22].

3.1. Research Questions

We used the following research questions to drive our data collection and analysis:

RQ1 What is the current state of the art in self-adaptive systems?

RQ2 How has the state of the art evolved over time?

RQ3 Which are the application domains of self-adaptive systems over time?

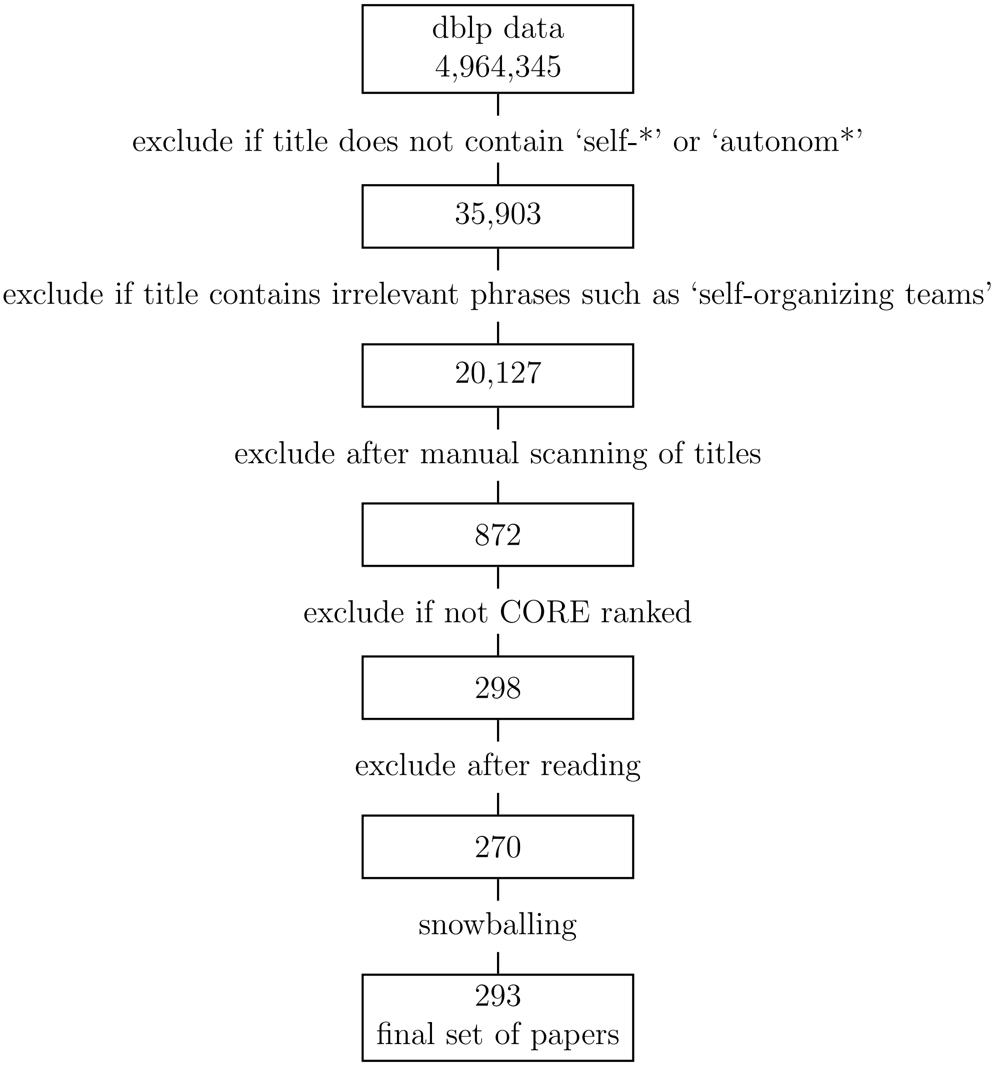
dblp data
4,964,345
exclude if title does not contain ‘self-*’ or ‘autonom*’
35,903
exclude if title contains irrelevant phrases such as ‘self-organizing teams’
20,127
exclude after manual scanning of titles
872
exclude if not CORE ranked
298
exclude after reading
270
snowballing
293
final set of papers

Figure 1: Methodology Overview

3.2. Study Protocol

To select literature to include in our systematic review on self-adaptive systems, we used the dblp computer science bibliography (dblp) database as a starting point, effectively eliminating non-computer science literature from the process. Dblp contains historical snapshots of the database enabling reproducibility of results. All results in this review were produced using the snapshot file dblp-2020-03-02.xml.[1]

We chose dblp as the starting point of our analysis since it provides a consistent format for all articles indexed in the database which allowed us to employ consistent search criteria (such as lower-case vs. upper-case) across content published by different publishers. Dblp's focus on "major computer science publications"[2] gives us a narrower focus than a more general search engine and a finite number of search results that does not vary on a daily basis, as it would for example on Google Scholar. We acknowledge that using a different database as a starting point would have resulted in a different set of papers, see Section 5 for a discussion of the corresponding trade-offs.

The methodology is broken down into five main stages which combine automated and manual processes to keep the required amount of manual work (i.e., reading abstracts and papers) manageable while ensuring the quality of the selection process: Pre-filtering, word frequency filtering, venue selection, abstract reading, and snowballing to reduce a first subset of 35,903 publications to 293 publications. Figure 1 shows a high-level overview of the process.

3.2.1. Pre-Filtering

The dblp data containing 4,964,345 papers was filtered to include only journal articles and papers published in conference proceedings, with a page count of over five to focus our analysis on substantial and fully evaluated research contributions which appeared in peer-reviewed venues. Capitalization was removed to assist with the filtering process. This established the base dataset. The base dataset was searched with grep for two terms: self-* and autonom* to produce two separate datasets. These two keywords were deemed as broadly relevant to the study area. The total number of publications matching these keywords was 35,903.

[1] `https://dblp.org/xml/release/dblp-2020-03-02.xml.gz`

[2] `https://dblp.org/faq/What+is+dblp.html`

Table 1: Words after self-

Word	Frequency
organizing	2,981
adaptive	1,682
stabilizing	632
organization	564
organized	515

Table 2: Words after autonom*

Word	Frequency
vehicles	576
mobile	554
driving	407
systems	399
agents	399

3.2.2. Word Frequency Filtering

The initial search for publication titles including 'self-*' or 'autonom*' produced too many papers to manually analyze. However, we noticed that many papers which matched the search could be easily excluded since their topics were clearly out of scope, e.g., self-driving cars. To formalize this process, we further filtered papers based on additional words in the paper titles. We first determined the distribution of words after the keywords self-* and autonom*. Tables 1 and 2 show the distribution of the five most frequent words for self-* and for autonom*.

Because we observed that self-organizing and self-adaptive frequently co-occurred with terms irrelevant to self-adaptive systems (such as self-organizing teams), we conducted a second word frequency analysis to identify common words appearing after self-organizing and self-adaptive. The top results are shown in Tables 3 and 4.

To ensure the validity of deciding which keywords we deemed to be out of scope, two of the authors manually and independently analyzed all words occurring after 'self-', 'autonom*', 'self-organizing', and 'self-adaptive' at least 20 times to indicate those that were out of scope (such as self-organizing teams) for exclusion. We calculated inter-rater reliability using Cohen's κ,

Table 3: Words after self-organizing

Word	Frequency
maps	706
map	582
neural	165
feature	109
networks	79

Table 4: Words after self-adaptive

Word	Frequency
systems	159
differential	77
software	76
and	31
evolutionary	27

see Table 5. The Cohen's κ value was greater than 0.7 in all cases which was deemed as acceptable. Disagreements were resolved after confirming any ambiguities and biases. This step resulted in a list of words to exclude from the final data set. After omitting papers with the excluded phrases in the title, the number of publications reduced to 20,137.

The titles of the 20,137 publications were manually scanned for relevance, resulting in 872 articles. The purpose of this step was to eliminate papers that matched our keyword filters but were not related to self-adaptive systems. For example, papers on self-driving cars matched our keyword filters but are not related to self-adaptive systems. We again assessed the validity of this step using Cohen's κ by having two authors identify relevant papers in a randomly selected subset of 50 papers. The value calculated was 0.73 which was deemed acceptable.

Table 5: Inter-rater agreement

Dataset	Cohen's κ
Autonom*	0.828
Self-*	0.748
Self-organizing *	1.000
Self-adaptive *	1.000
50 randomly sampled titles	0.730

3.2.3. Venue Selection

As a quality gate, we only considered papers published in A*/A conferences or journals, as determined by the CORE ranking.[3] 46% of all journals and 55% of all conferences listed by CORE are ranked as A*/A. Papers published in B-ranked venues were included if they came from a journal or conference that was relevant to the study area (e.g., SEAMS[4], SASO, and TAAS). This step reduced the number of candidate papers to 298. We discuss the trade-offs associated with this step in Section 5.

3.2.4. Abstract Reading and Attribute Matrix

The abstracts of the 298 candidate papers were then read to categorize each paper into one of five categories based off the paper categorization introduced by the ICSE 2014 conference:[5]

- Analytical: A paper in which the main contribution relies on new algorithms or mathematical theory.
- Empirical: A paper in which the main contribution is the empirical study of a software engineering technology or phenomenon.
- Technological: A paper in which the main contribution is of a technical nature.

[3] The CORE rankings `http://portal.core.edu.au/conf-ranks/` and `http://portal.core.edu.au/jnl-ranks/` are maintained by the Computing Research and Education Association of Australasia and are used world-wide.

[4] The CORE rank of SEAMS has changed to A after we conducted this study

[5] The categories published in the ICSE 2014 call for papers at `https://2014.icse-conferences.org/research` provide one of the most comprehensive categorization schemes for software-related papers.

- Methodological: A paper in which the main contribution is a coherent system of broad principles and practices to interpret or solve a problem.
- Perspectives: A paper in which the main contribution is a novel perspective on the field as a whole, or part thereof.

Once each paper was categorized, an attribute matrix for each category was developed. Within each category, the abstract and a pass of the full paper was read to develop an understanding of the common types of attributes for a category. This was done in an iterative approach. Once the attribute matrix was finalized, all the papers from each category were categorized using the matrix. This step resulted in 28 papers being discarded as irrelevant. Note that each category has its own attribute matrix since not all attributes apply to all categories.

We further identified the application domain of each paper, if applicable. If the research has an application focus, the application domain is the business/application sector of the work (e.g. IoT, IaaS, Automotive), otherwise it is the engineering domain (e.g., Web Services, Robotics), or the more general domain (e.g., Bio-inspired, Software Engineering, Security). The application domains are shown in the attribute matrices at the end of the paper and the top 10 most common application domains are summarized in Table 7. Please refer to Tables 19 through 26 for the complete list of papers that we assigned to each application domain.

3.2.5. Snowballing

To capture additional papers, snowballing was used on the papers included in the attribute matrix. Where possible, each paper was added to a Scopus list where the references were extracted automatically. This step captured 90% of the papers in the matrix. Any paper that was referenced more than five times was eligible to be considered. These papers were then filtered through the same criteria as the original papers from the dblp step (i.e., abstracts, CORE rankings). From this step, 23 new papers were added. These papers were categorized into the five categories following the same approach as before. The final counts for each category are shown in Table 6 and are considered in the ratios reported in the previous section. Please refer to Tables 19 through 26 for the complete list of papers that we assigned to each category.

Table 6: Papers per category

Category	Count
Technological	105
Methodology	79
Perspective	51
Analytical	35
Empirical	23
Total	293

Table 7: Papers per application domain

Application Domain	Count
Web Services	48
IoT	36
Review	25
Robotics	23
Networking	19
IaaS	17
Intelligence Surveillance Reconnaissance	13
Software Engineering	12
Automotive	10
Mobile Systems	10
Service-Oriented Systems	10

4. Self-adaptive systems over the years

In the following, we provide a chronological overview of research on self-adaptive systems based on our systematic literature review. The complete matrices with the characterizations of all papers are shown at the end of the paper and summarized here:

- Technological: 81% (◕) of papers in this category follow a closed circle approach as opposed to human in the loop, and implementations cover tools (24% ◔), models (23% ◔), frameworks (22% ◔), languages, architectures, and algorithms. 40% (◑) of the papers rely on simulations. The vast majority of contributions are novel (87% ◕), compared to a

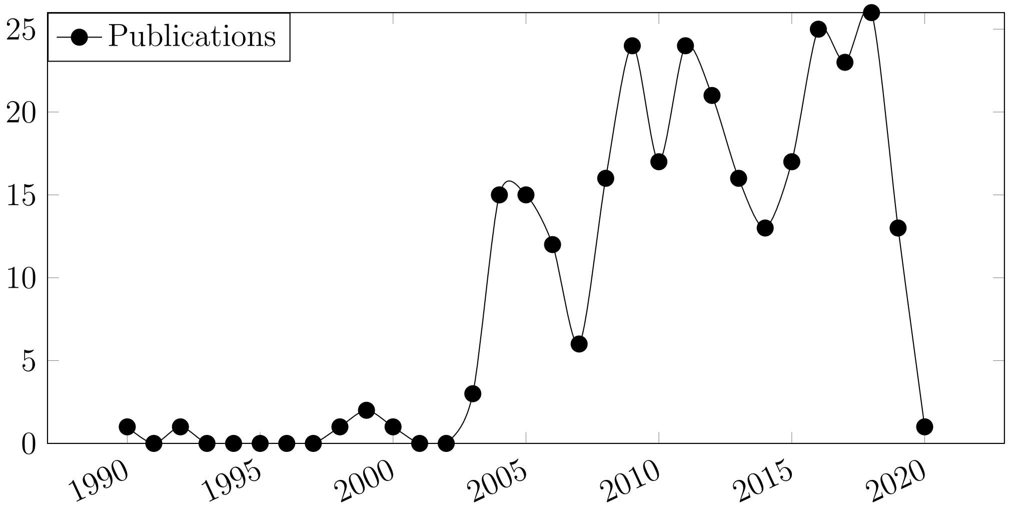

Figure 2: Number of publications over time

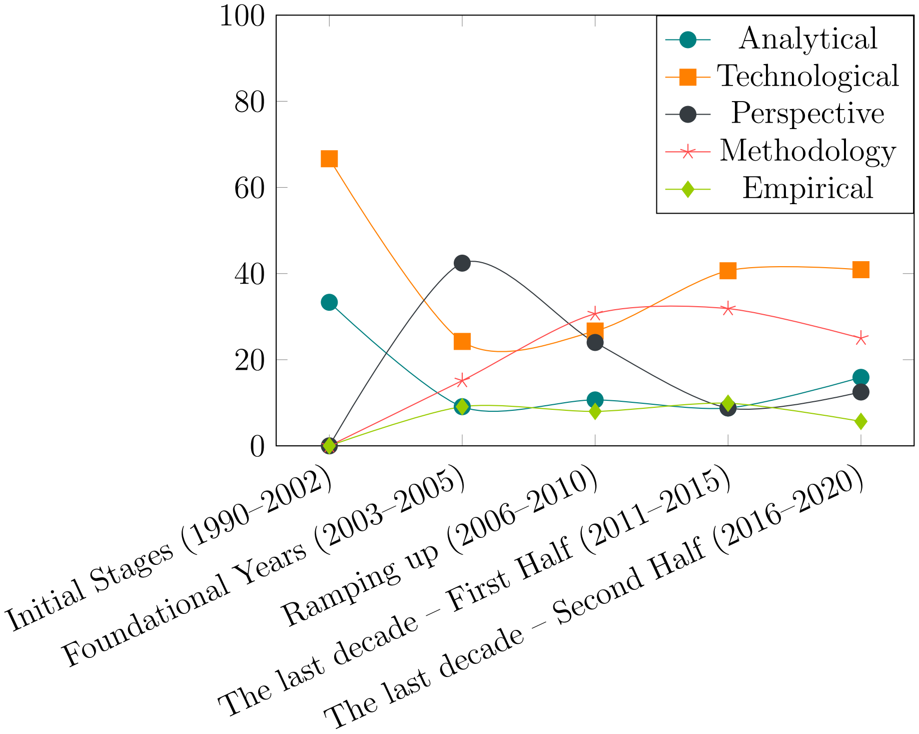

Figure 3: Categories over time

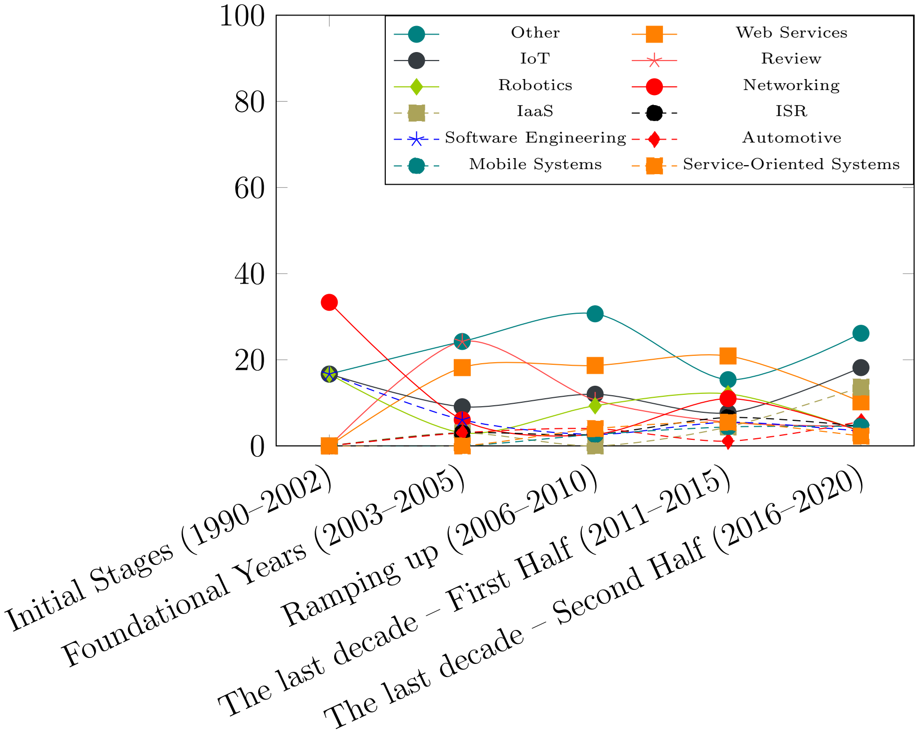

Figure 4: Applications domains over time

Table 8: Summary of attribute matrices. Note that totals can differ from 1.0 when attribute values are unknown or when articles attain multiple values.

Type	Algorithm	Architecture	Framework	Mathemat.	Language	Definition
Analytical	0.46	0.03	0.14	0.26	0.06	0.09
Type	General arch.	New Framew.	Analysis Techn.	New Pattern	Formal Crit.	New Appr.
Methodology	0.04	0.15	0.08	0.01	0.03	0.70
Type	Survey	Review	Evaluation	Reflection	Roadmap	Comparison
Perspective	0.20	0.31	0.04	0.35	0.02	0.08
Type	Human	Closed circle				
Technological	0.18	0.81				
Formalization	Yes	No				
Analytical	0.94	0.03				
Implementation	Tool	Model	Framework	Language	Architecture	Algorithm
Technological	0.24	0.23	0.22	0.14	0.10	0.07
Goals	Goals	Utility				
Technological	0.55	0.25				
Content	Taxonomy	Challenges	Future Work	Requirements		
Perspective	0.14	0.33	0.71	0.16		
Testing	Design-time	Run-time				
Empirical	0.00	0.74				
Strategy	Monitoring	Adaptation				
Empirical	0.83	0.87				
Adaptation	Parameter	Component				
Empirical	0.87	0.09				
Contribution	Extension	Novel				
Analytical	0.31	0.69				
Technological	0.12	0.87				
Methodology	0.09	0.90				
Application	Case study	Simulated				
Analytical	0.57	0.40				
Technological	0.60	0.40				
Methodology	0.78	0.20				
Empirical	0.61	0.35				
Evaluation	Preliminary	Case Study	Industrial	Comparison	Human	Quant.
Analytical	0.09	0.29	0.00	0.17	0.00	0.37
Technological	0.01	0.35	0.00	0.15	0.00	0.44
Perspective	0.00	0.29	0.00	0.31	0.00	0.16
Methodology	0.00	0.38	0.00	0.16	0.00	0.41
Empirical	0.00	0.39	0.04	0.13	0.00	0.57

few extensions of previous work. The methods for evaluation range from quantitative approaches (44%) and case studies (35%) to comparisons (15%).

- Methodology: The majority of papers in this category (70%) introduce new approaches for self-adaptive systems, followed by frameworks (15%), analysis techniques (8%), and architectures (4%). The ratio of simulations (20%) is lower than in the Technological category, and the vast amount of papers provide new contributions (90%) here as well. Evaluation follows a similar pattern to the Technological category, with quantitative evaluations (41%) being the most common, followed by case studies (38%) and comparisons (16%).

- Perspective: Many of the papers in this category can be classified as reflections (35%), reviews (31%), and surveys (20%), with papers outlining future work (71%), challenges (33%), and requirements (16%), or providing a taxonomy (14%). Evaluation methods, if applicable, in this category are mostly focused on comparisons (31%) and case studies (29%).

- Analytical: Many papers in this category focus on algorithms (46%), followed by mathematical contributions (26%) and frameworks (14%). Like other categories, the number of case studies vs. simulations follows roughly a 60/40 split. A substantial number of papers are extensions of previous work (31%), and almost all (94%) papers provide a formalization. In terms of evaluation methods, the largest group are quantitative (37%), followed by case studies (29%) and comparisons (17%).

- Empirical: Most of the empirical papers focus on run-time (74%) instead of design-time, with an explicit monitoring (83%) and/or adaptation (87%) strategy. Studies again follow the 60/40 split between case studies and simulations. The adaptation techniques focus on parameters (87%) rather than components (9%), and evaluation is mostly quantitative (57%) or done through case studies (39%).

Our analysis revealed the following five time periods, mostly defined by the number of papers published on the topic and their primary research focus:

- 1990-2002 - Initial Stages, Section 4.1

- 2003-2005 - Foundational Years, Section 4.2
- 2006-2010 - Ramping up, Section 4.3
- 2011-2015 - The last decade - First Half, Section 4.4
- 2016-2020 - The last decade - Second Half, Section 4.5

As an orthogonal perspective, the tables at the end of this paper group all 293 papers by category and list their attributes. Table 8 provides a high-level summary by showing the distribution of attribute values in each category.

We briefly describe the attributes and their values in the following, before characterizing the field in each time period. Note that attributes and their values emerged from our analysis of the papers:

- Type: The type of contribution of a paper depends on the paper category. Many of the analytical papers introduce a new algorithm or provide a formal mathematical contribution. Other contribution types of analytical papers include frameworks, definitions, languages, and architectures for self-adaptive systems. The majority of methodology papers introduce a new approach for self-adaptive systems, with other contributions including new frameworks and architectures, new analysis techniques, new patterns, and formal criteria, e.g., for diagnosing components in self-adaptive systems. Perspective papers typically contribute reflections, reviews, or surveys, with a smaller number of papers focusing on evaluating specific frameworks, presenting roadmaps, or comparing approaches to a particular problem. Lastly, the types of contributions of technological papers can be divided into human-in-the-loop and closed-circle approaches, depending on the level of human intervention in the self-adaptive system.
- Formalization: The vast majority of analytical papers provides a formalization of their contribution, e.g., using mathematical definitions and formulas.
- Implementation: Technological papers provide an implementation of something, ranging from tools, models, and frameworks, to languages, architectures, and algorithms.

- Goals: Technological papers can further be distinguished based on whether they focus on the goals of a self-adaptive system and/or associated utility functions. A self-adaptive system should respect the utility while trying to achieve its goal.

- Content: The content of the majority of perspective papers is a thorough discussion of future work, with others listing challenges or requirements, or providing a taxonomy.

- Testing: While testing can generally be divided into design-time testing and run-time testing depending on whether a system is running during testing, our review only revealed empirical papers focused on run-time testing.

- Strategy: Most empirical papers employ an adaptation strategy which defines possible actions and their implementation as well as a monitoring strategy which defines how to extract information from the system.

- Adaptation: The majority of empirical papers employ parameter adaptation, i.e., fine tuning of applications through the modification of application variables and deployment parameters, rather than component adaptation, i.e., the replacement, addition, or removal of components.

- Contribution: Most papers exclusively present original work, while a minority are extensions of other published work, e.g., journal extensions of conference papers.

- Application: The majority of approaches introduced across all paper categories (except perspective) are evaluated using real-world case studies, with less reliance on simulations.

- Evaluation: Across all paper categories, the methods for evaluation of research on self-adaptive systems are dominated by quantitative approaches, case studies, and comparisons. We use the attribute value ‘quantitative’ in cases of statistical tests based on simulations, often in absence of a baseline. We use the attribute value ‘case study’ to refer to in-depth studies of one or a small number of systems. We use the attribute value ‘comparison’ to indicate work that was primarily evaluated by comparing to a baseline, e.g., from previous work. Other

Table 9: Categories during initial stages (1990–2002)

Category	Count
Technological	4
Analytical	2
Total	6

evaluation methods include real-world ('industrial') and user studies ('human'), or those explicitly labeled by the authors as 'preliminary'.

In addition, Figures 2, 3, and 4 visualize the trends of number of papers over time, categories over time, and application domains over time.

4.1. 1990-2002 - Initial Stages

According to our literature review, the earliest reference to self-adaptive systems was in 1990. The period of 1990-2003 was the phase before 'The Vision of Autonomic Computing' [2], was published. This was where the field was in its beginning stages.

References in this phase contained theoretical and model based papers such as a model for dynamic change management [23], self-stabilizing real-time rule based systems [24], and convergence for self-stabilizing systems [25]. These papers were focused on presenting a theoretical model or proving a proposed theorem to advance self-adaptive systems theory.

By 1998, practical implementations were seen such as architecture based run-time software evolution [1], self-supervised category detection [26], and self-adaptive control software [27]. The papers were among the first to discuss terms like evolution, self-supervision, control theory, and run-time design in the context of self-adaptive systems. Thirty years later, these terms are commonplace and part of the general understanding of the field.

Tables 9 and 10 characterize this time period in terms of the prevalent paper categories and application domains.

4.2. 2003-2005 - Foundational Years

In the subsequent years after 2002, two seminal works were produced that formed the foundations of self-adaptive systems. In 2003, The Vision of Autonomic Computing [2] was published which kick-started the field of Autonomic Computing. The paper presented a grand challenge of self-configuration,

Table 10: Application domains during initial stages (1990–2002)

Application Domain	Count
Networking	2
Software Engineering	1
IoT	1
Speech Recognition	1
Robotics	1

self-optimization, self-healing, and self-protection in computing systems. It foretasted the rising need for these systems in the modern age. Shortly after this, an implementation of these concepts was developed, known as the RAINBOW framework [28]. This framework was architecture based and extensible, meaning existing architectures could be imbued with self-adaptive properties. To this day, the RAINBOW framework is used as a standard, test bed, extensible tool in self-adaptive systems research.

Many papers were published during this time period that were in the perspective category (42%). As the excitement of a new field grew, many researchers sought to publish their thoughts and ideas as to how the field should and could develop. During this period, there were fewer practical demonstrations and implementations compared to future time periods as the field had to have time to grow and mature.

This application domain distribution of this period was skewed towards review papers. Of the eight review papers, four were on autonomic computing [2, 13, 11, 12] and four were on self-* properties [29, 30, 31, 32]. The key messages of these papers were that a new challenge was emerging in the field of autonomic computing and self-* computing. Due to the rise in system complexity, it was necessary to develop a new kind of system that was self-adaptive. These papers also envisioned what such realized systems might look like.

The most immediate application domain for self-adaptive systems was web services. At the time, technologies such as IaaS and IoT had not fully emerged yet. Web services were a prominent tool used in software engineering.

The improvement of the system administrator role was a focus area of research by improving collaboration and coordination, rehearsal and planning, maintaining situation awareness, and managing multitasking, interruptions,

Table 11: Categories during foundational years (2003–2005)

Category	Count
Perspective	14
Technological	8
Methodology	5
Analytical	3
Empirical	3
Total	33

and diversions [33]. Utility functions emerged as a potential metric to solve self-adaptive problems [34]. In later years, this would prove to be true. The automatic management of web services was an important test bed to develop self-adaptive algorithms and theory [35, 36, 37].

There was a major parallel between autonomic computing, self-* computing, and biological systems, and a branch of research was formed to gain inspiration from nature to bring to these systems [38, 39, 40]. As a precursor to the cloud based systems of the present day, load balancing was an area of research focus. Work forecasting [41], scheduling algorithms [42], and managing system level properties [43] were all part of the groundwork of this research area, as was the placement of replicants in an edge computing scenario [44].

IoT and IaaS papers were present during this time. Concepts like autonomous deployment [45], generic control frameworks [46], and self-healing distributed architectures [47] were explored, as was the concept of self-aware systems [48] in general. These two fields would emerge as strong research fields in later years.

Other research areas include Robotics [49], Automotive [50], Computer [51], Software Stack [52], Networking [53, 54], Software Engineering [55], and Intelligence Surveillance Reconnaissance [56].

Tables 11 and 12 characterize this time period in terms of the prevalent paper categories and application domains, confirming the large number of perspective papers.

4.3. 2006-2010 - Ramping up

The distribution of the years 2006-2010 showed a more even spread across the categories. Technological, Perspective, and Methodology papers were

Table 12: Application domains during foundational years (2003–2005)

Application Domain	Count
Review	8
Web Services	6
Load Balancing	3
Bio-inspired	3
IoT	3
Networking	2
Software Engineering	2

evenly spread. There was still a high demand for perspective type papers but enough time had passed since 2003 for technology and methodology papers to emerge.

Web services was still the highest polling application domain but in this period IoT papers surfaced as the second most frequent domain. Web services in this phase focused on the principles and ideas of autonomic and self-adaptive systems developed in the previous phase. These manifested initially in new languages [57], utility functions [57], and distributed solutions as opposed to centralized solutions [58]. There was a clear focus on run time instead of design time solutions [59, 60] as well as making systems dynamic in their configuration and adaptation [61, 62, 63]. Self-healing was an important part of web services research during this time, highlighting the use in workarounds [64] and composition cycles [65]. The RAINBOW framework was evaluated on a signature case study: Znn.com to assess the effectiveness of self-adaptation with good results [66]. In the latter half of this period, the focus of web services grew to become more holistic with concepts such as architectural self-reconfiguration and self-tuning [67, 68], behavioral adaptations [69], expansion into more self-* properties [70], and adaption logic based approaches [71]. The emphasis was on adaptation of the entire system as opposed to individual components.

IoT papers were initially focused on solvers [72, 73] with a key driver being dynamic behavior [74, 75]. Fault tolerance [74] and multi-agent models [72] were important themes to establish in IoT due to their distributed and always-on nature. A tool called RELAX was developed to handle requirements and uncertainty in a smart home self-adaptive system domain [76]. Requirements and uncertainty are important in any self-adaptive system, not just IoT systems [77]. The requirements of the system determine the goals of the adaptations. Appropriately managing uncertainty gives a system flexibility and dynamism. As with web services, a holistic approach to self-adaptation using architectural self-reconfiguration was a part of the research [78]. Adaption logic in IoT was addressed using heuristics in availability and response time [79] and in a tool called FUSION [80]. The research efforts of IoT compared with web services in this phase share a parallel. Initially, solvers and languages were created and tested, then various research concepts were explored like distributed solutions for web services or fault tolerance for IoT. Then, a shift in focus to more holistic approaches to self-adaptability were emphasized such as architectural self-reconfiguration, or the adaptation logic.

This phase was the first time robotics was seen as an application domain with seven papers. Robotics represented a high potential area for the application of self-adaptive systems. By their nature, robots are intended to automatically do the tasks of humans and so self-adaptive robots are a natural progression of the goals. Requirements and modelling [81, 82] were the early emphasis with a progression to adaptable software architectures [82, 83] and self-organized and distributed systems [84]. A reference model to compare adaptation approaches was developed [85]. This started to become necessary as the popularity of the field gave rise to multiple different adaptation techniques. Learning and planning was a component of the research of this phase [86, 87]. Using reinforcement learning, planning and architecture self-management was explored [86]. The learning research theme was in its early stages in this phase. In future years, learning would prove to be key to widely used tools like machine and deep learning.

This phase also gave rise to nine review papers. Compared to the previous phase, review papers made up a smaller proportion of the total papers in the group. The reviews were conducted on self-organization [88, 89], autonomic computing [90, 14, 91], self-healing [18], and self-adaptation [15, 92], as well as a general overview of self-* properties [93]. The common theme of this work was to highlight the current work and future needs of the field. One remark confirms the trend highlighted above of needing a more holistic approach to self-adaptation: *"but what is missing is an holistic approach focusing explicitly on providing autonomic properties"* [91]. Written in 2009, it perhaps explains the trend toward holistic solutions to self-adaptive systems in the literature.

Table 13: Categories during ramping up stage (2006–2010)

Category	Count
Methodology	23
Technological	20
Perspective	18
Analytical	8
Empirical	6
Total	75

This phase showed a variety of other application domains. Decentralized solutions were popular in automated traffic control [94], e-Commerce [95], water networks [96], service oriented systems [97], and load balancing solutions [98]. Learning was a prominent theme for automotive [99], traffic management control [100], and e-Commerce [101]. The ability for a system to adapt its adaptation logic was a strong area of focus for mobile systems[102], ISR [103], e-Commerce [104], software engineering applications [105, 106], and systems on chip [107]. Evaluation [108, 109], self-organization [110, 111, 112], self-protection [113], self-healing [114], resource allocation [115], reflection [116], as well as dynamic solutions [117] were all explored during this time. Consistent with the theme of holistic solutions, generic architectures were developed in speech recognition tools [118] as well as generic frameworks such as SASSY [119]. Other application domains not listed include UML [120], Holonic Systems [121], Video Encoding [122], task scheduling [123], and Physics [124].

Tables 13 and 14 characterize this time period in terms of the prevalent paper categories and application domains, underlining that papers in the 'Methodology' category and about Web Services were becoming increasingly dominant.

4.4. 2011-2015 - The last decade - First Half

The first half of the 2010s showed a bimodal result, with technological and methodology papers being the most frequent. The proportion of perspective papers decreased, perhaps attributed to the saturation of them in previous time periods.

Web services still dominated the domain distribution, followed by robotics and networking. IoT still had a high ranking with seven papers. This was

Table 14: Application domains during ramping up stage (2006–2010)

Application Domain	Count
Web Services	14
IoT	9
Review	8
Robotics	7
E-Commerce	3
Service Oriented Systems	3
Automated Traffic Control	3
Automotive	3
Intelligence Surveillance Reconnaissance	2
Mobile Systems	2
Computer	2
Software Engineering	2

the first time period where IaaS made it in to the top domains. This is in line with the timeline of cloud based services as they rose to prominence.

Web services papers in this time period displayed a slightly different flavor to the previous time periods. Forms of validation were more popular in this phase. Performance [125] and integration [126] testing, probabilistic model checking [127, 128, 129], quality assurance [130], and evaluation [131] indicate that more emphasis was now being put on verifying the outcomes of self-adaptive solutions. It was not enough to claim that a system was self-adaptive, but the claims had to be backed up and tested. Frameworks formed part of the contributions to the research. With the push towards more holistic self-adaptive solutions, this was a natural progression. Multi-model [132], dynamic allocation [133], monitoring [134], and behavioral [135] frameworks highlight the variety of holistic generic approaches being explored. The RAINBOW framework still held influence during this period, with a framework called REFRACT extending RAINBOW to target fault avoidance [136]. Planning was a key component of this phase – automatic reconfiguration plans [137], adapting manager optimization [138], and plan generation techniques [139] all highlight the continued importance of forecasting in self-adaptive systems. Distributed techniques [140], fault localization [141], and architecture based self-adaptation [142] all were continued self-adaptive themes from the previous phase.

Robotics papers also shared an emphasis on building generic frameworks for self-adaptation. Architectural compilers [143], reference models [144], and testing frameworks [145] highlight some of the work done here. There was a continued focus on dynamic [146] and run-time [147] applications as well as verification of systems [148]. The modelling of uncertainty became a strong focus during this phase. As decisions in the system are pushed from design time to run time, the amount of possible outcomes for the system dramatically increases. The behavior of the system also becomes non-deterministic. Verification [149], consequence modelling [150], and latency modelling [128] all attest to this effort. A strong research theme of the robotics domain in self-adaptive systems is goal modelling. The goals of a robot and associated utility functions are highly important to successful behavior. Dealing with fuzzy goals [151], interactions [150], and learning [152] all contributed to this focus.

Networking was a domain that was well represented during this time period. With the explosion of the Internet and always-on devices, networking was a ripe domain to apply self-adaptive principles to. Consistent with the theme of frameworks during this time period, mathematical [153], scheduling [154], sensor modelling [155], and testing [156] frameworks were created. There was a similar emphasis on validation [157] and fault tolerance [158]. The RAINBOW framework was again used as an exemplar during this phase. It was applied to manage and monitor highly populated networks of devices [159]. Consistent with the themes from previous phases, self-organization [160] and self-reconfiguration [161, 162] were popular research areas in networking.

The IoT domain shares a high overlap with the networking domain as IoT solutions are essentially localized networks. The trends of the IoT research in this phase is consistent with the themes of generic frameworks [163, 164, 165, 166, 167] and validation [168, 169].

The IaaS domain entered the top domains in this phase. In the second half of the decade its popularity would explode. During this time, IaaS papers focused on regression testing [170], control theory [171], decision making [172, 173], transaction management [174], and on benchmarking and elasticity [175]. These themes would later on be developed and expanded.

There were only six reviews in this time period, the lowest proportion of any time period so far. This could be due to the reduced need because of the ongoing work. Reviews in this period focused on self-healing [19, 20], self-

Table 15: Categories during first half of the last decade (2011–2015)

Category	Count
Technological	37
Methodology	29
Empirical	9
Analytical	8
Perspective	8
Total	91

adaptation [7, 16], self-protection [21], and on control engineering approaches to self-adaptive system design [176].

Frameworks in ISR [177, 178], service oriented systems [179, 180, 181], software engineering [182, 183], and mobile systems [184, 185] highlight the trend of the theme in developing holistic solutions to self-adaptive systems. Continued trends are requirements [186, 187], dynamic solutions [188], multi-agent systems [189, 190], and utility [191] as well as reliability [192], with more abstract ideas like systems evaluation [193], uncertainty handling [194, 195], and feedback loops [196, 197] now getting covered.

Bio-inspired approaches were again present in this phase. These approaches have the understanding that self-adaptive systems are much like biological systems and that there is much inspiration to draw from nature. Papers discuss chemically inspired architectures for reusable models [198], as well as cloud based applications inspired by biological principles [199] and multi-objective control for self-adaptive software design [200]. These biological inspired approaches are a potential growth area for self-adaptive systems.

Other domain applications not listed include e-Commerce [201], UML [202], automotive [203], water networks [204], automated traffic management [205], fault recovery [206], video encoding [207], application containers [208], and human participation [209].

Tables 15 and 16 characterize this time period in terms of the prevalent paper categories and application domains, showing a large number of papers in the categories ‘Technological’ and ‘Methodology’, again dominated by the application area of Web Services.

Table 16: Application domains during first half of the last decade (2011–2015)

Application Domain	Count
Web Services	19
Robotics	11
Networking	10
IoT	7
Intelligence Surveillance Reconnaissance	6
Service Oriented Systems	5
Software Engineering	5
Review	5
IaaS	4
Mobile Systems	4
Bio-inspired	2
Control Engineering	2

4.5. 2016-2020 - The last decade - Second Half

The second half of the 2010s displayed a similar distribution to the first half, with technological and methodology papers being most frequent, followed by analytical, perspective, and empirical. This is not surprising as the developments of the field were at a comparable maturation stage.

The distribution of the domains during this period shows an interesting trend. Web services are no longer the most frequent domain, rather IoT and IaaS are the most frequent domains. There is a distinct trend of these cloud based technologies from obscurity (1990-2003) to niche (2003-2010) to growth (2011-2015) to now in the 2020s where cloud based services are mainstream. This trend is reflected in the rise of IoT and IaaS in the domain distribution of the papers.

IoT has become a popular application domain in the last five years of the 2010s. The number of devices with access to the Internet has increased exponentially in quantity but also variety. Devices are not just limited to phones and computers but extend to watches, cars, buildings, sensors, and more. The research in this phase gave rise to a number of exemplars in IoT. Exemplars can be generic such as artifacts or address specific self-adaptive problems. They are used as a demonstration of a working solution in the problem space. This was the period of time where exemplars began to be widely seen and used. DeltaIoT [210], an evaluation exemplar and

DingNet [211], a simulation exemplar highlight the work done. Carrying on from the work done in the previous time periods, frameworks and generic architectures were seen in HAFLoop [212], decentralized approaches [213, 214], and modelling frameworks [215]. Behavioral modelling of IoT behavior was a continued trend [216, 217, 218, 219]. Specifically, emergent behaviors were explored [220]. Emergent behavior is an important concept in self-adaptive systems. These behaviors are the byproduct of allowing decisions to be made at run time. When this occurs, the system may display new behavior not previously conceived or seen before. The appropriate handling of these behaviors is important to a large scale self-adaptive system like IoT. Common research focuses like new languages [221], evaluation and testing [222, 223], modelling [224], learning [225, 226, 227], recovery [228], uncertainty [229], and integration [230] were seen in this phase. A general review of self-improving system integration can be found in [231] and industrial experience reports in [232, 233].

IaaS became more of a focus during this period. As it went further into the decade, cloud based solutions became more and more common in business and hence in research. It became cheaper to rent out infrastructure in the cloud and outsource maintenance costs than to handle everything in-house. This is also reflected in research focusing on concurrent approaches [234, 235] or hierarchical systems [236].

From the previous time period, the research areas of testing [237], control theory [213], decision making [238, 239, 240], and elasticity [225] in IaaS were expanded on. New areas of research, such as trust [241], structural and parametric adaptation [242], monitoring [224, 243], modelling [244, 245], and service level maintenance [246] were established in the field. Trust in self-adaptive systems is an important concept. Even if the self-adaptation loops are robust and effective, without establishing trust for the system, using these systems in large scale or critical environments is infeasible. This has to to with the understood error rate of the self-adaptive system and the tolerance of the user. In some cases it may be acceptable to have a 20% error rate in a non-critical scenario but for another critical scenario like a Defence setting, even a 5% error rate may not be acceptable given the possible consequences. The research efforts in self-adaptive systems are usually split between structural and parametric adaptation. Structural adaptation involves modifying or improving the components of the system whereas parametric adaptation involves optimizing the configurable parameters of the system, leaving the components unchanged. Addressing both of the styles at once is an area of need and potential [242].

The research into web services has benefited from a 20 plus year build up. Consistent with the trend of this time period, exemplars were used to demonstrate the capabilities of self-adaptive web services using TCP communication [247]. Multi-agent systems [248], uncertainty [249], planning [250], models [251, 252], and programming concepts [253, 254] were all continued research themes into web services. The state of web services after 20 plus years has moved from foundational theory to generic frameworks and to exemplars. Even despite this trend, the various research themes are still being explored and mined for use after 20 years which indicates that there is more to learn.

Cyber-Physical Systems (CPS) are a combination of computation, networking, and physical processes where a physical component is controlled by a chip or software component. This time period is the first time CPS are seen. This indicates that they are a relatively new research area to self-adaptive systems. The dominating application (24%) to CPS is energy and the dominant adaptation mechanism is MAPE-K [255]. A new language, Adaptive CSP was developed to support compositional verification of systems [256]. Continuing with the trend of exemplars in this phase, DARTSim represents a simulation of UAVs on a reconnaissance mission communicating via TCP [257]. According to [258], a central concept in these systems is homeostasis, the capacity to maintain an operational state despite runtime uncertainty. This is addressed by four principles: collaborative sensing, faulty component isolation from adaptation, enhancing mode switching, and adjusting guards in mode switching. CPS are naturally employed in safety-critical environments as their small nature allows them to be embedded into any physical tool. The successful integration [259] and the tracability of these components [260] are critical to the field.

Continuing on with the last time period, bio inspired approaches were seen in this time period, addressing emergent behavior [261] and artificial DNA [262]. Security was a focus of this theme with guarantees [263] and verifications [264] being explored. The self-protection aspect of self-adaptive systems has been sprinkled amongst the time periods (with works focusing on trust [265, 266, 267] and on situational awareness [268]), however as these systems gain traction and popularity, there will be an increased need to secure these systems in the same fashion as micro-transactions are secured in financial institutions. It would not be surprising if in the next decade, security became a more prominent application domain.

There was a paper in this time period on smart factory or industry 4.0 [269]. Seen as the next progression in industrial activity, this application domain has potential to grow going in to the next decade. Exemplars were again seen in this period across other domains using architectural self-healing and self-optimization [270].

The frequencies of robotics and networking decreased in this phase. This could be because they are less popular or that there is some overlap between these domains and the top two domains, IoT and IaaS. The second reason is more likely. Planning [271, 272, 273, 274], testing [275], fault tolerance [276], and uncertainty [277, 278] all highlight common research trends seen before in the timelines, as is model-predictive control [279]. Mobile systems most likely also share the same similarities with overlap as networking and robotics to IoT and IaaS. They have been a consistent theme across the timelines and have a presence in this one with dynamic decisions [280], input space mapping [281] and emotion measurement [282]. A review of self-adaptive systems in the context of mobile systems is given in [283], one on monitoring self-adaptive applications within edge computing frameworks in [284], and one on learning in self-adaptive systems in [285].

The automotive application domain increased in this time period compared to the previous time period. This may be explained by the new found viability of smart cars and self-driving cars in recent years. Key trends for this domain were adaptive, scalable, and robust systems. These systems are proactively aware of latency and can act in swarms [286, 287, 288, 289, 290].

ISR has had a consistent presence across the timelines. Resilience [291], goal theory [292], control theory [293], and assurance [294] highlight the research efforts in this time period. ISR is an important application domain to self-adaptive systems. It enables real time situational awareness and allows analysts to make decisions based off current and useful information. In a Defence context, generating this intelligence from data is extremely important to the decision makers.

Other application domains included clonal plasticity [295], smart travels [296], agriculture [297], UML [298], system on chip [299], MAPE-K [300], traffic management [301], holonic systems [302], and managing support of recoonfigurable software components [303].

Table 17: Categories during second half of the last decade (2016–2020)

Category	Count
Technological	36
Methodology	22
Analytical	14
Perspective	11
Empirical	5
Total	88

Tables 17 and 18 characterize this time period in terms of the prevalent paper categories and application domains, underlining the focus on IoT and IaaS in recent years.

4.6. Summary

In this section, we briefly revisit the research questions set out at the beginning of this review to summarize our findings.

RQ1 What is the current state of the art in self-adaptive systems? The current state of the art in self-adaptive systems is focused on developing methodologies and technology in the area of cloud-based services, such as IoT and IaaS. Although research on self-adaptive systems tends to be diverse, empirical and analytical research is currently playing a smaller role, as are other application domains. The importance of self-adaptive systems is rapidly growing in areas such as bio-inspired approaches, security, and cyber physical systems.

RQ2 How has the state of the art evolved over time? In the 1990s, research on self-adaptive systems started with theoretical and model based papers to establish the foundations of the field. Practical implementations and frameworks together with forward-thinking perspective research gave rise to the rapid growth of the field in the 2000s and 2010s, with a need for and a trend towards holistic approaches and exemplars. Throughout the evolution of the field, researchers have published a large number of perspective papers to challenge the status quo and outline the needs of practitioners.

RQ3 Which are the application domains of self-adaptive systems over time? After an initial focus on networking, web services have dominated self-adaptive systems as an application area for much of the field's evolution, up until around 2015 when IoT and IaaS became the most frequent

Table 18: Application domains during second half of the last decade (2016–2020)

Application Domain	Count
IoT	16
IaaS	12
Web Services	9
Cyber Physical Systems	6
Automotive	5
Review	4
Mobile Systems	4
Intelligence Surveillance Reconnaissance	4
Robotics	3
Networking	3
Load Balancing	2
Service Oriented Systems	2
Software Engineering	2
Security	2
Bio-inspired	2

domains. From the beginning, the field has exhibited a large and diverse number of application domains, from robotics and networking to automotive and intelligence surveillance reconnaissance.

5. Threats to Validity and Limitations

Unlike related literature reviews on self-adaptive systems which characterize the state of the art of a narrow and specific aspect of self-adaptive systems such as requirements [5, 6], claims and evidence [7], quality attributes [8], and machine learning in the context of SAS [9], we took a broader view of the literature in this work, which necessarily limits the amount of detail presented for each of the 293 papers. The tables in the appendix provide the high-level overview from our review at a glance.

This systematic literature review was conducted with some assumptions. The dblp database was a suitable database to capture self-adaptive systems. Dblp is a computer science bibliography, and the review would not capture papers outside this bibliography in fields like medicine, science, and other engineering fields. We used the CORE ranking of a publication venue as a proxy for paper quality. The corresponding filtering step may have excluded high-quality papers relevant for our review that were published in other venues.

The application domains mentioned in this paper are subject to the interpretation of the papers. A paper may have multiple application domains but only one was chosen for each paper. This means there is some overlap across the domains. At best, it is useful to get a flavor of the types of papers in self-adaptive systems across the 30 year time period but it is not a comprehensive survey of all the types of domains.

The inclusion and exclusion criteria described in Section 3.2 bias the selection of primary studies, e.g., by using keywords for pre-filtering. These steps were necessary to handle the large amount of papers. The pre-filtering would likely have resulted in a different set of papers if we had considered abstracts and keywords in addition to titles when computing word frequency. Note that some of these concerns are mitigated by our use of snowballing to pick up papers that were missed through the initial search. Focusing on a single main contribution per paper also introduces bias since papers may have more than one contribution.

6. Conclusion

Self-adaptive systems research dates back to the 1990s where theoretical and model based papers established foundational self-adaptive theory. These theories gave rise to practical implementations and frameworks such as the RAINBOW framework in the early 2000s. During this time several perspective papers were published such as the seminal work 'The Vision of Autonomic Computing' which outlined the grand challenges of the field moving forward. The ramping up years of 2006-2010 were characterized by principles and ideas leading up to a need for more holistic generic approaches. In the first half of the 2010s the need for holistic generic approaches was met with several new frameworks. By the second half of the 2010s, these frameworks were extended to become exemplars, working solutions with real use cases. In the 2020s, if the popularity of self-adaptive systems continues to grow, these exemplars are likely to turn into mainstream adopted solutions.

The ebb and flow of the application domains across the time period show web services being most popular in the 2000s before IoT and IaaS papers joined them as the most popular in the 2010s. In the late 2010s domains like bio-inspired approaches, security, and cyber physical systems showed promise to grow heading into the 2020s. As time goes on, often an unknown disruptive solution arises that slowly makes its way to the top of the domains. In the 2020s technologies could arise like this from unlikely sources.

In their systematic review on claims and supporting evidence for self-adaptive systems from 2012, Weyns et al. [7] concluded that only a few systematic empirical studies had been undertaken at that point. This trend has not really changed over the last decade: while the overall number of papers on self-adaptive systems continues to grow, less than 8% of the papers identified in our systematic review focus on the empirical aspect, compared to 36% technological papers and 27% methodology papers. Perhaps encouragingly, many approaches are evaluated using real-world case studies, with less reliance on simulations. Going forward, we expect to see a shift towards empirical studies as the research field continues to mature, with industrial case studies in many of the application domains identified here.

For a young research field such as self-adaptive systems, a surprisingly large ratio of papers focus on reflecting on the current state of the field and/or providing a road map going forward (17% of the papers identified in our review). Due to the size of the field, very few of these perspective papers encompass self-adaptive systems as a whole, instead focusing on particular sub-classes of or challenges related to self-adaptive systems. In contrast, we provide a high-level overview of the field across categories and application domains.

In our future work, we aim to work towards closing some of the gaps identified in this systematic literature review, with a particular focus on systematic empirical studies. In line with Gerostathopoulos et al.'s recent study [304] which concluded that "most data of users and the environment used in experiments is synthetically generated", we aim to experiment with human subjects – an aspect that has not received much attention from the self-adaptive systems research community thus far. At the same time, the maturity of the field now allows for the development and deployment of such systems in real environments, with large-scale evaluations using the empirical methods that are well-established in other areas of software engineering [305]. Through our industry collaborators, we further will put particular focus on self-adaptation for cyber-physical systems. Cyber-physical systems have to handle uncertainty and change during operation, control their emergent behavior, and be scalable and tolerant to threats [255], yet their complexities introduce new challenges to self-adaptive systems that are difficult to capture without real-world evaluations.

Table 19: Analytical

		Type						Applic.		Contr.		Formal.		Evaluation Method							
Reference	Year	Algorithm	Architecture	Framework	Mathematical	Language	Definition	Case study	Simulated	Extension	Novel	Yes	No	Preliminary	Case Study	Industrial	Comparison	Human Subject	Quantitative	Unknown/None	Domain
[24]	1992	✓							✓		✓	✓							✓		Networking
[25]	1999				✓				✓		✓								✓		Networking
[54]	2005	✓						✓			✓	✓			✓						Networking
[37]	2005	✓						✓			✓	✓			✓						Web Services
[49]	2005			✓				✓			✓	✓			✓						Robotics
[266]	2006			✓					✓		✓	✓					✓				Security
[121]	2007			✓				✓			✓	✓			✓						Holonic Systems
[59]	2008	✓						✓		✓		✓					✓				Web Services
[124]	2009				✓			✓		✓		✓		✓							Physics
[93]	2009						✓				✓	✓								✓	Other
[73]	2009				✓			✓		✓		✓		✓							IoT
[101]	2010						✓	✓		✓		✓			✓						e-Commerce
[69]	2010	✓						✓			✓	✓			✓						Web Services
[198]	2011				✓			✓		✓		✓							✓		Bio-inspired
[187]	2011						✓	✓			✓	✓							✓		IoT
[153]	2011				✓			✓		✓		✓								✓	Networking
[182]	2011				✓			✓			✓	✓			✓						Software Engineering
[186]	2011				✓				✓		✓	✓							✓		Service-Oriented Systems
[178]	2012					✓		✓			✓	✓			✓						ISR
[155]	2012		✓					✓			✓	✓					✓				Networking
[152]	2013	✓							✓		✓	✓							✓		Robotics
[228]	2016	✓							✓	✓		✓							✓		IoT
[261]	2016	✓							✓	✓		✓		✓							Bio-inspired
[241]	2016	✓						✓			✓	✓							✓		IaaS
[276]	2016	✓							✓		✓	✓							✓		Networking
[235]	2016				✓				✓		✓	✓								✓	Other
[274]	2017				✓				✓	✓		✓					✓				IaaS
[220]	2017	✓				✓		✓			✓	✓			✓						IoT
[288]	2018	✓						✓			✓	✓							✓		Automotive
[240]	2018	✓							✓	✓			✓				✓				IaaS
[224]	2018	✓							✓		✓	✓							✓		IoT
[271]	2018	✓						✓			✓	✓			✓						Robotics
[272]	2018			✓				✓			✓	✓							✓		Robotics
[263]	2018			✓					✓	✓		✓					✓				Security
[250]	2019	✓							✓		✓	✓							✓		Web Services

Table 20: Technological (1/3)

		Type		Implementation						Applic.		Contr.		Goals		Evaluation Method							
Reference	Year	Human in the loop	Closed circle	Tool	Model	Framework	Language	Architecture	Algorithm	Case Study	Simulated	Extension	Novel	Goals	Utility	Preliminary	Case Study	Industrial	Comparison	Human Subject	Quantitative	Unknown/None	Domain
[23]	1990		✓		✓					✓			✓				✓						Software Engineering
[1]	1998		✓	✓							✓		✓								✓		IoT
[26]	1999		✓						✓		✓		✓								✓		Speech Recognition
[27]	2000		✓			✓				✓			✓	✓			✓						Robotics
[28]	2004	✓						✓		✓			✓	✓	✓		✓						Software Engineering
[42]	2004		✓	✓							✓	✓							✓				Load Balancing
[45]	2004		✓					✓			✓	✓		✓							✓		IaaS
[39]	2004		✓			✓					✓		✓	✓							✓		Bio-inspired
[34]	2004		✓					✓		✓			✓	✓	✓						✓		Web Services
[36]	2005	✓		✓						✓			✓				✓						Web Services
[52]	2005		✓			✓					✓		✓								✓		Software Stack
[40]	2005		✓	✓							✓		✓						✓				Bio-inspired
[104]	2006		✓				✓			✓			✓		✓		✓						e-Commerce
[63]	2006		✓		✓						✓	✓		✓	✓						✓		Networking
[57]	2006	✓					✓				✓		✓		✓						✓		Web Services
[265]	2006		✓			✓					✓		✓								✓		Security
[103]	2007		✓	✓							✓		✓	✓								✓	ISR
[122]	2007		✓	✓						✓			✓				✓						Video Encoding
[58]	2007		✓			✓					✓	✓										✓	Web Services
[98]	2008	✓			✓						✓		✓	✓								✓	Load Balancing
[74]	2008		✓		✓						✓		✓	✓							✓		IoT
[81]	2008		✓			✓				✓			✓	✓								✓	Robotics
[120]	2009		✓		✓					✓			✓	✓			✓						UML
[76]	2009		✓				✓			✓			✓	✓			✓						IoT
[78]	2009	✓						✓		✓			✓	✓			✓						IoT
[83]	2009		✓					✓		✓			✓	✓			✓						Robotics
[65]	2009	✓				✓				✓			✓						✓				Web Services
[60]	2009		✓		✓					✓			✓	✓							✓		Web Services
[95]	2010		✓					✓		✓			✓				✓						e-Commerce
[75]	2010		✓			✓				✓			✓	✓	✓		✓						IoT
[85]	2010		✓							✓			✓	✓			✓						Robotics
[87]	2010	✓					✓			✓			✓	✓			✓						Robotics
[192]	2011		✓		✓						✓		✓								✓		Programmer
[190]	2011		✓		✓						✓		✓	✓							✓		Resource Management
[206]	2011		✓				✓				✓		✓									✓	Software Stack
[165]	2011		✓		✓						✓		✓	✓	✓				✓				IoT
[169]	2011		✓						✓		✓		✓		✓						✓		IoT
[143]	2011		✓	✓							✓		✓	✓							✓		Robotics
[179]	2011		✓			✓					✓	✓									✓		Service-Oriented Systems

Table 21: Technological (2/3)

		Type		Implementation						Applic.		Contr.		Goals		Evaluation Method							
Reference	Year	Human in the loop	Closed circle	Tool	Model	Framework	Language	Architecture	Algorithm	Case Study	Simulated	Extension	Novel	Goals	Utility	Preliminary	Case Study	Industrial	Comparison	Human Subject	Quantitative	Unknown/None	Domain
[180]	2011		✓					✓			✓		✓	✓							✓		Service-Oriented Systems
[125]	2011		✓			✓					✓		✓	✓					✓				Web Services
[132]	2011		✓			✓				✓			✓	✓			✓						Web Services
[133]	2011		✓			✓				✓			✓				✓						Web Services
[199]	2012		✓			✓					✓		✓								✓		Bio-inspired
[163]	2012		✓			✓				✓			✓	✓			✓						IoT
[177]	2012		✓		✓					✓			✓	✓			✓						ISR
[160]	2012		✓					✓		✓			✓	✓							✓		Networking
[144]	2012		✓		✓					✓			✓	✓			✓						Robotics
[196]	2012		✓				✓				✓		✓			✓							Software Engineering
[183]	2012	✓					✓			✓			✓		✓						✓		Software Engineering
[203]	2013		✓				✓			✓			✓				✓						Automotive
[173]	2013		✓		✓					✓		✓		✓	✓						✓		Other
[184]	2013		✓			✓				✓			✓	✓			✓						Mobile Systems
[148]	2013		✓				✓			✓		✓									✓		Robotics
[208]	2013	✓		✓							✓		✓						✓				Software Engineering
[134]	2013		✓			✓				✓			✓	✓					✓				Web Services
[164]	2014		✓				✓			✓			✓				✓						IoT
[167]	2014		✓						✓	✓			✓								✓		Mobile Systems
[145]	2014								✓		✓		✓	✓							✓		Robotics
[147]	2014	✓			✓					✓			✓	✓	✓		✓						Robotics
[151]	2014		✓		✓					✓			✓	✓			✓						Robotics
[142]	2014		✓		✓					✓		✓		✓							✓		Web Services
[136]	2014		✓			✓				✓		✓			✓						✓		Web Services
[209]	2015	✓				✓				✓			✓	✓	✓				✓				ISR
[189]	2015		✓						✓	✓			✓	✓					✓				ISR
[158]	2015	✓		✓							✓		✓	✓							✓		Networking
[140]	2015		✓				✓			✓			✓				✓						Web Services
[138]	2015		✓	✓						✓			✓	✓			✓						Web Services
[128]	2015		✓		✓						✓		✓	✓	✓						✓		Web Services
[269]	2016		✓			✓				✓			✓				✓						Smart Factory
[221]	2016		✓				✓			✓		✓		✓			✓						IoT
[217]	2016		✓	✓						✓				✓			✓						IoT
[291]	2016		✓			✓				✓			✓				✓						ISR
[280]	2016		✓			✓					✓		✓		✓						✓		Mobile Systems
[278]	2016		✓						✓	✓			✓				✓						Networking
[234]	2016		✓			✓					✓		✓								✓		Software Engineering
[253]	2016		✓	✓						✓			✓	✓			✓						Web Services
[254]	2016		✓		✓						✓		✓	✓	✓				✓				Web Services

Table 22: Technological (3/3)

		Type		Implementation						Applic.		Contr.		Goals		Evaluation Method							
Reference	Year	Human in the loop	Closed circle	Tool	Model	Framework	Language	Architecture	Algorithm	Case Study	Simulated	Extension	Novel	Goals	Utility	Preliminary	Case Study	Industrial	Comparison	Human Subject	Quantitative	Unknown/None	Domain
[300]	2017		✓					✓			✓		✓	✓	✓				✓				MAPE-K
[296]	2017		✓				✓				✓		✓	✓					✓				Smart Traveller
[286]	2017		✓	✓						✓			✓								✓		Automotive
[244]	2017		✓		✓					✓			✓	✓							✓		IaaS
[210]	2017	✓		✓						✓			✓	✓					✓				IoT
[215]	2017		✓	✓						✓			✓						✓				IoT
[214]	2017		✓	✓							✓		✓								✓		IoT
[213]	2017		✓			✓				✓			✓		✓						✓		IoT
[293]	2017		✓	✓						✓			✓	✓			✓						ISR
[279]	2017		✓		✓						✓		✓	✓	✓				✓				Load Balancing
[243]	2017	✓		✓						✓		✓									✓		Service-Oriented Systems
[303]	2017	✓					✓			✓		✓									✓		Web Services
[248]	2017		✓	✓						✓			✓	✓							✓		Web Services
[252]	2017		✓		✓					✓			✓				✓						Web Services
[301]	2018		✓		✓					✓			✓	✓	✓		✓						Traffic Mgmt.
[289]	2018		✓		✓					✓			✓	✓			✓						Automotive
[256]	2018		✓	✓						✓			✓								✓		Cyber Physical Systems
[270]	2018	✓		✓							✓		✓	✓	✓						✓		Software Engineering
[247]	2018		✓	✓							✓		✓		✓						✓		Web Services
[297]	2019		✓	✓							✓		✓	✓	✓						✓		Agriculture
[257]	2019	✓		✓							✓		✓						✓				Cyber Physical Systems
[246]	2019		✓						✓		✓		✓								✓		IaaS
[211]	2019	✓		✓						✓			✓	✓							✓		IoT
[225]	2019		✓		✓					✓			✓	✓	✓						✓		IoT
[216]	2019		✓				✓			✓		✓			✓		✓						IoT
[226]	2019		✓		✓						✓		✓								✓		Load Balancing
[212]	2020	✓						✓		✓			✓	✓	✓						✓		IoT

Table 23: Perspective

		Type						Content				Evaluation Method							
Reference	Year	Survey	Review	Evaluation of framework	Reflection	Roadmap	Comparison	Taxonomy	Challenges	Future Work	Requirements	Preliminary	Case Study	Industrial	Comparison	Human Subject	Quantitative	Unknown/None	Domain
[11]	2003				✓				✓	✓	✓							✓	Review
[38]	2003				✓					✓								✓	Bio-inspired
[2]	2003				✓				✓	✓	✓							✓	Review
[48]	2004				✓					✓								✓	IoT
[33]	2004				✓					✓			✓						Web Services
[12]	2004				✓				✓									✓	Review
[53]	2004				✓			✓					✓						Networking
[29]	2004				✓					✓			✓						Review
[30]	2004				✓			✓										✓	Review
[55]	2004				✓				✓	✓	✓							✓	Software Engineering
[31]	2005		✓							✓								✓	Review
[32]	2005		✓							✓	✓		✓						Review
[13]	2005	✓							✓									✓	Review
[51]	2005				✓						✓						✓		Computer
[88]	2006				✓					✓								✓	Review
[90]	2006	✓								✓								✓	Review
[18]	2007	✓								✓			✓						Review
[105]	2008						✓			✓					✓				Software Engineering
[89]	2008		✓							✓							✓		Review
[14]	2008	✓							✓		✓							✓	Review
[106]	2009		✓							✓							✓		Software Engineering
[116]	2009				✓				✓				✓						Traffic Mgmt.
[91]	2009	✓								✓					✓				Review
[15]	2009		✓						✓	✓					✓				Review
[267]	2009		✓						✓	✓							✓		Security
[66]	2009			✓						✓			✓						Web Services
[108]	2010		✓							✓					✓				Management
[94]	2010				✓				✓				✓						Traffic Mgmt.
[77]	2010				✓						✓						✓		IoT
[79]	2010				✓					✓			✓						IoT
[92]	2010					✓			✓						✓				Review
[70]	2010				✓						✓		✓						Web Services
[19]	2011	✓								✓					✓				Review
[176]	2012	✓						✓									✓		Control Engineering
[7]	2012		✓							✓					✓				Review
[21]	2012	✓						✓							✓				Review
[159]	2013			✓						✓			✓						Networking
[20]	2015	✓								✓					✓				Review
[16]	2015	✓						✓	✓	✓					✓				Review
[139]	2015						✓		✓	✓					✓				Web Services
[302]	2016		✓							✓			✓						Holonic Systems
[255]	2016		✓							✓					✓				Cyber Physical Systems
[242]	2016						✓	✓		✓					✓				IaaS
[223]	2017				✓					✓					✓				Software Engineering
[298]	2018		✓							✓					✓				UML
[284]	2018		✓						✓	✓					✓				IaaS
[233]	2018						✓	✓		✓			✓						Review
[231]	2018		✓						✓								✓		Review
[236]	2018		✓							✓			✓						Review
[283]	2019		✓						✓	✓							✓		Mobile Systems
[285]	2019		✓						✓				✓						Review

Table 24: Methodology (1/2)

		Type						Applic.		Contr.		Evaluation Method							
Reference	Year	General architecture	New Framework	Analysis Technique	New Pattern	Formal Criteria	New Approach	Case study	Simulated	Extension	Novel	Preliminary	Case Study	Industrial	Comparison	Human Subject	Quantitative	Unknown/None	Domain
[43]	2004	✓							✓		✓						✓		Load Balancing
[50]	2005						✓	✓			✓								Automotive
[46]	2005		✓					✓			✓						✓		IoT
[35]	2005						✓	✓			✓		✓						Web Services
[47]	2005						✓	✓			✓								IoT
[117]	2006						✓	✓			✓		✓						ISR
[102]	2006						✓	✓			✓		✓				✓		Mobile Systems
[118]	2006	✓						✓			✓							✓	Speech Recognition
[268]	2006		✓					✓			✓				✓				Security
[113]	2007			✓				✓			✓						✓		Programmer
[111]	2008						✓	✓			✓		✓						Computer
[112]	2008		✓					✓			✓						✓		Computer
[72]	2008						✓		✓	✓					✓				IoT
[82]	2008						✓		✓		✓							✓	Robotics
[84]	2008						✓		✓		✓				✓				Robotics
[123]	2008			✓					✓		✓						✓		Networking
[61]	2008						✓	✓		✓							✓		Web Services
[64]	2008						✓	✓			✓		✓						Web Services
[62]	2008						✓		✓		✓						✓		Web Services
[107]	2009						✓		✓		✓						✓		System on Chip
[97]	2009						✓	✓			✓				✓				Service-Oriented Systems
[119]	2009		✓					✓			✓		✓						Service-Oriented Systems
[86]	2009						✓	✓			✓		✓						Robotics
[67]	2009						✓	✓			✓		✓						Web Services
[114]	2010						✓	✓			✓		✓						Automotive
[109]	2010						✓	✓			✓				✓				Mobile Systems
[80]	2010		✓					✓			✓						✓		IoT
[71]	2010						✓		✓		✓						✓		Web Services
[170]	2011						✓		✓		✓						✓		IaaS
[154]	2011		✓					✓			✓						✓		Networking
[194]	2011						✓	✓			✓						✓		Robotics
[150]	2011						✓	✓			✓						✓		Robotics
[193]	2011		✓					✓		✓							✓		Other
[126]	2011						✓	✓			✓		✓						Web Services
[161]	2012						✓	✓			✓						✓		Networking
[146]	2012		✓						✓		✓						✓		Robotics
[188]	2012						✓	✓			✓				✓				Software Engineering
[174]	2012				✓			✓			✓						✓		Service-Oriented Systems
[127]	2012						✓	✓							✓				Web Services
[130]	2012						✓		✓		✓						✓		Web Services
[202]	2013						✓	✓		✓			✓						UML
[171]	2013						✓	✓			✓		✓						IaaS
[185]	2013		✓					✓			✓		✓						Mobile Systems
[181]	2013			✓					✓		✓						✓		Service-Oriented Systems
[131]	2013						✓	✓			✓				✓				Web Services

Table 25: Methodology (2/2)

		Type						Applic.		Contr.		Evaluation Method							
Reference	Year	General architecture	New Framework	Analysis Technique	New Pattern	Formal Criteria	New Approach	Case study	Simulated	Extension	Novel	Preliminary	Case Study	Industrial	Comparison	Human Subject	Quantitative	Unknown/None	Domain
[135]	2013		✓					✓			✓		✓						Web Services
[207]	2014						✓	✓			✓		✓						Video Encoding
[166]	2014						✓	✓			✓		✓						IoT
[149]	2014						✓	✓			✓						✓		Robotics
[141]	2014					✓				✓					✓				Web Services
[129]	2014			✓				✓			✓		✓						Web Services
[168]	2015		✓					✓			✓						✓		IoT
[195]	2015						✓	✓			✓		✓						ISR
[200]	2015						✓	✓			✓						✓		Mobile Systems
[156]	2015		✓					✓			✓				✓				Networking
[128]	2015						✓	✓			✓								Web Services
[137]	2015						✓		✓		✓						✓		Web Services
[262]	2016						✓	✓			✓						✓		Bio-inspired
[227]	2016						✓	✓		✓			✓						Web Services
[260]	2016						✓	✓			✓						✓		Cyber Physical Systems
[238]	2016						✓	✓			✓						✓		IaaS
[239]	2016						✓		✓		✓						✓		IaaS
[292]	2016						✓	✓			✓		✓						ISR
[281]	2016						✓	✓			✓				✓				Mobile Systems
[219]	2016						✓		✓		✓				✓				Service-Oriented Systems
[299]	2017						✓	✓			✓						✓		System on Chip
[287]	2017						✓		✓		✓				✓				Automotive
[245]	2017						✓	✓			✓		✓						IaaS
[222]	2017						✓	✓			✓		✓						IoT
[295]	2018			✓				✓			✓						✓		Clonal Plasticity
[232]	2018						✓	✓			✓						✓		Retrofitting systems
[259]	2018						✓	✓			✓		✓						Cyber Physical Systems
[294]	2018					✓		✓			✓		✓						ISR
[277]	2018						✓	✓			✓		✓						Networking
[264]	2018						✓	✓			✓		✓						Security
[251]	2018						✓	✓			✓		✓						Web Services
[258]	2019			✓				✓			✓		✓						Cyber Physical Systems
[229]	2019						✓	✓			✓		✓						IoT
[230]	2019	✓						✓		✓			✓						IoT

Table 26: Empirical

		Testing		Strategy		Applic.		Adaptation Technique		Evaluation Method							
Reference	Year	Design-time	Run-time	Monitoring	Adaptation	Case Study	Simulated	Parameter	Component	Preliminary	Case Study	Industrial	Comparison	Human Subject	Quantitative	Unknown/None	Domain
[41]	2004			✓			✓	✓							✓		Load Balancing
[56]	2004		✓		✓	✓		✓							✓		ISR
[44]	2005		✓	✓	✓		✓								✓		Web Services
[115]	2006		✓	✓	✓	✓		✓	✓		✓						Service-Oriented Systems
[110]	2009		✓	✓	✓	✓		✓			✓						Automotive
[99]	2009		✓		✓		✓	✓							✓		Automotive
[100]	2009		✓	✓	✓	✓		✓					✓				Traffic Mgmt.
[68]	2009			✓		✓		✓			✓						Web Services
[96]	2010		✓	✓	✓			✓							✓		Water Networks
[191]	2011			✓	✓	✓			✓		✓						ISR
[201]	2012		✓	✓	✓	✓		✓			✓				✓		e-Commerce
[197]	2012		✓	✓	✓	✓		✓			✓						Control Engineering
[162]	2012				✓		✓	✓							✓		Networking
[205]	2013		✓	✓	✓		✓	✓							✓		Traffic Mgmt.
[204]	2013			✓	✓		✓	✓							✓		Water Networks
[172]	2014		✓	✓	✓	✓		✓			✓		✓		✓		IaaS
[175]	2015		✓	✓	✓	✓		✓			✓						IaaS
[157]	2015		✓	✓	✓	✓		✓							✓		Networking
[273]	2017		✓	✓	✓	✓		✓				✓					IaaS
[282]	2017		✓	✓	✓	✓		✓							✓		Mobile Systems
[290]	2018						✓								✓		Automotive
[237]	2018		✓	✓	✓		✓	✓					✓				IaaS
[275]	2018		✓	✓	✓	✓		✓			✓						Robotics

References

[1] P. Oreizy, N. Medvidovic, R. Taylor, Architecture-based runtime software evolution, in: Proceedings of the 20th International Conference on Software Engineering, IEEE Comput. Soc, 1998, pp. 177–186. doi:10.1109/icse.1998.671114.

[2] J. Kephart, D. Chess, The vision of autonomic computing, Computer 36 (1) (2003) 41–50. doi:10.1109/mc.2003.1160055.

[3] J. Kephart, M. Parashar, V. Sunderam, R. Das, Message from the general chairs and program chairs, in: International Conference on Autonomic Computing, IEEE Computer Society, Los Alamitos, CA, USA, 2004.

[4] D. Weyns, Engineering self-adaptive software systems – an organized tour, in: 2018 IEEE 3rd International Workshops on Foundations and Applications of Self* Systems (FAS*W), IEEE, 2018, pp. 1–2. doi:10.1109/fas-w.2018.00012.

[5] Z. Yang, Z. Li, Z. Jin, Y. Chen, A systematic literature review of requirements modeling and analysis for self-adaptive systems, in: Requirements Engineering: Foundation for Software Quality, Springer, Springer International Publishing, 2014, pp. 55–71. doi:10.1007/978-3-319-05843-6_5.

[6] S. Sucipto, R. S. Wahono, A systematic literature review of requirements engineering for self-adaptive systems, Journal of Software Engineering 1 (1) (2015) 17–27.

[7] D. Weyns, M. U. Iftikhar, S. Malek, J. Andersson, Claims and supporting evidence for self-adaptive systems: A literature study, in: 2012 7th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS), IEEE, 2012, pp. 89–98. doi:10.1109/seams.2012.6224395.

[8] S. Mahdavi-Hezavehi, V. H. Durelli, D. Weyns, P. Avgeriou, A systematic literature review on methods that handle multiple quality attributes in architecture-based self-adaptive systems, Information and Software Technology 90 (2017) 1–26. doi:10.1016/j.infsof.2017.03.013.

[9] T. R. D. Saputri, S.-W. Lee, The application of machine learning in self-adaptive systems: A systematic literature review, IEEE Access 8 (2020) 205948–205967. doi:10.1109/access.2020.3036037.

[10] W. Chainbi, Why applying agent technology to autonomic computing?, Frontiers in Artificial Intelligence and Computing 135 (2005) 282.

[11] A. G. Ganek, T. A. Corbi, The dawning of the autonomic computing era, IBM Systems Journal 42 (1) (2003) 5–18. doi:10.1147/sj.421.0005.

[12] A. Ganek, C. Hilkner, J. Sweitzer, B. Miller, J. Hellerstein, The response to IT complexity: autonomic computing, in: Third IEEE International Symposium on Network Computing and Applications, 2004. (NCA 2004). Proceedings., IEEE, 2004, pp. 151–157. doi:10.1109/nca.2004.1347772.

[13] J. O. Kephart, Research challenges of autonomic computing, in: Proceedings of the 27th international conference on Software engineering - ICSE '05, ACM Press, 2005, pp. 15–22. doi:10.1145/1062455.1062464.

[14] M. C. Huebscher, J. A. McCann, A survey of autonomic computing—degrees, models, and applications, ACM Computing Surveys 40 (3) (2008) 1–28. doi:10.1145/1380584.1380585.

[15] M. Salehie, L. Tahvildari, Self-adaptive software, ACM Transactions on Autonomous and Adaptive Systems 4 (2) (2009) 1–42. doi:10.1145/1516533.1516538.

[16] C. Krupitzer, F. M. Roth, S. VanSyckel, G. Schiele, C. Becker, A survey on engineering approaches for self-adaptive systems, Pervasive and Mobile Computing 17 (PB) (2015) 184–206. doi:10.1016/j.pmcj.2014.09.009.

[17] E. G. Pereira, R. Pereira, A. Taleb-Bendiab, Performance evaluation for self-healing distributed services and fault detection mechanisms, Journal of Computer and System Sciences 72 (7) (2006) 1172–1182. doi:10.1016/j.jcss.2005.12.008.

[18] D. Ghosh, R. Sharman, H. R. Rao, S. Upadhyaya, Self-healing systems — survey and synthesis, Decision Support Systems 42 (4) (2007) 2164–2185. doi:10.1016/j.dss.2006.06.011.

[19] H. Psaier, S. Dustdar, A survey on self-healing systems: approaches and systems, Computing 91 (1) (2010) 43–73. doi:10.1007/s00607-010-0107-y.

[20] C. Schneider, A. Barker, S. Dobson, A survey of self-healing systems frameworks, Software: Practice and Experience 45 (10) (2014) 1375–1398. doi:10.1002/spe.2250.

[21] E. Yuan, S. Malek, A taxonomy and survey of self-protecting software systems, in: 2012 7th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS), IEEE, 2012, p. 109–118. doi:10.1109/seams.2012.6224397.

[22] B. Kitchenham, O. P. Brereton, D. Budgen, M. Turner, J. Bailey, S. Linkman, Systematic literature reviews in software engineering – a systematic literature review, Information and Software Technology 51 (1) (2009) 7–15. doi:10.1016/j.infsof.2008.09.009.

[23] J. Kramer, J. Magee, The evolving philosophers problem: dynamic change management, IEEE Transactions on Software Engineering 16 (11) (1990) 1293–1306. doi:10.1109/32.60317.

[24] A. Cheng, Self-stabilizing real-time rule-based systems, in: [1992] Proceedings 11th Symposium on Reliable Distributed Systems, IEEE Comput. Soc. Press, 1992, pp. 172–179. doi:10.1109/reldis.1992.235129.

[25] J. Beauquier, B. Bérard, L. Fribourg, A new rewrite method for proving convergence of self-stabilizing systems, in: Lecture Notes in Computer Science, Springer Berlin Heidelberg, 1999, pp. 240–255. doi:10.1007/3-540-48169-9_17.

[26] K. Yamauchi, M. Oota, N. Ishii, A self-supervised learning system for pattern recognition by sensory integration, Neural Networks 12 (10) (1999) 1347–1358. doi:10.1016/s0893-6080(99)00064-7.

[27] T. Q. Pham, K. R. Dixon, P. K. Khosla, Software systems facilitating self-adaptive control software, in: International Conference on Intelligent Robots and Systems (IROS), Vol. 2, 2000, pp. 1094–1100 vol.2. doi:10.1109/IROS.2000.893165.

[28] D. Garlan, S.-W. Cheng, A.-C. Huang, B. Schmerl, P. Steenkiste, Rainbow: architecture-based self-adaptation with reusable infrastructure, Computer 37 (10) (2004) 46–54. doi:10.1109/mc.2004.175.

[29] D. Hales, Sociologically inspired approaches for self-*: Examples and prospects, in: Self-star Properties in Complex Information Systems, Springer Berlin Heidelberg, 2005, pp. 433–445. doi:10.1007/11428589_28.

[30] R. de Lemos, The conflict between self-* capabilities and predictability, in: Self-star Properties in Complex Information Systems, Springer Berlin Heidelberg, 2005, pp. 219–228. doi:10.1007/11428589_15.

[31] O. Babaoglu, M. Jelasity, A. Montresor, C. Fetzer, S. Leonardi, A. van Moorsel, M. van Steen, The self-star vision, in: Self-star Properties in Complex Information Systems, Springer Berlin Heidelberg, 2005, pp. 1–20. doi:10.1007/11428589_1.

[32] S.-W. Cheng, D. Garlan, B. Schmerl, Making self-adaptation an engineering reality, in: Self-star Properties in Complex Information Systems, Springer Berlin Heidelberg, 2005, pp. 158–173. doi:10.1007/11428589_11.

[33] R. Barrett, P. P. Maglio, E. Kandogan, J. Bailey, Usable autonomic computing systems: The system administrators' perspective, Advanced Engineering Informatics 19 (3) (2005) 213–221. doi:10.1016/j.aei.2004.11.001.

[34] W. Walsh, G. Tesauro, J. Kephart, R. Das, Utility functions in autonomic systems, in: International Conference on Autonomic Computing, 2004. Proceedings., IEEE, 2004, pp. 70–77. doi:10.1109/icac.2004.1301349.

[35] V. Kapoor, Services and autonomic computing: a practical approach for designing manageability, in: 2005 IEEE International Confer-

ence on Services Computing (SCC'05) Vol-1, IEEE, 2005, pp. 41–48. doi:10.1109/scc.2005.88.

[36] G.-J. Houben, Z. Fiala, K. van der Sluijs, M. Hinz, Building self-managing web information systems from generic components, in: Advanced Information Systems Engineering (CAiSE) Workshops, FEUP Edições, Porto, 2005, pp. 53–67.

[37] K. Wolter, Self-management of systems through automatic restart, in: Self-star Properties in Complex Information Systems, Springer Berlin Heidelberg, 2005, pp. 189–203. doi:10.1007/11428589_13.

[38] F. Heylighen, C. Gershenson, S. Staab, G. Flake, D. Pennock, D. Fain, D. D. Roure, K. Aberer, W.-M. Shen, O. Dousse, P. Thiran, Neurons, viscose fluids, freshwater polyp hydra-and self-organizing information systems, IEEE Intelligent Systems 18 (4) (2003) 72–86. doi:10.1109/mis.2003.1217631.

[39] M. Nowostawski, M. Purvis, A. Gecow, Software self-adaptability by means of artificial evolution, in: Computational Science - ICCS 2004, Springer Berlin Heidelberg, 2004, pp. 552–559. doi:10.1007/978-3-540-24688-6_72.

[40] J. Yang, H. Chen, S. Hariri, M. Parashar, Autonomic runtime manager for adaptive distributed applications, in: HPDC-14. Proceedings. 14th IEEE International Symposium on High Performance Distributed Computing, 2005., IEEE, 2005, pp. 69–78. doi:10.1109/hpdc.2005.1520937.

[41] M. Bennani, D. Menasce, Assessing the robustness of self-managing computer systems under highly variable workloads, in: International Conference on Autonomic Computing, 2004. Proceedings., IEEE, 2004, pp. 62–69. doi:10.1109/icac.2004.1301348.

[42] Z. Kurmas, K. Keeton, Using the distiller to direct the development of self-configuration software, in: International Conference on Autonomic Computing, 2004. Proceedings., IEEE, 2004, pp. 172–179. doi:10.1109/icac.2004.1301361.

[43] S. White, J. Hanson, I. Whalley, D. Chess, J. Kephart, An architectural approach to autonomic computing, in: International Conference

on Autonomic Computing, 2004. Proceedings., IEEE, 2004, pp. 2–9. doi:10.1109/icac.2004.1301340.

[44] S. Sivasubramanian, G. Pierre, M. van Steen, Autonomic data placement strategies for update-intensiveWeb applications, in: First International Workshop on Advanced Architectures and Algorithms for Internet Delivery and Applications (AAA-IDEA'05), IEEE, IEEE, 2005, pp. 2–9. doi:10.1109/AAA-IDEA.2005.4.

[45] M. Mikic-Rakic, N. Medvidovic, Support for disconnected operation via architectural self-reconfiguration, in: International Conference on Autonomic Computing, 2004. Proceedings., IEEE, 2004, pp. 114–121. doi:10.1109/icac.2004.1301354.

[46] N. Kandasamy, S. Abdelwahed, G. C. Sharp, J. P. Hayes, An online control framework for designing self-optimizing computing systems: Application to power management, in: Self-star Properties in Complex Information Systems, Springer Berlin Heidelberg, 2005, pp. 174–188. doi:10.1007/11428589_12.

[47] M. E. Shin, Self-healing components in robust software architecture for concurrent and distributed systems, Science of Computer Programming 57 (1) (2005) 27–44. doi:10.1016/j.scico.2004.10.003.

[48] P. Andras, B. G. Charlton, Self-aware software – will it become a reality?, in: Self-star Properties in Complex Information Systems, Springer Berlin Heidelberg, 2005, pp. 229–259. doi:10.1007/11428589_16.

[49] Y. Zhao, M. Kardos, S. Oberthür, F. J. Rammig, Comprehensive verification framework for dependability of self-optimizing systems, in: Automated Technology for Verification and Analysis, Springer Berlin Heidelberg, 2005, pp. 39–53. doi:10.1007/11562948_6.

[50] T. D. Wolf, G. Samaey, T. Holvoet, D. Roose, Decentralised autonomic computing: Analysing self-organising emergent behaviour using advanced numerical methods, in: Second International Conference on Autonomic Computing (ICAC'05), IEEE, 2005, pp. 52–63. doi:10.1109/icac.2005.20.

[51] J. P. Sousa, V. Poladian, D. Garlan, B. R. Schmerl, Capitalizing on awareness of user tasks for guiding self-adaptation, in: Advanced Information Systems Engineering (CAiSE) Workshops, FEUP Edições, Porto, 2005, pp. 83–96.

[52] J. Wildstrom, P. Stone, E. Witchel, R. Mooney, M. Dahlin, Towards self-configuring hardware for distributed computer systems, in: Second International Conference on Autonomic Computing (ICAC'05), IEEE, 2005, pp. 241–249. doi:10.1109/icac.2005.63.

[53] I. Gupta, S. Ko, N. Thompson, M. Nagda, C. Devaraj, R. Morales, J. A. Patel, A case for design methodology research in self-* distributed systems, in: Self-star Properties in Complex Information Systems, Springer Berlin Heidelberg, 2005, pp. 260–272. doi:10.1007/11428589_17.

[54] D. Breitgand, E. Henis, O. Shehory, Automated and adaptive threshold setting: Enabling technology for autonomy and self-management, in: Second International Conference on Autonomic Computing (ICAC'05), IEEE, 2005, pp. 204–215. doi:10.1109/icac.2005.11.

[55] P. McKinley, S. Sadjadi, E. Kasten, B. Cheng, Composing adaptive software, Computer 37 (7) (2004) 56–64. doi:10.1109/mc.2004.48.

[56] N. Bulusu, J. Heidemann, D. Estrin, T. Tran, Self-configuring localization systems, ACM Transactions on Embedded Computing Systems 3 (1) (2004) 24–60. doi:10.1145/972627.972630.

[57] S.-W. Cheng, D. Garlan, B. Schmerl, Architecture-based self-adaptation in the presence of multiple objectives, in: Proceedings of the 2006 international workshop on Self-adaptation and self-managing systems - SEAMS '06, ACM Press, 2006, p. 2–8. doi:10.1145/1137677.1137679.

[58] H. Meling, An architecture for self-healing autonomous object groups, in: Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2007, pp. 156–168. doi:10.1007/978-3-540-73547-2_18.

[59] S. Sicard, F. Boyer, N. D. Palma, Using components for architecture-based management, in: Proceedings of the 13th international confer-

ence on Software engineering - ICSE '08, ACM Press, 2008, p. 101–110. doi:10.1145/1368088.1368103.

[60] I. Epifani, C. Ghezzi, R. Mirandola, G. Tamburrelli, Model evolution by run-time parameter adaptation, in: 2009 IEEE 31st International Conference on Software Engineering, IEEE, 2009, pp. 111–121. doi:10.1109/icse.2009.5070513.

[61] J. Pastrana, E. Pimentel, M. Katrib, Composition of self-adapting components for customizable systems, The Computer Journal 51 (4) (2007) 481–496. doi:10.1093/comjnl/bxm094.

[62] N. Bartolini, G. Bongiovanni, S. Silvestri, Self-$*$ through self-learning: Overload control for distributed web systems, Computer Networks 53 (5) (2009) 727–743. doi:10.1016/j.comnet.2008.11.015.

[63] V. Bhat, M. Parashar, H. Liu, M. Khandekar, N. Kandasamy, S. Abdelwahed, Enabling self-managing applications using model-based online control strategies, in: 2006 IEEE International Conference on Autonomic Computing, IEEE, IEEE, 2006, pp. 15–24. doi:10.1109/icac.2006.1662377.

[64] A. Carzaniga, A. Gorla, M. Pezzè, Self-healing by means of automatic workarounds, in: Proceedings of the 2008 international workshop on Software engineering for adaptive and self-managing systems - SEAMS '08, ACM Press, 2008, p. 17–24. doi:10.1145/1370018.1370023.

[65] K. M. Chan, J. Bishop, The design of a self-healing composition cycle for web services, in: 2009 ICSE Workshop on Software Engineering for Adaptive and Self-Managing Systems, IEEE, 2009, pp. 20 – 27. doi:10.1109/seams.2009.5069070.

[66] S.-W. Cheng, D. Garlan, B. Schmerl, Evaluating the effectiveness of the rainbow self-adaptive system, in: 2009 ICSE Workshop on Software Engineering for Adaptive and Self-Managing Systems, IEEE, 2009, pp. 132–141. doi:10.1109/seams.2009.5069082.

[67] C. E. da Silva, R. de Lemos, Using dynamic workflows for coordinating self-adaptation of software systems, in: 2009 ICSE Workshop on Software Engineering for Adaptive and Self-Managing Systems, IEEE, 2009, pp. 86–95. doi:10.1109/seams.2009.5069077.

[68] H. Ghanbari, M. Litoiu, Identifying implicitly declared self-tuning behavior through dynamic analysis, in: ICSE Workshop on Software Engineering for Adaptive and Self-Managing Systems (SEAMS), IEEE, 2009, pp. 48–57. doi:10.1109/seams.2009.5069073.

[69] N. H. Kacem, A. H. Kacem, K. Drira, A formal approach to enforcing consistency in self-adaptive systems, in: Software Architecture, Springer Berlin Heidelberg, 2010, pp. 279–294. doi:10.1007/978-3-642-15114-9_21.

[70] E. Kaddoum, C. Raibulet, J.-P. Georgé, G. Picard, M.-P. Gleizes, Criteria for the evaluation of self-* systems, in: Proceedings of the 2010 ICSE Workshop on Software Engineering for Adaptive and Self-Managing Systems - SEAMS '10, ACM Press, 2010, p. 29–38. doi:10.1145/1808984.1808988.

[71] J. Philippe, N. D. Palma, F. Boyer, et Olivier Gruber, Self-adaptation of service level in distributed systems, Software: Practice and Experience 40 (2010). doi:10.1002/spe.957.

[72] G. Clair, E. Kaddoum, M.-P. Gleizes, G. Picard, Self-regulation in self-organising multi-agent systems for adaptive and intelligent manufacturing control, in: 2008 Second IEEE International Conference on Self-Adaptive and Self-Organizing Systems, IEEE, 2008, pp. 107–116. doi:10.1109/saso.2008.19.

[73] F. Nafz, F. Ortmeier, H. Seebach, J.-P. Steghöfer, W. Reif, A universal self-organization mechanism for role-based organic computing systems, in: Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2009, pp. 17–31. doi:10.1007/978-3-642-02704-8_3.

[74] P. Hu, J. Indulska, R. Robinson, An autonomic context management system for pervasive computing, in: 2008 Sixth Annual IEEE International Conference on Pervasive Computing and Communications (PerCom), IEEE, 2008, pp. 213–223. doi:10.1109/percom.2008.56.

[75] R. Calinescu, L. Grunske, M. Kwiatkowska, R. Mirandola, G. Tamburrelli, Dynamic QoS management and optimization in service-based systems, IEEE Transactions on Software Engineering 37 (3) (2011) 387–409. doi:10.1109/tse.2010.92.

[76] J. Whittle, P. Sawyer, N. Bencomo, B. H. Cheng, J.-M. Bruel, RELAX: Incorporating uncertainty into the specification of self-adaptive systems, in: 2009 17th IEEE International Requirements Engineering Conference, IEEE, 2009, pp. 79–88. `doi:10.1109/re.2009.36`.

[77] P. Sawyer, N. Bencomo, J. Whittle, E. Letier, A. Finkelstein, Requirements-aware systems: A research agenda for RE for self-adaptive systems, in: 2010 18th IEEE International Requirements Engineering Conference, IEEE, 2010, pp. 95–103. `doi:10.1109/re.2010.21`.

[78] F. Dalpiaz, P. Giorgini, J. Mylopoulos, An architecture for requirements-driven self-reconfiguration, in: Notes on Numerical Fluid Mechanics and Multidisciplinary Design, Springer International Publishing, 2009, pp. 246–260. `doi:10.1007/978-3-642-02144-2\_22`.

[79] D. A. Menasce, J. P. Sousa, S. Malek, H. Gomaa, Qos architectural patterns for self-architecting software systems, in: Proceeding of the 7th international conference on Autonomic computing - ICAC '10, ACM Press, 2010, p. 195–204. `doi:10.1145/1809049.1809084`.

[80] A. Elkhodary, N. Esfahani, S. Malek, FUSION, in: Proceedings of the eighteenth ACM SIGSOFT international symposium on Foundations of software engineering - FSE '10, ACM Press, 2010, pp. 7–16. `doi:10.1145/1882291.1882296`.

[81] M. Morandini, L. Penserini, A. Perini, Towards goal-oriented development of self-adaptive systems, in: Proceedings of the 2008 international workshop on Software engineering for adaptive and self-managing systems - SEAMS '08, ACM Press, 2008, p. 9–16. `doi:10.1145/1370018.1370021`.

[82] D. Sykes, W. Heaven, J. Magee, J. Kramer, From goals to components, in: Proceedings of the 2008 international workshop on Software engineering for adaptive and self-managing systems - SEAMS '08, ACM Press, 2008, p. 1–8. `doi:10.1145/1370018.1370020`.

[83] Y. Wang, J. Mylopoulos, Self-repair through reconfiguration: A requirements engineering approach, in: 2009 IEEE/ACM International

Conference on Automated Software Engineering, IEEE, 2009, pp. 257–268. doi:10.1109/ase.2009.66.

[84] B. Satzger, A. Pietzowski, W. Trumler, T. Ungerer, Using automated planning for trusted self-organising organic computing systems, in: Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2008, pp. 60–72. doi:10.1007/978-3-540-69295-9_7.

[85] D. Weyns, S. Malek, J. Andersson, FORMS, in: Proceeding of the 7th international conference on Autonomic computing - ICAC '10, ACM Press, 2010, p. 205–214. doi:10.1145/1809049.1809078.

[86] D. Kim, S. Park, Reinforcement learning-based dynamic adaptation planning method for architecture-based self-managed software, in: 2009 ICSE Workshop on Software Engineering for Adaptive and Self-Managing Systems, IEEE, 2009, pp. 76–85. doi:10.1109/seams.2009.5069076.

[87] H. Tajalli, J. Garcia, G. Edwards, N. Medvidovic, PLASMA, in: Proceedings of the IEEE/ACM international conference on Automated software engineering - ASE '10, ACM Press, 2010, pp. 467–476. doi:10.1145/1858996.1859092.

[88] M. Jelasity, O. Babaoglu, R. Laddaga, R. Nagpal, F. Zambonelli, E. Sirer, H. Chaouchi, M. Smirnov, Interdisciplinary research: Roles for self-organization, IEEE Intelligent Systems 21 (2) (2006) 50–58. doi:10.1109/mis.2006.30.

[89] H. Kasinger, B. Bauer, J. Denzinger, The meaning of semiochemicals to the design of self-organizing systems, in: 2008 Second IEEE International Conference on Self-Adaptive and Self-Organizing Systems, IEEE, 2008, pp. 139–148. doi:10.1109/saso.2008.51.

[90] X. Li, H. Kang, P. Harrington, J. Thomas, Autonomic and trusted computing paradigms, in: Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2006, pp. 143–152. doi:10.1007/11839569_14.

[91] M. Rambold, H. Kasinger, F. Lautenbacher, B. Bauer, Towards Autonomic Service Discovery A Survey and Comparison, IEEE, 2009. doi:10.1109/scc.2009.59.

[92] S. Dustdar, C. Dorn, F. Li, L. Baresi, G. Cabri, C. Pautasso, F. Zambonelli, A roadmap towards sustainable self-aware service systems, in: Proceedings of the 2010 ICSE Workshop on Software Engineering for Adaptive and Self-Managing Systems - SEAMS '10, ACM Press, 2010, pp. 10–19. doi:10.1145/1808984.1808986.

[93] A. Berns, S. Ghosh, Dissecting self-* properties, in: 2009 Third IEEE International Conference on Self-Adaptive and Self-Organizing Systems, IEEE, IEEE, 2009, pp. 10–19. doi:10.1109/saso.2009.25.

[94] D. Weyns, S. Malek, J. Andersson, On decentralized self-adaptation, in: Proceedings of the 2010 ICSE Workshop on Software Engineering for Adaptive and Self-Managing Systems - SEAMS '10, ACM Press, 2010, p. 84–93. doi:10.1145/1808984.1808994.

[95] X. Vilajosana, J. Marques, A. Juan, D. Lazaro, R. Krishnaswamy, L. Navarro, A self-* auction server: design principles, architecture and implementation, International Journal of Autonomic Computing 1 (4) (2010) 374. doi:10.1504/ijac.2010.037513.

[96] F. Dotsch, J. Denzinger, H. Kasinger, B. Bauer, Decentralized Real-Time Control of Water Distribution Networks Using Self-Organizing Multi-agent Systems, IEEE, 2010. doi:10.1109/saso.2010.20.

[97] J. Camara, C. Canal, G. Salaun, Behavioural self-adaptation of services in ubiquitous computing environments, in: 2009 ICSE Workshop on Software Engineering for Adaptive and Self-Managing Systems, IEEE, 2009, pp. 28–37. doi:10.1109/seams.2009.5069071.

[98] L. Baresi, S. Guinea, G. Tamburrelli, Towards decentralized self-adaptive component-based systems, in: Proceedings of the 2008 international workshop on Software engineering for adaptive and self-managing systems - SEAMS '08, ACM Press, 2008, pp. 57–64. doi:10.1145/1370018.1370029.

[99] I. Dusparic, V. Cahill, Distributed w-learning: Multi-policy optimization in self-organizing systems, in: 2009 Third IEEE International Conference on Self-Adaptive and Self-Organizing Systems, IEEE, 2009, pp. 20–29. doi:10.1109/saso.2009.23.

[100] I. Dusparic, V. Cahill, Using reinforcement learning for multi-policy optimization in decentralized autonomic systems – an experimental evaluation, in: Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2009, pp. 105–119. doi:10.1007/978-3-642-02704-8_9.

[101] N. Honing, H. L. Poutre, Designing comprehensible self-organising systems, in: 2010 Fourth IEEE International Conference on Self-Adaptive and Self-Organizing Systems, IEEE, 2010, pp. 233–242. doi:10.1109/saso.2010.18.

[102] J. Zhang, B. H. C. Cheng, Model-based development of dynamically adaptive software, in: Proceedings of the 28th international conference on Software engineering, ACM, 2006, pp. 371–380. doi:10.1145/1134285.1134337.

[103] D. Sykes, W. Heaven, J. Magee, J. Kramer, Plan-directed architectural change for autonomous systems, in: Proceedings of the 2007 conference on Specification and verification of component-based systems 6th Joint Meeting of the European Conference on Software Engineering and the ACM SIGSOFT Symposium on the Foundations of Software Engineering - SAVCBS '07, ACM Press, 2007, pp. 15–21. doi:10.1145/1292316.1292318.

[104] R. Anthony, A policy-definition language and prototype implementation library for policy-based autonomic systems, in: 2006 IEEE International Conference on Autonomic Computing, IEEE, 2006, pp. 265–276. doi:10.1109/icac.2006.1662407.

[105] C. Gacek, H. Giese, E. Hadar, Friends or foes?, in: Proceedings of the 2008 international workshop on Software engineering for adaptive and self-managing systems - SEAMS '08, ACM Press, 2008, pp. 121–128. doi:10.1145/1370018.1370040.

[106] R. J. Anthony, Policy-based autonomic computing with integral support for self-stabilisation, International Journal of Autonomic Computing 1 (1) (2009) 1. doi:10.1504/ijac.2009.024497.

[107] A. Bernauer, O. Bringmann, W. Rosenstiel, Generic self-adaptation to reduce design effort for system-on-chip, in: 2009 Third IEEE International Conference on Self-Adaptive and Self-Organizing Systems, IEEE, 2009, pp. 126–135. doi:10.1109/saso.2009.41.

[108] Y. Brun, Improving impact of self-adaptation and self-management research through evaluation methodology, in: Proceedings of the 2010 ICSE Workshop on Software Engineering for Adaptive and Self-Managing Systems - SEAMS '10, ACM Press, 2010, pp. 1–9. `doi:10.1145/1808984.1808985`.

[109] B. Solomon, D. Ionescu, M. Litoiu, G. Iszlai, Autonomic computing control of composed web services, in: Proceedings of the 2010 ICSE Workshop on Software Engineering for Adaptive and Self-Managing Systems - SEAMS '10, ACM Press, 2010, pp. 94–103. `doi:10.1145/1808984.1808995`.

[110] G. Weiss, M. Zeller, D. Eilers, R. Knorr, Towards self-organization in automotive embedded systems, in: Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2009, pp. 32–46. `doi:10.1007/978-3-642-02704-8\_4`.

[111] S. Dolev, R. Yagel, Towards self-stabilizing operating systems, IEEE Transactions on Software Engineering 34 (4) (2008) 564–576. `doi:10.1109/tse.2008.46`.

[112] H. Hemmati, R. Jalili, Self-reconfiguration in highly available pervasive computing systems, in: Lecture Notes in Computer Science, Vol. 5060, Springer Berlin Heidelberg, 2008, pp. 289–301. `doi:10.1007/978-3-540-69295-9\_24`.

[113] D. Lorenzoli, L. Mariani, M. Pezze, Towards self-protecting enterprise applications, in: The 18th IEEE International Symposium on Software Reliability (ISSRE '07), IEEE, 2007, pp. 39–48. `doi:10.1109/issre.2007.21`.

[114] H. Seebach, F. Nafz, J. Holtmann, J. Meyer, M. Tichy, W. Reif, W. Schäfer, Designing self-healing in automotive systems, in: Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2010, pp. 47–61. `doi:10.1007/978-3-642-16576-4\_4`.

[115] J. Almeida, V. Almeida, D. Ardagna, C. Francalanci, M. Trubian, Resource management in the autonomic service-oriented architecture, in: 2006 IEEE International Conference on Autonomic Computing, IEEE, 2006, pp. 84–92. `doi:10.1109/icac.2006.1662385`.

[116] J. Andersson, R. de Lemos, S. Malek, D. Weyns, Reflecting on self-adaptive software systems, in: 2009 ICSE Workshop on Software Engineering for Adaptive and Self-Managing Systems, IEEE, 2009, pp. 38–47. doi:10.1109/seams.2009.5069072.

[117] J. B. van Veelen, SMDS (2006) 58–64doi:10.1145/1137677.1137689.

[118] H. Klus, A. Rausch, A general architecture for self-adaptive AmI components applied in speech recognition, in: Proceedings of the 2006 international workshop on Self-adaptation and self-managing systems - SEAMS '06, ACM Press, 2006, p. 72–78. doi:10.1145/1137677.1137692.

[119] S. Malek, N. Esfahani, D. A. Menasce, J. P. Sousa, H. Gomaa, Self-architecting software SYstems (SASSY) from QoS-annotated activity models, in: 2009 ICSE Workshop on Principles of Engineering Service Oriented Systems, IEEE, 2009, pp. 62–69. doi:10.1109/pesos.2009.5068821.

[120] C. Ballagny, N. Hameurlain, F. Barbier, MOCAS: A state-based component model for self-adaptation, in: 2009 Third IEEE International Conference on Self-Adaptive and Self-Organizing Systems, IEEE, 2009, pp. 206–215. doi:10.1109/saso.2009.11.

[121] S. RODRIGUEZ, V. HILAIRE, P. GRUER, A. KOUKAM, A FORMAL HOLONIC FRAMEWORK WITH PROVED SELF-ORGANIZING CAPABILITIES, International Journal of Cooperative Information Systems 16 (01) (2007) 7–25. doi:10.1142/s0218843007001548.

[122] L. Bauer, M. Shafique, D. Teufel, J. Henkel, A self-adaptive extensible embedded processor, in: First International Conference on Self-Adaptive and Self-Organizing Systems (SASO 2007), IEEE, 2007, pp. 344–350. doi:10.1109/saso.2007.2.

[123] T. Tidwell, R. Glaubius, C. Gill, W. D. Smart, Scheduling for reliable execution in autonomic systems, in: Lecture Notes in Computer Science, Springer, Springer Berlin Heidelberg, 2008, pp. 149–161. doi:10.1007/978-3-540-69295-9_14.

[124] G. Smith, J. W. Sanders, Formal development of self-organising systems, in: Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2009, pp. 90–104. doi:10.1007/978-3-642-02704-8_8.

[125] C. Barna, M. Litoiu, H. Ghanbari, Autonomic load-testing framework, in: Proceedings of the 8th ACM international conference on Autonomic computing - ICAC '11, ACM Press, 2011, p. 91–100. doi:10.1145/1998582.1998598.

[126] C. E. da Silva, R. de Lemos, Dynamic plans for integration testing of self-adaptive software systems, in: Proceeding of the 6th international symposium on Software engineering for adaptive and self-managing systems - SEAMS '11, ACM Press, 2011, pp. 148–157. doi:10.1145/1988008.1988029.

[127] J. Camara, R. de Lemos, Evaluation of resilience in self-adaptive systems using probabilistic model-checking, in: 2012 7th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS), IEEE, 2012, pp. 53–62. doi:10.1109/seams.2012.6224391.

[128] G. A. Moreno, J. Cámara, D. Garlan, B. Schmerl, Proactive self-adaptation under uncertainty: a probabilistic model checking approach, in: Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering, ACM, 2015, p. 1–12. doi:10.1145/2786805.2786853.

[129] J. Cámara, G. A. Moreno, D. Garlan, Stochastic game analysis and latency awareness for proactive self-adaptation, in: Proceedings of the 9th International Symposium on Software Engineering for Adaptive and Self-Managing Systems - SEAMS 2014, ACM Press, 2014, p. 155–164. doi:10.1145/2593929.2593933.

[130] M. Luckey, C. Thanos, C. Gerth, G. Engels, Multi-staged quality assurance for self-adaptive systems, in: 2012 IEEE Sixth International Conference on Self-Adaptive and Self-Organizing Systems Workshops, IEEE, 2012, pp. 111–118. doi:10.1109/sasow.2012.28.

[131] J. Cámara, R. de Lemos, M. Vieira, R. Almeida, R. Ventura, Architecture-based resilience evaluation for self-adaptive systems, Computing 95 (8) (2013) 689–722. doi:10.1007/s00607-013-0311-7.

[132] T. Patikirikorala, A. Colman, J. Han, L. Wang, A multi-model framework to implement self-managing control systems for QoS management, in: Proceeding of the 6th international symposium on Software engineering for adaptive and self-managing systems - SEAMS '11, ACM Press, 2011, pp. 218–227. doi:10.1145/1988008.1988040.

[133] G. Lulli, P. Potena, C. Raibulet, Resource allocation, trading and adaptation in self-managing systems, in: Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications, Vol. 83, Springer International Publishing, 2011, pp. 385–396. doi:10.1007/978-3-642-22056-2_41.

[134] G. Tamura, N. M. Villegas, H. A. Muller, L. Duchien, L. Seinturier, Improving context-awareness in self-adaptation using the DYNAMICO reference model, in: 2013 8th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS), IEEE, 2013, pp. 153–162. doi:10.1109/seams.2013.6595502.

[135] L. Rosa, L. Rodrigues, A. Lopes, M. Hiltunen, R. Schlichting, Self-management of adaptable component-based applications, IEEE Transactions on Software Engineering 39 (3) (2013) 403–421. doi:10.1109/tse.2012.29.

[136] J. Swanson, M. B. Cohen, M. B. Dwyer, B. J. Garvin, J. Firestone, Beyond the rainbow: self-adaptive failure avoidance in configurable systems, in: Proceedings of the 22nd ACM SIGSOFT International Symposium on Foundations of Software Engineering, ACM, 2014, p. 377–388. doi:10.1145/2635868.2635915.

[137] G. G. Pascual, M. Pinto, L. Fuentes, Self-adaptation of mobile systems driven by the common variability language, Future Generation Computer Systems 47 (2015) 127–144. doi:10.1016/j.future.2014.08.015.

[138] F. Alvares, E. Rutten, L. Seinturier, Behavioural model-based control for autonomic software components, in: 2015 IEEE International Conference on Autonomic Computing, IEEE, 2015, pp. 187–196. doi:10.1109/icac.2015.31.

[139] Z. Coker, D. Garlan, C. L. Goues, SASS: Self-adaptation using stochastic search, in: 2015 IEEE/ACM 10th International Symposium on Software Engineering for Adaptive and Self-Managing Systems, IEEE, 2015, pp. 168–174. doi:10.1109/seams.2015.16.

[140] G. Salvaneschi, C. Ghezzi, M. Pradella, ContextErlang: A language for distributed context-aware self-adaptive applications, Science of Computer Programming 102 (2015) 20–43. doi:10.1016/j.scico.2014.11.016.

[141] P. Casanova, D. Garlan, B. Schmerl, R. Abreu, Diagnosing unobserved components in self-adaptive systems, in: Proceedings of the 9th International Symposium on Software Engineering for Adaptive and Self-Managing Systems - SEAMS 2014, ACM Press, 2014, p. 75–84. doi:10.1145/2593929.2593946.

[142] B. Chen, X. Peng, Y. Yu, B. Nuseibeh, W. Zhao, Self-adaptation through incremental generative model transformations at runtime, in: Proceedings of the 36th International Conference on Software Engineering, ACM, 2014, p. 676–687. doi:10.1145/2568225.2568310.

[143] H. Nakagawa, A. Ohsuga, S. Honiden, gocc, in: Proceeding of the 6th international symposium on Software engineering for adaptive and self-managing systems - SEAMS '11, ACM Press, 2011, pp. 40–49. doi:10.1145/1988008.1988015.

[144] D. Weyns, S. Malek, J. Andersson, FORMS, ACM Transactions on Autonomous and Adaptive Systems. 7 (1) (2012) 1–61. doi:10.1145/2168260.2168268.

[145] E. M. Fredericks, B. DeVries, B. H. C. Cheng, Towards runtime adaptation of test cases for self-adaptive systems in the face of uncertainty, International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS) (2014). doi:10.1145/2593929.2593937.

[146] H. Nakagawa, A. Ohsuga, S. Honiden, Towards dynamic evolution of self-adaptive systems based on dynamic updating of control loops, in: 2012 IEEE Sixth International Conference on Self-Adaptive and Self-Organizing Systems, IEEE, 2012, pp. 59–68. doi:10.1109/saso.2012.17.

[147] M. U. Iftikhar, D. Weyns, ActivFORMS: active formal models for self-adaptation, in: Proceedings of the 9th International Symposium on Software Engineering for Adaptive and Self-Managing Systems - SEAMS 2014, ACM Press, 2014, p. 125–134. doi:10.1145/2593929.2593944.

[148] J. F. Inglés-Romero, C. Vicente-Chicote, Towards a formal approach for prototyping and verifying self-adaptive systems, in: Lecture Notes in Business Information Processing, Vol. 148, Springer Berlin Heidelberg, 2013, pp. 432–446. doi:10.1007/978-3-642-38490-5_39.

[149] W. Yang, C. Xu, Y. Liu, C. Cao, X. Ma, J. Lu, Verifying self-adaptive applications suffering uncertainty, in: Proceedings of the 29th ACM/IEEE international conference on Automated software engineering, ACM, 2014, p. 199–210. doi:10.1145/2642937.2642999.

[150] N. Rosemann, W. Brockmann, C. Lintze, Controlling the learning dynamics of interacting self-adapting systems, in: 2011 IEEE Fifth International Conference on Self-Adaptive and Self-Organizing Systems, IEEE, 2011, pp. 1–10. doi:10.1109/saso.2011.11.

[151] L. Baresi, L. Pasquale, P. Spoletini, Fuzzy goals for requirements-driven adaptation, in: 2010 18th IEEE International Requirements Engineering Conference, IEEE, 2010, pp. 125–134. doi:10.1109/re.2010.25.

[152] D. Sykes, D. Corapi, J. Magee, J. Kramer, A. Russo, K. Inoue, Learning revised models for planning in adaptive systems, in: 2013 35th International Conference on Software Engineering (ICSE), IEEE, 2013, pp. 63–71. doi:10.1109/icse.2013.6606552.

[153] S. Balasubramanian, R. Desmarais, H. A. Müller, U. Stege, S. Venkatesh, Characterizing problems for realizing policies in self-adaptive and self-managing systems, in: Proceeding of the 6th international symposium on Software engineering for adaptive and self-managing systems - SEAMS '11, ACM Press, 2011, pp. 70–79. doi:10.1145/1988008.1988019.

[154] L. Tomás, A. C. Caminero, C. Carrión, B. Caminero, Network-aware meta-scheduling in advance with autonomous self-tuning system, Future Generation Computer Systems 27 (5) (2011) 486–497. doi:10.1016/j.future.2010.12.004.

[155] K. Bartos, M. Rehak, Self-organized mechanism for distributed setup of multiple heterogeneous intrusion detection systems, in: 2012 IEEE Sixth International Conference on Self-Adaptive and Self-Organizing Systems Workshops, IEEE, 2012, pp. 31–38. doi:10.1109/sasow.2012.15.

[156] E. M. Fredericks, B. H. C. Cheng, Automated generation of adaptive test plans for self-adaptive systems, in: 2015 IEEE/ACM 10th International Symposium on Software Engineering for Adaptive and Self-Managing Systems, IEEE, 2015, pp. 157–167. doi:10.1109/seams.2015.15.

[157] E. M. Fredericks, B. H. Cheng, An empirical analysis of providing assurance for self-adaptive systems at different levels of abstraction in the face of uncertainty, in: 2015 IEEE/ACM 8th International Workshop on Search-Based Software Testing, IEEE, 2015, pp. 8–14. doi:10.1109/sbst.2015.9.

[158] A. Martin, T. Smaneoto, T. Dietze, A. Brito, C. Fetzer, User-constraint and self-adaptive fault tolerance for event stream processing systems, in: 2015 45th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, IEEE, 2015, pp. 462–473. doi:10.1109/dsn.2015.56.

[159] J. Camara, P. Correia, R. de Lemos, D. Garlan, P. Gomes, B. Schmerl, R. Ventura, Evolving an adaptive industrial software system to use architecture-based self-adaptation, in: 2013 8th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS), IEEE, 2013, pp. 13–22. doi:10.1109/seams.2013.6595488.

[160] B. Debbabi, A. Diaconescu, P. Lalanda, Controlling self-organising software applications with archetypes, in: 2012 IEEE Sixth International Conference on Self-Adaptive and Self-Organizing Systems, IEEE, 2012, pp. 69–78. doi:10.1109/saso.2012.21.

[161] S. Junuzovic, P. Dewan, Towards self-optimizing collaborative systems, in: Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work - CSCW '12, ACM Press, 2012, pp. 1421–1430. doi:10.1145/2145204.2145414.

[162] E. D. Nitto, D. J. Dubois, A. Margara, Reconfiguration primitives for self-adapting overlays in distributed publish-subscribe systems, in: 2012 IEEE Sixth International Conference on Self-Adaptive and Self-Organizing Systems, IEEE, 2012. doi:10.1109/saso.2012.27.

[163] Y. Maurel, S. Chollet, V. Lestideau, J. Bardin, P. Lalanda, A. Bottaro, fANFARE: Autonomic framework for service-based pervasive environment, in: 2012 IEEE Ninth International Conference on Services Computing, IEEE, 2012, pp. 65–72. doi:10.1109/scc.2012.7.

[164] P. Lalanda, C. Escoffier, C. Hamon, Cilia: An autonomic service bus for pervasive environments (2014) 488–495doi:10.1109/scc.2014.71.

[165] R. W. Moore, B. R. Childers, Inflation and deflation of self-adaptive applications, in: Proceeding of the 6th international symposium on Software engineering for adaptive and self-managing systems - SEAMS '11, ACM Press, 2011, pp. 228–237. doi:10.1145/1988008.1988041.

[166] E. Yuan, N. Esfahani, S. Malek, Automated mining of software component interactions for self-adaptation, in: Proceedings of the 9th International Symposium on Software Engineering for Adaptive and Self-Managing Systems - SEAMS 2014, ACM Press, 2014, p. 27–36. doi:10.1145/2593929.2593934.

[167] U. Adeel, S. Yang, J. A. McCann, Self-optimizing citizen-centric mobile urban sensing systems, in: International Conference on Autonomic Computing (ICAC), USENIX Association, 2014, pp. 161–167.

[168] J. Hudson, J. Denzinger, Risk management for self-adapting self-organizing emergent multi-agent systems performing dynamic task fulfillment, Autonomous Agents and Multi-Agent Systems 29 (5) (2014) 973–1022. doi:10.1007/s10458-014-9274-0.

[169] J. Hudson, J. Denzinger, H. Kasinger, B. Bauer, Dependable risk-aware efficiency improvement for self-organizing emergent systems, in: 2011 IEEE Fifth International Conference on Self-Adaptive and Self-Organizing Systems, IEEE, 2011, pp. 11–20. doi:10.1109/saso.2011.12.

[170] T. M. King, A. A. Allen, R. Cruz, P. J. Clarke, Safe runtime validation of behavioral adaptations in autonomic software, in: Lecture

Notes in Computer Science, Springer Berlin Heidelberg, 2011, pp. 31–46. doi:10.1007/978-3-642-23496-5_3.

[171] S. S. Andrade, R. J. de A. Macedo, A search-based approach for architectural design of feedback control concerns in self-adaptive systems, in: 2013 IEEE 7th International Conference on Self-Adaptive and Self-Organizing Systems, IEEE, 2013, pp. 61–70. doi:10.1109/saso.2013.42.

[172] P. Idziak, S. Clarke, An analysis of decision-making techniques in dynamic, self-adaptive systems, in: 2014 IEEE Eighth International Conference on Self-Adaptive and Self-Organizing Systems Workshops, IEEE, 2014, pp. 137–143. doi:10.1109/sasow.2014.23.

[173] N. Bencomo, A. Belaggoun, V. Issarny, Dynamic decision networks for decision-making in self-adaptive systems: A case study, in: 2013 8th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS), IEEE, IEEE, 2013, pp. 113–122. doi:10.1109/seams.2013.6595498.

[174] H. Gomaa, K. Hashimoto, Dynamic self-adaptation for distributed service-oriented transactions, in: 2012 7th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS), IEEE, 2012, pp. 11–20. doi:10.1109/seams.2012.6224386.

[175] N. R. Herbst, S. Kounev, A. Weber, H. Groenda, BUNGEE: An elasticity benchmark for self-adaptive IaaS cloud environments, in: 2015 IEEE/ACM 10th International Symposium on Software Engineering for Adaptive and Self-Managing Systems, IEEE, 2015, pp. 46–56. doi:10.1109/seams.2015.23.

[176] T. Patikirikorala, A. Colman, J. Han, L. Wang, A systematic survey on the design of self-adaptive software systems using control engineering approaches, in: 2012 7th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS), IEEE, 2012, pp. 33–42. doi:10.1109/seams.2012.6224389.

[177] N. Khakpour, S. Jalili, C. Talcott, M. Sirjani, M. Mousavi, Formal modeling of evolving self-adaptive systems, Science of Computer Programming 78 (1) (2012) 3–26. doi:10.1016/j.scico.2011.09.004.

[178] E. Vassev, M. Hinchey, The ASSL approach to specifying self-managing embedded systems, Concurrency and Computation: Practice and Experience 24 (16) (2011) 1860–1878. doi:10.1002/cpe.1758.

[179] S. van der Burg, E. Dolstra, A self-adaptive deployment framework for service-oriented systems, in: Proceeding of the 6th international symposium on Software engineering for adaptive and self-managing systems - SEAMS '11, ACM Press, 2011, pp. 208–217. doi:10.1145/1988008.1988039.

[180] Y. Maurel, P. Lalanda, A. Diaconescu, Towards a service-oriented component model for autonomic management, in: 2011 IEEE International Conference on Services Computing, IEEE, 2011, pp. 544–551. doi:10.1109/scc.2011.36.

[181] M. Viroli, On competitive self-composition in pervasive services, Science of Computer Programming 78 (5) (2013) 556–568. doi:10.1016/j.scico.2012.10.002.

[182] A. Filieri, C. Ghezzi, G. Tamburrelli, Run-time efficient probabilistic model checking, in: Proceedings of the 33rd International Conference on Software Engineering, ACM, 2011, p. 341–350. doi:10.1145/1985793.1985840.

[183] S.-W. Cheng, D. Garlan, Stitch: A language for architecture-based self-adaptation, Journal of Systems and Software 85 (12) (2012) 2860–2875. doi:10.1016/j.jss.2012.02.060.

[184] C. Ghezzi, L. S. Pinto, P. Spoletini, G. Tamburrelli, Managing non-functional uncertainty via model-driven adaptivity, in: 2013 35th International Conference on Software Engineering (ICSE), IEEE, 2013, pp. 33–42. doi:10.1109/icse.2013.6606549.

[185] J. Floch, C. Frà, R. Fricke, K. Geihs, M. Wagner, J. Lorenzo, E. Soladana, S. Mehlhase, N. Paspallis, H. Rahnama, P. Ruiz, U. Scholz, Playing MUSIC - building context-aware and self-adaptive mobile applications, Software: Practice and Experience 43 (3) (2012) 359–388. doi:10.1002/spe.2116.

[186] A. Filieri, C. Ghezzi, G. Tamburrelli, A formal approach to adaptive software: continuous assurance of non-functional requirements, Formal Aspects of Computing 24 (2) (2012) 163–186. doi:10.1007/s00165-011-0207-2.

[187] N. A. Qureshi, I. J. Jureta, A. Perini, Requirements engineering for self-adaptive systems: Core ontology and problem statement, in: Notes on Numerical Fluid Mechanics and Multidisciplinary Design, Springer International Publishing, 2011, pp. 33–47. doi:10.1007/978-3-642-21640-4_5.

[188] X. Peng, B. Chen, Y. Yu, W. Zhao, Self-tuning of software systems through dynamic quality tradeoff and value-based feedback control loop, Journal of Systems and Software 85 (12) (2012) 2707–2719. doi:10.1016/j.jss.2012.04.079.

[189] J. Bonnet, M.-P. Gleizes, E. Kaddoum, S. Rainjonneau, G. Flandin, Multi-satellite mission planning using a self-adaptive multi-agent system, in: 2015 IEEE 9th International Conference on Self-Adaptive and Self-Organizing Systems, IEEE, 2015, pp. 11–20. doi:10.1109/saso.2015.9.

[190] A. Campbell, C. Riggs, A. S. Wu, On the impact of variation on self-organizing systems, in: 2011 IEEE Fifth International Conference on Self-Adaptive and Self-Organizing Systems, IEEE, 2011, pp. 119–128. doi:10.1109/saso.2011.23.

[191] É. Piel, A. Gonzalez-Sanchez, H.-G. Gross, A. J. van Gemund, Spectrum-based health monitoring for self-adaptive systems, in: 2011 IEEE Fifth International Conference on Self-Adaptive and Self-Organizing Systems, IEEE, 2011, pp. 99–108. doi:10.1109/saso.2011.21.

[192] Y. Dai, Y. Xiang, Y. Li, L. Xing, G. Zhang, Consequence oriented self-healing and autonomous diagnosis for highly reliable systems and software, IEEE Transactions on Reliability 60 (2) (2011) 369–380. doi:10.1109/tr.2011.2136490.

[193] N. M. Villegas, H. A. Müller, G. Tamura, L. Duchien, R. Casallas, A framework for evaluating quality-driven self-adaptive software systems,

in: Proceeding of the 6th international symposium on Software engineering for adaptive and self-managing systems - SEAMS '11, ACM Press, 2011, pp. 80–89. doi:10.1145/1988008.1988020.

[194] N. Esfahani, E. Kouroshfar, S. Malek, Taming uncertainty in self-adaptive software, in: Proceedings of the 19th ACM SIGSOFT symposium and the 13th European conference on Foundations of software engineering - SIGSOFT/FSE '11, ACM Press, 2011, p. 234–244. doi:10.1145/2025113.2025147.

[195] S. Hassan, N. Bencomo, R. Bahsoon, Minimizing nasty surprises with better informed decision-making in self-adaptive systems, in: 2015 IEEE/ACM 10th International Symposium on Software Engineering for Adaptive and Self-Managing Systems, IEEE, 2015, pp. 134–145. doi:10.1109/seams.2015.13.

[196] T. Vogel, H. Giese, A language for feedback loops in self-adaptive systems: Executable runtime megamodels, in: 2012 7th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS), IEEE, 2012, pp. 129–138. doi:10.1109/seams.2012.6224399.

[197] T. Patikirikorala, A. Colman, J. Han, L. Wang, An evaluation of multi-model self-managing control schemes for adaptive performance management of software systems, Journal of Systems and Software 85 (12) (2012) 2678–2696. doi:10.1016/j.jss.2012.05.077.

[198] A.-E. Tchao, M. Risoldi, G. D. M. Serugendo, Modeling self-* systems using chemically-inspired composable patterns, in: 2011 IEEE Fifth International Conference on Self-Adaptive and Self-Organizing Systems, IEEE, 2011, pp. 109–118. doi:10.1109/saso.2011.22.

[199] P. Champrasert, J. Suzuki, C. Lee, Exploring self-optimization and self-stabilization properties in bio-inspired autonomic cloud applications, Concurrency and Computation: Practice and Experience 24 (9) (2012) 1015–1034. doi:10.1002/cpe.1906.

[200] A. Filieri, H. Hoffmann, M. Maggio, Automated multi-objective control for self-adaptive software design, in: Proceedings of the 2015 10th Joint

Meeting on Foundations of Software Engineering, ACM, 2015, p. 13–24. doi:10.1145/2786805.2786833.

[201] M. Salehie, L. Tahvildari, Towards a goal-driven approach to action selection in self-adaptive software, Software: Practice and Experience 42 (2) (2011) 211–233. doi:10.1002/spe.1066.

[202] M. Luckey, G. Engels, High-quality specification of self-adaptive software systems, in: 2013 8th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS), IEEE, 2013, pp. 143–152. doi:10.1109/seams.2013.6595501.

[203] C. Heinzemann, J. Rieke, W. Schafer, Simulating self-adaptive component-based systems using MATLAB/simulink, in: 2013 IEEE 7th International Conference on Self-Adaptive and Self-Organizing Systems, IEEE, 2013, pp. 71–80. doi:10.1109/saso.2013.17.

[204] P. Grosselfinger, J. Denzinger, B. Bauer, An advisor concept for distributed self-organizing systems acting in highly connected environments, in: International Conference on Self-Adaptive and Self-Organizing Systems (SASO), IEEE, 2013, pp. 121–130. doi:10.1109/saso.2013.15.

[205] D. Weyns, M. U. Iftikhar, J. Soderlund, Do external feedback loops improve the design of self-adaptive systems? a controlled experiment, in: 2013 8th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS), IEEE, 2013, pp. 3–12. doi:10.1109/seams.2013.6595487.

[206] T. Weis, A. Wacker, Self-stabilizing embedded systems, in: Proceedings of the 2011 workshop on Organic computing - OC '11, ACM Press, 2011, pp. 59–66. doi:10.1145/1998642.1998653.

[207] A. Filieri, H. Hoffmann, M. Maggio, Automated design of self-adaptive software with control-theoretical formal guarantees, in: Proceedings of the 36th International Conference on Software Engineering, ACM, 2014, p. 299–310. doi:10.1145/2568225.2568272.

[208] W.-C. Huang, W. J. Knottenbelt, Self-adaptive containers: Building resource-efficient applications with low programmer overhead, in:

2013 8th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS), IEEE, 2013, pp. 123–132. doi:10.1109/seams.2013.6595499.

[209] J. Camara, G. A. Moreno, D. Garlan, Reasoning about human participation in self-adaptive systems, in: International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS), Defense Technical Information Center, 2015, pp. 146–156. doi:10.21236/ada614218.

[210] M. U. Iftikhar, G. S. Ramachandran, P. Bollansee, D. Weyns, D. Hughes, DeltaIoT: A self-adaptive internet of things exemplar, in: 2017 IEEE/ACM 12th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS), IEEE, 2017, pp. 76–82. doi:10.1109/seams.2017.21.

[211] M. Provoost, D. Weyns, DingNet: A self-adaptive internet-of-things exemplar, in: 2019 IEEE/ACM 14th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS), IEEE, 2019, pp. 195–201. doi:10.1109/seams.2019.00033.

[212] E. Zavala, X. Franch, J. Marco, C. Berger, HAFLoop: An architecture for supporting highly adaptive feedback loops in self-adaptive systems, Future Generation Computer Systems 105 (2020) 607–630. doi:10.1016/j.future.2019.12.026.

[213] E. Pournaras, M. Yao, D. Helbing, Self-regulating supply–demand systems, Future Generation Computer Systems 76 (2017) 73–91. doi:10.1016/j.future.2017.05.018.

[214] M. WeiBbach, P. Chrszon, T. Springer, A. Schill, Decentrally coordinated execution of adaptations in distributed self-adaptive software systems, in: 2017 IEEE 11th International Conference on Self-Adaptive and Self-Organizing Systems (SASO), IEEE, 2017, pp. 111–120. doi:10.1109/saso.2017.20.

[215] L. Birdsey, C. Szabo, K. Falkner, Identifying self-organization and adaptability in complex adaptive systems, in: 2017 IEEE 11th International Conference on Self-Adaptive and Self-Organizing Systems (SASO), IEEE, 2017, pp. 131–140. doi:10.1109/saso.2017.22.

[216] D. Arellanes, K.-K. Lau, Workflow variability for autonomic IoT systems, in: 2019 IEEE International Conference on Autonomic Computing (ICAC), IEEE, 2019, pp. 24–30. doi:10.1109/icac.2019.00014.

[217] W. Jiao, Y. Sun, Self-adaptation of multi-agent systems in dynamic environments based on experience exchanges, Journal of Systems and Software 122 (2016) 165–179. doi:10.1016/j.jss.2016.09.025.

[218] J. Beal, M. Viroli, D. Pianini, F. Damiani, Self-adaptation to device distribution in the internet of things, ACM Transactions on Autonomous and Adaptive Systems 12 (3) (2017) 1–29. doi:10.1145/3105758.

[219] D. Weyns, M. U. Iftikhar, Model-based simulation at runtime for self-adaptive systems, in: 2016 IEEE International Conference on Autonomic Computing (ICAC), IEEE, 2016, pp. 364–373. doi:10.1109/icac.2016.67.

[220] F. Oquendo, Software architecture of self-organizing systems-of-systems for the internet-of-things with SosADL, in: 2017 12th System of Systems Engineering Conference (SoSE), IEEE, 2017, pp. 1–6. doi:10.1109/sysose.2017.7994959.

[221] A. Colin, E. Gerbert-Gaillard, G. Vega, P. Lalanda, S. Chollet, Autonomic service-oriented context for pervasive applications, in: 2016 IEEE International Conference on Services Computing (SCC), IEEE, 2016, pp. 491–498. doi:10.1109/scc.2016.70.

[222] J. Camara, R. de Lemos, N. Laranjeiro, R. Ventura, M. Vieira, Robustness-driven resilience evaluation of self-adaptive software systems, IEEE Transactions on Dependable and Secure Computing 14 (1) (2015) 1–1. doi:10.1109/tdsc.2015.2429128.

[223] C. Raibulet, F. A. Fontana, Evaluation of self-adaptive systems, in: Proceedings of the 11th European Conference on Software Architecture: Companion Proceedings, ACM, 2017, pp. 23–30. doi:10.1145/3129790.3129825.

[224] Z. Ding, Y. Zhou, M. Zhou, Modeling self-adaptive software systems by fuzzy rules and petri nets, IEEE Transactions on Fuzzy Systems 26 (2) (2018) 967–984. doi:10.1109/tfuzz.2017.2700286.

[225] F. Quin, D. Weyns, T. Bamelis, S. S. Buttar, S. Michiels, Efficient analysis of large adaptation spaces in self-adaptive systems using machine learning, in: 2019 IEEE/ACM 14th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS), IEEE, 2019, pp. 1–12. doi:10.1109/seams.2019.00011.

[226] S. Spinner, J. Grohmann, S. Eismann, S. Kounev, Online model learning for self-aware computing infrastructures, Journal of Systems and Software 147 (2019) 1–16. doi:10.1016/j.jss.2018.09.089.

[227] J. Cámara, A. Lopes, D. Garlan, B. Schmerl, Adaptation impact and environment models for architecture-based self-adaptive systems, Science of Computer Programming 127 (2016) 50–75. doi:10.1016/j.scico.2015.12.006.

[228] J. Beal, M. Viroli, D. Pianini, F. Damiani, Self-adaptation to device distribution changes, in: 2016 IEEE 10th International Conference on Self-Adaptive and Self-Organizing Systems (SASO), IEEE, 2016, pp. 60–69. doi:10.1109/saso.2016.12.

[229] G. F. Solano, R. D. Caldas, G. N. Rodrigues, T. Vogel, P. Pelliccione, Taming uncertainty in the assurance process of self-adaptive systems: a goal-oriented approach, in: 2019 IEEE/ACM 14th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS), IEEE, 2019, p. 89–99. doi:10.1109/seams.2019.00020.

[230] F. Burzlaff, C. Bartelt, A conceptual architecture for enabling future self-adaptive service systems, in: Proceedings of the Annual Hawaii International Conference on System Sciences, Hawaii International Conference on System Sciences, 2019, pp. 1–10. doi:10.24251/hicss.2019.899.

[231] K. L. Bellman, C. Gruhl, C. Landauer, S. Tomforde, Self-improving system integration - on a definition and characteristics of the challenge, in: 2019 IEEE 4th International Workshops on Foundations and Applications of Self* Systems (FAS*W), IEEE, 2019, pp. 1–3. doi:10.1109/fas-w.2019.00014.

[232] H. Li, T.-H. P. Chen, A. E. Hassan, M. Nasser, P. Flora, Adopting autonomic computing capabilities in existing large-scale systems, in: Proceedings of the 40th International Conference on Software Engineering: Software Engineering in Practice, ACM, 2018, pp. 1–10. `doi:10.1145/3183519.3183544`.

[233] M. Staron, W. Meding, M. Tichy, J. Bjurhede, H. Giese, O. Söder, Industrial experiences from evolving measurement systems into self-healing systems for improved availability, Software: Practice and Experience 48 (3) (2017) 719–739. `doi:10.1002/spe.2522`.

[234] C.-H. Lung, X. Zhang, P. Rajeswaran, Improving software performance and reliability in a distributed and concurrent environment with an architecture-based self-adaptive framework, Journal of Systems and Software 121 (2016) 311–328. `doi:10.1016/j.jss.2016.06.102`.

[235] P. C. Vinh, Concurrency of self-* in autonomic systems, Future Generation Computer Systems 56 (2016) 140–152. `doi:10.1016/j.future.2015.04.017`.

[236] A. Diaconescu, B. Porter, R. Rodrigues, E. Pournaras, Hierarchical self-awareness and authority for scalable self-integrating systems, in: 2018 IEEE 3rd International Workshops on Foundations and Applications of Self* Systems (FAS*W), IEEE, 2018, pp. 168–175. `doi:10.1109/fas-w.2018.00043`.

[237] V. Podolskiy, A. Jindal, M. Gerndt, Y. Oleynik, Forecasting models for self-adaptive cloud applications: A comparative study, in: 2018 IEEE 12th International Conference on Self-Adaptive and Self-Organizing Systems (SASO), IEEE, 2018, pp. 40–49. `doi:10.1109/saso.2018.00015`.

[238] J. M. Franco, F. Correia, R. Barbosa, M. Zenha-Rela, B. Schmerl, D. Garlan, Improving self-adaptation planning through software architecture-based stochastic modeling, Journal of Systems and Software 115 (2016) 42–60. `doi:10.1016/j.jss.2016.01.026`.

[239] A. Pandey, G. A. Moreno, J. Camara, D. Garlan, Hybrid planning for decision making in self-adaptive systems, in: 2016 IEEE 10th International Conference on Self-Adaptive and Self-Organizing Systems (SASO), IEEE, 2016, pp. 130–139. `doi:10.1109/saso.2016.19`.

[240] C. Kinneer, Z. Coker, J. Wang, D. Garlan, C. L. Goues, Managing uncertainty in self-adaptive systems with plan reuse and stochastic search, in: Proceedings of the 13th International Conference on Software Engineering for Adaptive and Self-Managing Systems, ACM, 2018, pp. 40–50. doi:10.1145/3194133.3194145.

[241] F. Messina, G. Pappalardo, D. Rosaci, C. Santoro, G. Sarné, A trust-aware, self-organizing system for large-scale federations of utility computing infrastructures, Future Generation Computer Systems 56 (2016) 77–94. doi:10.1016/j.future.2015.07.013.

[242] C. Krupitzer, F. M. Roth, M. Pfannemuller, C. Becker, Comparison of approaches for self-improvement in self-adaptive systems, in: 2016 IEEE International Conference on Autonomic Computing (ICAC), IEEE, 2016, pp. 308–314. doi:10.1109/icac.2016.18.

[243] D. M. Barbosa, R. G. de Moura Lima, P. H. M. Maia, E. Costa, Lotus@runtime: A tool for runtime monitoring and verification of self-adaptive systems, in: 2017 IEEE/ACM 12th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS), IEEE, 2017, pp. 24–30. doi:10.1109/seams.2017.18.

[244] T. Chen, R. Bahsoon, Self-adaptive and online QoS modeling for cloud-based software services, IEEE Transactions on Software Engineering 43 (5) (2017) 453–475. doi:10.1109/tse.2016.2608826.

[245] M. Salama, R. Bahsoon, Analysing and modelling runtime architectural stability for self-adaptive software, Journal of Systems and Software 133 (2017) 95–112. doi:10.1016/j.jss.2017.07.041.

[246] V. Podolskiy, M. Mayo, A. Koay, M. Gerndt, P. Patros, Maintaining SLOs of cloud-native applications via self-adaptive resource sharing, in: 2019 IEEE 13th International Conference on Self-Adaptive and Self-Organizing Systems (SASO), IEEE, 2019, pp. 72–81. doi:10.1109/saso.2019.00018.

[247] G. A. Moreno, B. Schmerl, D. Garlan, SWIM, in: Proceedings of the 13th International Conference on Software Engineering for Adaptive and Self-Managing Systems, ACM, 2018, pp. 137–143. doi:10.1145/3194133.3194163.

[248] S. Bosse, D. Lehmhus, Towards large-scale material-integrated computing: Self-adaptive materials and agents, in: 2017 IEEE 2nd International Workshops on Foundations and Applications of Self* Systems (FAS*W), IEEE, 2017, pp. 70–78. doi:10.1109/fas-w.2017.123.

[249] J. C. Moreno, A. Lopes, D. Garlan, B. Schmerl, Impact models for architecture-based self-adaptive systems, in: Formal Aspects of Component Software, Vol. 8997, Springer International Publishing, 2015, pp. 89–107. doi:10.1007/978-3-319-15317-9_6.

[250] J. Palmerino, Q. Yu, T. Desell, D. Krutz, Improving the decision-making process of self-adaptive systems by accounting for tactic volatility, in: 2019 34th IEEE/ACM International Conference on Automated Software Engineering (ASE), IEEE, 2019, pp. 949–961. doi:10.1109/ase.2019.00092.

[251] T. Vogel, H. Giese, Adaptation and abstract runtime models, in: Proceedings of the 2010 ICSE Workshop on Software Engineering for Adaptive and Self-Managing Systems - SEAMS '10, ACM Press, 2010, pp. 39–48. doi:10.1145/1808984.1808989.

[252] F. A. Moghaddam, R. Deckers, G. Procaccianti, P. Grosso, P. Lago, A domain model for self-adaptive software systems, in: Proceedings of the 11th European Conference on Software Architecture: Companion Proceedings, ACM, 2017, pp. 16–22. doi:10.1145/3129790.3129824.

[253] K. Colson, R. Dupuis, L. Montrieux, Z. Hu, S. Uchitel, P.-Y. Schobbens, Reusable self-adaptation through bidirectional programming, in: Proceedings of the 11th International Symposium on Software Engineering for Adaptive and Self-Managing Systems, ACM, 2016, pp. 4–15. doi:10.1145/2897053.2897055.

[254] G. A. Moreno, J. Camara, D. Garlan, B. Schmerl, Efficient decision-making under uncertainty for proactive self-adaptation, in: 2016 IEEE International Conference on Autonomic Computing (ICAC), IEEE, 2016, pp. 147–156. doi:10.1109/icac.2016.59.

[255] H. Muccini, M. Sharaf, D. Weyns, Self-adaptation for cyber-physical systems, in: Proceedings of the 11th International Symposium on

Software Engineering for Adaptive and Self-Managing Systems, ACM, 2016, pp. 75–81. doi:10.1145/2897053.2897069.

[256] A. Borda, L. Pasquale, V. Koutavas, B. Nuseibeh, Compositional verification of self-adaptive cyber-physical systems, in: Proceedings of the 13th International Conference on Software Engineering for Adaptive and Self-Managing Systems, ACM, 2018, pp. 1–11. doi:10.1145/3194133.3194146.

[257] G. Moreno, C. Kinneer, A. Pandey, D. Garlan, DARTSim: An exemplar for evaluation and comparison of self-adaptation approaches for smart cyber-physical systems, in: 2019 IEEE/ACM 14th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS), IEEE, 2019, pp. 181–187. doi:10.1109/seams.2019.00031.

[258] I. Gerostathopoulos, D. Skoda, F. Plasil, T. Bures, A. Knauss, Tuning self-adaptation in cyber-physical systems through architectural homeostasis, Journal of Systems and Software 148 (2019) 37–55. doi:10.1016/j.jss.2018.10.051.

[259] G. Weiss, P. Schleiss, D. Schneider, M. Trapp, Towards integrating undependable self-adaptive systems in safety-critical environments, in: Proceedings of the 13th International Conference on Software Engineering for Adaptive and Self-Managing Systems, ACM, 2018, pp. 26–32. doi:10.1145/3194133.3194157.

[260] I. Gerostathopoulos, T. Bures, P. Hnetynka, J. Keznikl, M. Kit, F. Plasil, N. Plouzeau, Self-adaptation in software-intensive cyber–physical systems: From system goals to architecture configurations, Journal of Systems and Software 122 (2016) 378–397. doi:10.1016/j.jss.2016.02.028.

[261] F. Rammig, K. Stahl, Online behavior classification for anomaly detection in self-x real-time systems, Concurrency and Computation: Practice and Experience 28 (14) (2014) 3751–3772. doi:10.1109/isorc.2014.24.

[262] U. Brinkschulte, An artificial DNA for self-descripting and self-building embedded real-time systems, Concurrency and Computation: Practice and Experience 28 (14) (2015) 3711–3729. doi:10.1002/cpe.3460.

[263] A. Marshall, S. Jahan, R. Gamble, Toward evaluating the impact of self-adaptation on security control certification, in: Proceedings of the 13th International Conference on Software Engineering for Adaptive and Self-Managing Systems, ACM, 2018, pp. 149–160. doi:10.1145/3194133.3194139.

[264] R. Calinescu, D. Weyns, S. Gerasimou, M. U. Iftikhar, I. Habli, T. Kelly, Engineering trustworthy self-adaptive software with dynamic assurance cases, IEEE Transactions on Software Engineering 44 (11) (2018) 1039–1069. doi:10.1109/tse.2017.2738640.

[265] Z. Yan, R. MacLaverty, Autonomic trust management in a component based software system, in: Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2006, pp. 279–292. doi:10.1007/11839569_27.

[266] Y. Wang, F. Xu, Y. Tao, C. Cao, J. Lü, Toward trust management in autonomic and coordination applications, in: Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2006, pp. 236–245. doi:10.1007/11839569_23.

[267] M. Serrano, S. van der Meer, J. Strassner, S. D. Paoli, A. Kerr, C. Storni, Trust and reputation policy-based mechanisms for self-protection in autonomic communications, in: Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2009, pp. 249–267. doi:10.1007/978-3-642-02704-8_19.

[268] S. S. Yau, Y. Yao, M. Yan, Development and runtime support for situation-aware security in autonomic computing, in: Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2006, pp. 173–182. doi:10.1007/11839569_17.

[269] S. Wang, J. Wan, D. Zhang, D. Li, C. Zhang, Towards smart factory for industry 4.0: a self-organized multi-agent system with big data based feedback and coordination, Computer Networks 101 (2016) 158–168. doi:10.1016/j.comnet.2015.12.017.

[270] T. Vogel, mrubis: An exemplar for model-based architectural self-healing and self-optimization (artifact), Dagstuhl Artifacts Series 4 (2018). doi:10.4230/DARTS.4.1.1.

[271] W. Yang, C. Xu, M. Pan, C. Cao, X. Ma, J. Lu, Efficient validation of self-adaptive applications by counterexample probability maximization, Journal of Systems and Software 138 (2018) 82–99. doi:10.1016/j.jss.2017.12.009.

[272] A. Reichstaller, A. Knapp, Risk-based testing of self-adaptive systems using run-time predictions, in: 2018 IEEE 12th International Conference on Self-Adaptive and Self-Organizing Systems (SASO), IEEE, 2018, pp. 80–89. doi:10.1109/saso.2018.00019.

[273] G. A. Moreno, A. V. Papadopoulos, K. Angelopoulos, J. Camara, B. Schmerl, Comparing model-based predictive approaches to self-adaptation: CobRA and PLA, in: 2017 IEEE/ACM 12th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS), IEEE, 2017, pp. 42–53. doi:10.1109/seams.2017.2.

[274] A. Pandey, I. Ruchkin, B. Schmerl, J. Camara, Towards a formal framework for hybrid planning in self-adaptation, in: 2017 IEEE/ACM 12th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS), IEEE, 2017, pp. 109–115. doi:10.1109/seams.2017.14.

[275] E. M. Fredericks, An empirical analysis of the mutation operator for run-time adaptive testing in self-adaptive systems (2018) 59–66doi:10.1145/3194718.3194726.

[276] R. Bloem, N. Braud-Santoni, S. Jacobs, Synthesis of self-stabilising and byzantine-resilient distributed systems, in: Computer Aided Verification, Springer International Publishing, 2016, pp. 157–176. doi:10.1007/978-3-319-41528-4_9.

[277] G. A. Moreno, J. Cámara, D. Garlan, M. Klein, Uncertainty reduction in self-adaptive systems, in: Proceedings of the 13th International Conference on Software Engineering for Adaptive and Self-Managing Systems, ACM, 2018, pp. 51–57. doi:10.1145/3194133.3194144.

[278] E. M. Fredericks, Automatically hardening a self-adaptive system against uncertainty, in: Proceedings of the 11th International Symposium on Software Engineering for Adaptive and Self-Managing Systems, ACM, 2016, pp. 16–27. doi:10.1145/2897053.2897059.

[279] E. Incerto, M. Tribastone, C. Trubiani, Software performance self-adaptation through efficient model predictive control, in: 2017 32nd IEEE/ACM International Conference on Automated Software Engineering (ASE), IEEE, 2017, pp. 485–496. `doi:10.1109/ase.2017.8115660`.

[280] N. Z. Naqvi, J. Devlieghere, D. Preuveneers, Y. Berbers, MAsCOT: Self-adaptive opportunistic offloading for cloud-enabled smart mobile applications with probabilistic graphical models at runtime, in: 2016 49th Hawaii International Conference on System Sciences (HICSS), IEEE, 2016, pp. 5701–5710. `doi:10.1109/hicss.2016.705`.

[281] Y. Qin, C. Xu, P. Yu, J. Lu, SIT: Sampling-based interactive testing for self-adaptive apps, Journal of Systems and Software 120 (2016) 70–88. `doi:10.1016/j.jss.2016.07.002`.

[282] N. Condori-Fernandez, F. S. Lopez, Using emotions to empower the self-adaptation capability of software services, in: 2017 IEEE/ACM 2nd International Workshop on Emotion Awareness in Software Engineering (SEmotion), IEEE, 2017, pp. 15–21. `doi:10.1109/semotion.2017.8`.

[283] E. M. Grua, I. Malavolta, P. Lago, Self-adaptation in mobile apps: a systematic literature study, in: 2019 IEEE/ACM 14th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS), IEEE, 2019, pp. 51–62. `doi:10.1109/seams.2019.00016`.

[284] S. Taherizadeh, A. C. Jones, I. Taylor, Z. Zhao, V. Stankovski, Monitoring self-adaptive applications within edge computing frameworks: A state-of-the-art review, Journal of Systems and Software 136 (2018) 19–38. `doi:10.1016/j.jss.2017.10.033`.

[285] M. D'Angelo, S. Gerasimou, S. Ghahremani, J. Grohmann, I. Nunes, E. Pournaras, S. Tomforde, On learning in collective self-adaptive systems: State of practice and a 3d framework, in: 2019 IEEE/ACM 14th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS), IEEE, 2019, pp. 13–24. `doi:10.1109/seams.2019.00012`.

[286] J. Boes, F. Migeon, Self-organizing multi-agent systems for the control of complex systems, Journal of Systems and Software 134 (2017) 12–28. doi:10.1016/j.jss.2017.08.038.

[287] G. A. Moreno, O. Strichman, S. Chaki, R. Vaisman, Decision-making with cross-entropy for self-adaptation, in: 2017 IEEE/ACM 12th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS), IEEE, 2017, pp. 90–101. doi:10.1109/seams.2017.7.

[288] M. Camilli, A. Gargantini, P. Scandurra, Zone-based formal specification and timing analysis of real-time self-adaptive systems, Science of Computer Programming 159 (2018) 28–57. doi:10.1016/j.scico.2018.03.002.

[289] B. DeVries, B. H. C. Cheng, Run-time monitoring of self-adaptive systems to detect n-way feature interactions and their causes, in: Proceedings of the 13th International Conference on Software Engineering for Adaptive and Self-Managing Systems, ACM, 2018, pp. 94–100. doi:10.1145/3194133.3194141.

[290] D. King, G. Peterson, A macro-level order metric for self-organizing adaptive systems, in: 2018 IEEE 12th International Conference on Self-Adaptive and Self-Organizing Systems (SASO), IEEE, 2018, pp. 60–69. doi:10.1109/saso.2018.00017.

[291] S. Pradhan, A. Dubey, T. Levendovszky, P. S. Kumar, W. A. Emfinger, D. Balasubramanian, W. Otte, G. Karsai, Achieving resilience in distributed software systems via self-reconfiguration, Journal of Systems and Software 122 (2016) 344–363. doi:10.1016/j.jss.2016.05.038.

[292] S. Shevtsov, D. Weyns, Keep it SIMPLEX: satisfying multiple goals with guarantees in control-based self-adaptive systems, in: Proceedings of the 2016 24th ACM SIGSOFT International Symposium on Foundations of Software Engineering, ACM, 2016, pp. 229–241. doi:10.1145/2950290.2950301.

[293] S. Shevtsov, D. Weyns, M. Maggio, Handling new and changing requirements with guarantees in self-adaptive systems using SimCA, in:

2017 IEEE/ACM 12th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS), IEEE, 2017, pp. 12–23. doi:10.1109/seams.2017.3.

[294] A. Rodrigues, R. D. Caldas, G. N. Rodrigues, T. Vogel, P. Pelliccione, A learning approach to enhance assurances for real-time self-adaptive systems, in: Proceedings of the 13th International Conference on Software Engineering for Adaptive and Self-Managing Systems, ACM, 2018, p. 206–216. doi:10.1145/3194133.3194147.

[295] V. Nallur, S. Clarke, Clonal plasticity: an autonomic mechanism for multi-agent systems to self-diversify, Autonomous Agents and Multi-Agent Systems 32 (2) (2017) 275–311. doi:10.1007/s10458-017-9380-x.

[296] S. Jahan, A. Marshall, R. Gamble, Embedding verification concerns in self-adaptive system code, in: 2017 IEEE 11th International Conference on Self-Adaptive and Self-Organizing Systems (SASO), IEEE, 2017, pp. 121–130. doi:10.1109/saso.2017.21.

[297] J. Boubin, J. Chumley, C. Stewart, S. Khanal, Autonomic computing challenges in fully autonomous precision agriculture, in: 2019 IEEE International Conference on Autonomic Computing (ICAC), IEEE, 2019, pp. 11–17. doi:10.1109/icac.2019.00012.

[298] J. P. S. da Silva, M. Ecar, M. S. Pimenta, G. T. A. Guedes, L. P. Franz, L. Marchezan, A systematic literature review of UML-based domain-specific modeling languages for self-adaptive systems, in: Proceedings of the 13th International Conference on Software Engineering for Adaptive and Self-Managing Systems, ACM, 2018, pp. 87–93. doi:10.1145/3194133.3194136.

[299] M. S. Reorda, L. Sterpone, A. Ullah, An error-detection and self-repairing method for dynamically and partially reconfigurable systems, IEEE Transactions on Computers 66 (6) (2017) 1022–1033. doi:10.1109/tc.2016.2607749.

[300] S. Ghahremani, H. Giese, T. Vogel, Efficient utility-driven self-healing employing adaptation rules for large dynamic architectures, International Conference on Autonomic Computing (ICAC) (2017) 59–68doi:10.1109/icac.2017.35.

[301] C. Krupitzer, M. Pfannemuller, J. Kaddour, C. Becker, SATISFy: Towards a self-learning analyzer for time series forecasting in self-improving systems, in: 2018 IEEE 3rd International Workshops on Foundations and Applications of Self* Systems (FAS*W), IEEE, 2018, pp. 182–189. doi:10.1109/fas-w.2018.00045.

[302] A. Diaconescu, S. Frey, C. Muller-Schloer, J. Pitt, S. Tomforde, Goal-oriented holonics for complex system (self-)integration: Concepts and case studies, in: 2016 IEEE 10th International Conference on Self-Adaptive and Self-Organizing Systems (SASO), IEEE, 2016, pp. 100–109. doi:10.1109/saso.2016.16.

[303] F. Alvares, G. Delaval, E. Rutten, L. Seinturier, Language support for modular autonomic managers in reconfigurable software components, in: 2017 IEEE International Conference on Autonomic Computing (ICAC), IEEE, 2017, pp. 271–278. doi:10.1109/icac.2017.48.

[304] I. Gerostathopoulos, T. Vogel, D. Weyns, P. Lago, How do we evaluate self-adaptive software systems?: A ten-year perspective of seams, in: 2021 International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS), IEEE, 2021, pp. 59–70. doi:10.1109/SEAMS51251.2021.00018.

[305] P. Ralph, N. b. Ali, S. Baltes, D. Bianculli, J. Diaz, Y. Dittrich, N. Ernst, M. Felderer, R. Feldt, A. Filieri, et al., Empirical standards for software engineering research, arXiv preprint arXiv:2010.03525 (2020). doi:10.48550/arXiv.2010.03525.