Self-X Characterization of Autonomous Systems: A Systematic Literature Review

INGA MIADOWICZ*, Deutsches Zentrum für Luft- und Raumfahrt (DLR), Germany
DANIEL MALDONADO QUINTO, Deutsches Zentrum für Luft- und Raumfahrt (DLR), Germany
MICHAEL FELDERER, Deutsches Zentrum für Luft- und Raumfahrt (DLR), Germany

Autonomous systems are being discussed increasingly as a next-level system paradigm after automated systems. Although the vision of autonomous systems seems to be on the verge of realization today, there is still a lack of standards, and consequently a missing common understanding on what exactly constitutes an autonomous system in the current research state. This paper contributes to the clarification of system autonomy based on a systematic literature review. Thereby, the characteristics of autonomous systems and the relation to comparable system classes are examined and a conceptual model for the self-x characterization of autonomous systems is introduced.

CCS Concepts: • **General and reference → Surveys and overviews**; • **Theory of computation**; • **Information systems**;

Additional Key Words and Phrases: Autonomous System, Autonomic Computing, Automated System, Adaptive System, Self-X Capabilities

ACM Reference Format:
Inga Miadowicz, Daniel Maldonado Quinto, and Michael Felderer. 2024. Self-X Characterization of Autonomous Systems: A Systematic Literature Review. 1, 1 (June 2024), 34 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

The idea of autonomous systems has been discussed in society, science and industry for centuries [183]. Based on technological advances in recent years [183] and progress in fields such as autonomous cars [47], the vision of autonomous systems already seems to be reality today or is at least on the verge of becoming reality. Although autonomous systems are increasingly being discussed [118], there is still a lack of standards and a missing common understanding of the concepts of system autonomy in various application domains [77].

In research, many authors use self-x capabilities to characterize autonomous systems [8] such as self-management, self-organization, or self-protection. These characteristics describe the main abilities that make a system autonomous. Therefore, they are often used to define the vision of autonomy for a certain scenario and to decide whether a system is autonomous or not [165]. As the concept of self-x capabilities is not standardized as well, the literature proposes a large number of self-x properties today. The result is that authors across all domains use different sets of self-x requirements to define the vision and characteristics of autonomy as a baseline for future work. Even if they describe similar system capabilities and goals, they often have a slightly different skill set, use another wording for the same characteristics, or classify the capabilities at conflicting levels as minor or major traits. This makes the self-x capabilities of autonomous

Authors' Contact Information: Inga Miadowicz, Inga.Miadowicz@dlr.de, Deutsches Zentrum für Luft- und Raumfahrt (DLR), Cologne, North Rhine-Westphalia, Germany; Daniel Maldonado Quinto, Deutsches Zentrum für Luft- und Raumfahrt (DLR), Cologne, Germany; Michael Felderer, Michael.Felderer@dlr.de, Deutsches Zentrum für Luft- und Raumfahrt (DLR), Cologne, Germany.

systems difficult to compare between scenarios, authors, and domains. Consequently, it is still challenging to choose a set of self-x properties that characterize system autonomy and help to decide whether a system is autonomous or not.

What intensifies the fuzzy concept of autonomous systems and their self-x capabilities is the description of similar ideas by different system classes. While some authors write about autonomous systems [40, 41, 118], others describe comparable concepts as automated [55, 100, 135], autonomic [39, 52, 109, 168, 192], intelligent [34, 66, 148, 176], adaptive [21, 68, 77, 189], organic [70, 157, 163, 190] or use system class names interchangeably [108, 110, 165, 192]. Analogously to the characterization of autonomous systems, self-x capabilities are also used to describe related system classes. Although these system classes have different names, their self-x capabilities can be identical, similar but with a different wording, partially overlapping or completely distinct from others depending on the authors and domain. This leads to a whole bunch of related system concepts with similar characterizations that are hard to distinguish and compare.

Overall, this paper is motivated by the still fuzzy definition of autonomous systems, related system classes, and self-x characterizations throughout the literature. The goal is to provide a clear distinction between different system concepts and their desired self-x capabilities as a baseline for future work. In our view, a clear distinction of different system concepts and their capabilities is needed as a basis for the discussion of autonomous system concepts and their further development. It supports the definition of a clear picture of future autonomous systems and the combination of progress from similar research fields to finally realize fully autonomous systems on a faster scale. For this reason, the authors of this paper contribute to the current research state on the basis of a systematic literature review to investigate the relations of autonomous and similar system classes, identify popular self-x terms from literature, and introduce a conceptual model for the self-x characterization of autonomous systems and related system classes.

The outline of this work is divided into background, related work, method, results, and a conclusion. The paper starts with a theoretical background on system autonomy, related system classes, and self-x characteristics (see Section 2). Then related work (see Section 3) and the methodology of the study are discussed, including research questions, objectives, and research design (see Section 4). Afterwards, the results of the work are presented with respect to the relations of comparable system classes, a summary of common self-x terms used in the current research state, and finally the conceptual self-x characterization model for autonomous systems (see Section 5). The closing chapter summarizes the results and explicitly answers the identified research questions (see Section 6).

2 BACKGROUND

The background of the paper discusses theoretical concepts on the definition of autonomous systems, related system classes, and self-x characteristics. Thereby, challenges regarding the characterization of autonomous systems and related system classes are summarized to demonstrate the need for a common understanding on self-x capabilities and agreed self-x characterization models across different application domains.

2.1 Autonomous system definition

As mentioned in the introduction, there is no standard today that defines autonomous systems and their capabilities [118, 138]. Depending on the authors and domains, the concept of autonomy and its characteristics are presented differently [77]. Therefore, this paper summarizes the definitions of autonomous systems from the current research state and points out the agreements and disagreements.

Beginning with concepts which are widely accepted, we start with the origin of the word autonomy. It was sourced in the early 17th century from the Greek word *autonomia* or *autonomous* which means self-governing [41, 77, 118, 149, 182].

Therefore, the term originally expresses the right to make decisions and organize independently without external interference. Later, the word has been transferred to automation systems. Thereby, it expanded its meaning to the role of machines that can take over the labor of humans independently [118, 182].

As the concept of autonomy has been broadly discussed since then, its description has been enhanced with further contributions from the literature. A system is described as autonomous if it can pursue its goals without or with only limited external intervention despite uncertain environmental conditions [10, 32, 40, 41, 77, 96, 118, 149, 183]. Therefore, the system needs to be able to perceive its environment, make its own decisions, and act on its environment to solve complex tasks. As the skills sense, analyze, and act are parts of a classical control loop, many authors agree on the statement that every autonomous system is a control system [10, 41, 65, 183]. Compared to classical control systems, they have higher capabilities and can cope with uncertainties or even find solutions to modified goals [40, 41, 65, 118, 183].

Other aspects of autonomy are described differently. First, there is the role of humans that interact with autonomous systems. While some authors say that systems are only autonomous if no human intervention is required [65, 96, 118, 149, 187], others point out the pivotal role of humans for reasons of safety, security, or bias [40, 110]. The authors of the underlying literature still agree that the degree of autonomy is higher or lower depending on the level of human intervention [10, 41, 183], but the role of humans is described differently. Another disagreement is the need for learning capabilities in autonomous systems. Some authors say that learning is superior to other capabilities, but not mandatory [41, 77]. Others point out that learning is an essential part of autonomous systems, so that they are able to learn and evolve in uncertain environments [6, 40, 149]. Another disagreement is related to the characterization of autonomous systems. As pointed out in the introduction, this is typically done on the basis of self-x capabilities that differ between authors and domains. Chapter 2.3 covers this in detail.

To summarize the definition of system autonomy, most authors of the underlying literature agree that a system is autonomous if it can achieve its goals in an uncertain environment with minimal or no human influences. Some authors have different opinions on whether an autonomous system must be completely independent of humans, provide learning capabilities, or have a specific set of self-x characteristics.

2.2 Similar system classifications

Today, there exist similar system concepts along with the idea of autonomous systems, such as automated, autonomic, intelligent, adaptive, and organic systems. This chapter describes each system class based on the underlying literature as a basis for a comparison and delimitation of the concepts.

First, the transition from mass production and mechanization to automated systems began with the third industrial revolution in 1970 [112]. Computers and electronics have been used to digitize production and replace manual labor by machines [182]. Thereby, automated systems execute actions or whole processes using a set of rules with as little human intervention as possible [110]. The main goal of automation is independence [77], so less human labor is required, productivity in terms of energy and material savings is increased, and quality, accuracy, and precision can be improved [149, 182].

Next, the concept of autonomic computing was suggested by Paul Horn from IBM in 2001 [39]. He criticizes the ever-growing complexity of digital infrastructures that undermines the benefits of information technology (IT). The basic idea of autonomic computing is inspired by the human nervous system. Humans can think and live without caring for underlying body functions such as breath, blood circulation, or immune system. The same should become true for computer systems that are supposed to manage themselves and adjust to varying circumstances, while human users can concentrate on their actual tasks. Therefore, Horn proposes to embed the complexity of systems in the infrastructure itself, so that the management of the IT infrastructure can be automated. The general goal of autonomic computing is to provide a zero-cost maintenance and highly reliable system to end users with a transparent interface to complex infrastructures [76].

Paul Horn characterized autonomic computing by eight key elements a system should provide [76]:

(1) it needs to *know itself* in terms of knowledge of its components, current status, capacity and connections to other systems
(2) it must *configure and reconfigure* itself under changing conditions
(3) it should *optimize itself*, so that it can improve the workflow to achieve its system goals
(4) it is supposed to *heal itself*, so that it is able to recover from failure
(5) it has to maintain overall system security by detecting, identifying and *protecting itself* from attacks
(6) it needs to be *aware of its context* and interact with neighbouring systems accordingly
(7) it has to be *open*, because it is interdependent to other systems and therefore needs to function in a heterogeneous world
(8) it should hide complexity from users and be *anticipatory*

Another class of systems are intelligent systems. They typically apply AI for some kind of exercise [77] and are able to deal with uncertain problems or find solutions to new situations based on experience [66]. Therefore, they emulate intelligent behaviors such as learning, adaptability, robustness, optimization, or reasoning [148]. AI technologies are an important base for building intelligent systems [66]. Applications of AI can include, for example, perception of the environment through computer vision, identification of patterns in data, or grouping of observations into categories [176].

Adaptive systems are able to adjust themselves according to internal system states or external influences in the environment [189]. They constantly evaluate their behavior and changes in their performance to achieve a set of objectives or reach a goal in an optimized way [152]. They are described as a subclass of intelligent systems that uses some sort of intelligence to adapt to changes in an uncertain environment or to find solutions to new or modified goals [77]. Self-adaptive systems are capable of operating without human intervention, but typically receive guidance through human policies as higher-level objectives [21]. The literature differentiates between structural and parametric adoption. The first approach is adopted through the exchange of physical elements in a machine or the replacement of components from the application software. Thus, the structure of the system is changed so that components can be added, removed, or replaced. An example of structural adoption can be the replacement of a faulty or worn component, such as a wrench. Parametric adoption modifies the configuration or properties of a system to adapt to changing circumstances [34]. If a setting of a system is adjusted, for example, the speed of an assembly line in case of danger, it is called a parametric adoption.

Similarly to autonomic systems, organic systems are inspired by the simplicity of living systems. They use mechanisms observed in nature to organize complex computing systems that typically consist of many components and solve computationally difficult problems [190]. Thereby, they have sufficient degrees of freedom and are able to adapt to the current requirements of their execution environment to provide robust and reliable services to end users [163]. Organic computing systems focus in particular on self-organizing distributed computing entities in the system structure to effectively achieve the goals of the overall system as a result of local actions. The cooperation of computing resources and the combination of local actions into global functionality is referred to as emergent behavior. It leads to a further level of complexity due to its non-linearity [157]. Comparable to natural organisms, this complexity is hidden in the inner structure of the system itself. From the outside, the complexity is transparent, so that organic systems are easy to operate by humans [190].

In summary, the concept of autonomous systems is comparable to other system classes such as automated, autonomic, intelligent, adaptive, and organic systems [77]. These types of systems are difficult to distinguish, because they have similar definitions and shared characteristics. In the literature, it can be seen that these system class names are therefore often used interchangeably [108, 110, 165, 192] or their relation is described in conflicting ways when comparing different authors [39, 68, 184].

2.3 Self-x characteristics

Self-x capabilities or self-x properties are used to describe system autonomy [8, 110, 175] or the vision of related system concepts such as automated, autonomic [39, 53, 165, 168, 192] or adaptive [68, 108, 184] resources and infrastructures.

The eight key elements of autonomic computing proposed by Horn [76] are widely accepted as a common ground to characterize autonomic systems as well as similar system classes [39, 53, 108, 110, 152, 165, 168, 184, 192]. They are often divided into major and minor attributes based on the understanding that minor attributes contribute to the main objectives and support the abilities indirectly [165, 168]. The main characteristics are also known as self-CHOP capabilities [165]:

- *self-configuration*: The system can automatically configure and reconfigure itself and its components.
- *self-healing*: The system can detect faulty components, diagnose the issue, and heal or recover itself from failure.
- *self-optimizing*: To improve efficiency or performance, the system can optimize itself and its components.
- *self-protecting*: The system is able to defend itself and its components from malicious attacks.

In addition, a self-managing system needs to be aware of itself, know its current external state, integrate into a heterogeneous real-life environment, and hide complexity from end users. Therefore, minor characteristics are often named self-knowledge or self-awareness, context-aware or environment-aware, open, and anticipatory [76, 165]. Other authors name slightly different minor capabilities and add self-monitoring and self-adjusting as supporting autonomic capabilities to detect changing circumstances and make adoptions accordingly [168].

With growing research interest in autonomous systems and related system classes, the number of self-x characteristics grows steadily. Many authors invent new self-x capabilities according to their specialized research context and requirements, so there are hundreds of self-x terms today. Similarly to self-CHOP capabilities, the terms are linked and can be structured hierarchically in various ways into major, minor, and further supportive subproperties [61, 130]. To give you an impression of that, Attachment A.1 presents 33 self-x capabilities of autonomous, autonomic, and adaptive systems that have been extracted from 15 references only. Although there are five papers from each system class across different application domains, almost two-thirds of the references use self-CHOP capabilities to define system characteristics. Therefore, autonomous, adaptive, and autonomic systems are sometimes described identically or at least with similar self-x capabilities [39, 52, 108, 110, 152, 165, 168, 189, 192]. Only a few authors enhance self-CHOP with additional capabilities [39, 39, 53, 108, 110, 165, 184], suggest new individual collections of self-x capabilities [40, 41, 109, 118] or use partially overlapping self-x sets [68, 189].

Additionally, proposed sets of self-x capabilities from different authors for the same system class can be conflicting. To give an example, two different self-x characterizations of autonomous systems are compared. The first is from Sing and Kumar with a general focus on autonomous systems [165]. The other is concerned with an industrial autonomous system and has been suggested by Müller et al. [118]. While Sing and Kumar use self-CHOP main capabilities inspired by Salehi [150] and Horn [76], Müller et al. use self-governance, systematic process execution, adaptability, and self-containedness as major skills. In comparison to Sing and Kumar, they also use more fine-grained minor and subattributes that contribute to the main attributes. Therefore, both models have a different structure that makes the characteristics difficult to compare. The main reasons for this are further examined using three examples (see Figure 1):

- *Different wording (blue)*: Different words are used for the same self-x capabilities. While Müller et al. use the term adaptability which includes abilities for robustness, optimization, and resilience, Sing and Kumar express similar concepts covered by the major characteristics self-healing, self-optimization, and self-protecting.
- *Conflicting layers (green):* Some terms can be found on contradictory layers of the models. Sing and Kumar use self-optimization, for example, as a major characteristic, while Müller et al. use self-optimization as a second-level attribute of adaptability.
- *Different properties (gray):* Some attributes cannot be matched to the other model at all, such as open, anticipatory, or learning abilities.

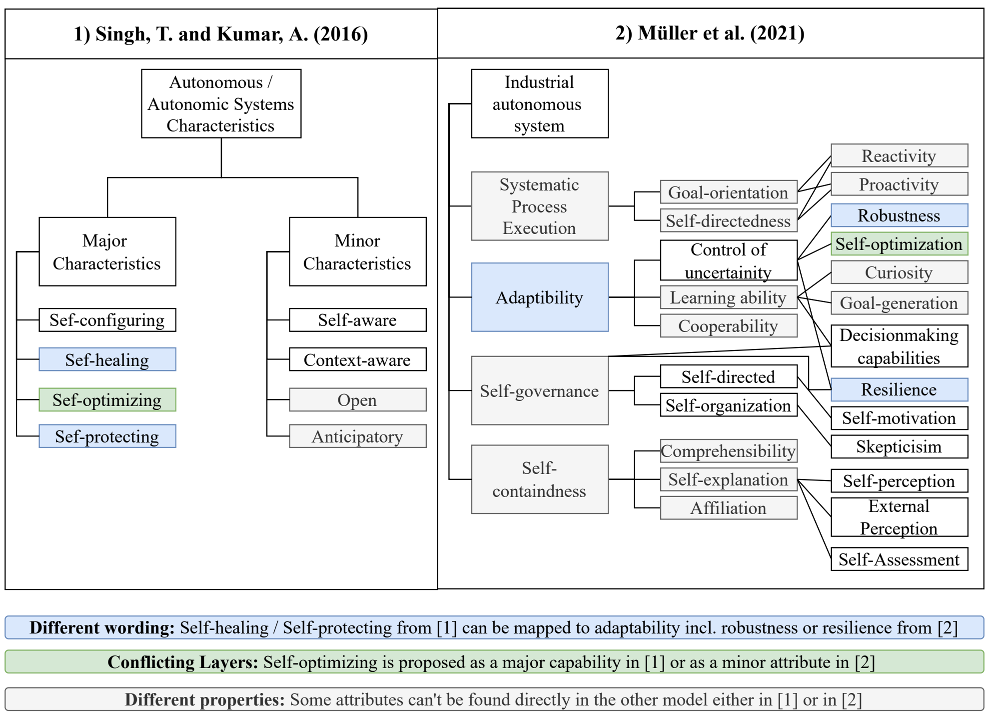

Fig. 1. An example of conflicts between autonomous self-x characterization models on the basis of [118] and [165]

In summary, there is still no consent in the literature on a dedicated set of self-x capabilities that describe autonomous systems or related system classes. In contrast, there are hundreds of self-x terms that have become overused buzzwords and are described differently depending on the area of science and the authors using the term [20, 63]. Moreover, specific sets of self-x characteristics from different authors cannot be compared easily because of a missing common wording for the same terms, a conflicting structure of the properties into major and minor levels, or distinct skill sets in general. Therefore, it can still be challenging today to define the vision for autonomous systems based on a common set of self-x characteristics from the current research state. Additionally, different system classes can have identical, overlapping, or distinct self-x characteristics. This makes it difficult to differentiate the system types and classify a system with one or more system classes based on its capabilities.

3 RELATED WORK

The related work of the study is associated with different research fields such as self-x capability models, the definition and characterization of individual system classes, and the relation of comparable system concepts.

The first group of related papers focuses on the research of self-x capability models for autonomous systems and related system classes. Thereby, Horn has proposed one of the first collections of self-x capabilities to characterize autonomic systems (see Section 2.3). Most of the other underlying research papers suggest self-x capabilities for the characterization of a certain system class based on general system theory [40, 41, 68, 152, 184], the deduction of skills from a certain application domain [109, 118, 130, 177], or previous proposed self-x characterization models, especially the one from Horn [39, 52, 108, 110, 165, 168, 189, 192]. Only one article from Brooks and Roy deals with a systematic review of the self-x requirements of self-engineering systems [20]. The concept of self-engineering systems is closely related to autonomic systems, but has higher capabilities and addresses not only computing systems. It can be seen as a special form of autonomous systems that focuses on maintenance, repair, and overhaul and is applied to a wider range of biological and engineered system types. In their review, the authors manually collect self-x terms used in the field of self-engineering systems and analyze their frequency of use in papers and patents over the years to identify which self-x terms have been the primary area of research. The review concentrates only on material and hardware systems and leaves out other types of system, such as software, control, or operating systems. Two other research studies by Berns and Ghosh [17] and Nuno et al. [130] investigate the relationship and meaning of different self-x properties. Both works provide a valuable basis for defining the meanings of the terms and recognizing synonymous expressions in the course of the study.

The second group of related papers is concerned with the literature on system classes such as automated [182], autonomic [76, 168], adaptive [21, 108, 152, 189], intelligent [66, 148, 176], organic [70, 157, 163, 190], and autonomous systems [40, 65, 110, 118]. The authors of the articles thereby describe basic system definitions and characteristics of each system type to specify details of the concepts. The third group of related papers focuses on the relation of comparable system classes. This group overlaps with the references of the second group of relevant papers in the sense that many authors who do research on a certain system class tend to compare them with one or more other system classes. Most of the authors who write about autonomous systems compare them with system automation [40, 139, 149]. Some authors who write about adaptive systems relate them with autonomic systems in various ways, for example, as complementary classes, identical approaches, or concepts that build on top of each other [39, 68, 184, 192]. The authors of the organic computing domain especially draw a line between the concepts of organic and autonomic systems [157]. Only one article from Hrabia et al. briefly addresses the relation of multiple system classes including autonomous, adaptive, intelligent, autonomic, automated and organic systems. Compared with our study, their work focuses primarily on the definition of a measurement framework to quantify autonomy, rather than providing a detailed discussion of the relationships between system classes and their characteristics [77].

In summary, aspects of autonomous systems, its characteristics and relations to other system classes are distributed in various research papers. Most of the authors write about a single system class and its self-x characteristics with a narrow focus. However, detailed discussions on the relations between autonomous systems and similar system concepts and a holistic view on the self-x properties of multiple system classes are lacking. In addition, the authors from current research state often use self-x capabilities from previous models or individually deduce them from a theoretical or practical research context. In this way, the number of self-x terms is continuously growing, resulting in diverse self-x characterization models with different and often non-comparable self-x abilities and structures of the model. A systematic review of self-x capabilities is missing that identifies popular terms, specifies their meanings, and harmonizes their wording.

4 METHOD

The study addresses three research gaps and questions that will be described in the next subsection of this chapter. The details of the research method are explained in the second subsection.

4.1 Research questions and objectives

Based on the theoretical background and related work, three research questions (RQ) have been identified that will be examined as part of the study.

RQ1: *What relationships do autonomous systems have to comparable system classes?*

The first research question addresses the difficult categorization of system scenarios with one or more system classes. There are comparable system concepts that exist along with the idea of autonomous systems as described in Chapter 2.2. Although system classes are treated as different system types and are investigated in scattered research fields, the system concepts are similar to the vision of autonomous systems and seem to intersect to some extent. Hence, it is notable that some authors in the current research state tend to use system class names interchangeably [108, 110, 165, 192] or specify the details of a system class based on the concepts of another system class [68, 110, 184, 189, 192]. Therefore, it is challenging to draw a line between autonomous systems and similar system classes and categorize system scenarios with a dedicated type of system.

To address the first research question, the relations of similar system classes are explored based on a literature review. Thereby, the details of each system class from current research state are being used as a basis to understand their goals, benefits and capabilities and derive knowledge on their relations afterwards. The goal is to provide a clear picture of autonomous systems and the relations to comparable system classes, so that it becomes easier to classify systems as autonomous or another related system class.

RQ2: *Which self-x properties are used in the literature to describe characteristics of system autonomy and related system classes?*

As outlined in Chapter 2.3, self-x terms are used to describe autonomous systems and related system classes. Because the meanings of the terms are not standardized, there are hundreds of self-x terms today that can be related to one or more system classes. Furthermore, authors have a different understanding of the terms themselves, use another wording for comparable characteristics, or use the terms incorrectly as overused buzzwords [20]. Therefore, it is still challenging to define the vision for autonomous systems based on a common set of self-x characteristics of the current research state. From the viewpoint of the authors, there is a need to identify popular self-x terms, develop a common understanding of their meanings, and identify synonyms to reduce the number of relevant self-x terms to a distinct collection of properties.

Therefore, the second research question addresses the need to collect, harmonize, and unify self-x requirements, so that the specific terms and their understanding among the authors can be aligned. The goal is to systematically find and count the self-x terms used in the underlying literature to provide a list of the most common self-x terms together with their meanings. Thereby, terms that describe the same concepts should be identified, so that the list is condensed and contains unique skills instead of presenting a huge collection with different words for the same capabilities.

RQ3: *How can autonomous systems be characterized on the basis of self-x properties?*

Finally, there are many self-x characterization models that are typically created for a specific scenario, domain, or system class. Due to a different terminology, structure, and their narrow focus, it is difficult to compare existing self-x capability models and rely on a specific set of self-x skills from the current research state for the characterization of individual system scenarios (see Chapter 2.3). From the viewpoint of the authors, there is a lack of a general self-x characterization model for autonomous systems that uses a harmonized set of self-x properties and takes into account not only autonomous systems, but also related system classes with overlapping capabilities.

Therefore, we want to contribute to the current research state by presenting a conceptual model that characterizes autonomous systems on the basis of the results of research questions one and two. First, it is based on a harmonized list of self-x capabilities from current literature so that the most common characteristics for autonomous systems and related system classes are included and condensed to a more unified view. Second, as autonomous systems skills overlap with those from related system classes, the model does not take into account autonomous systems only, but provides a holistic view that characterizes autonomous systems and their related system classes. It is not supposed to list all capabilities an autonomous system needs to provide, but to structure popular self-x capabilities for autonomous systems and related system classes from the literature to achieve a more consistent characterization throughout all system classes. In the end, the characterization model supports characterizing autonomous systems with a common set of skills and helps to categorize a system as autonomous or another related system class based on its desired self-x capabilities.

4.2 Research method

To address the identified research gaps and questions, a systematic literature search is conducted based on a tool chain of a bibliographic database, reference management software, programming techniques and finally a manual review. The study research method is based on a framework of Brocke et al. which comprises five steps as illustrated in Figure 2. All of the steps will be discussed in the following subsections in detail.

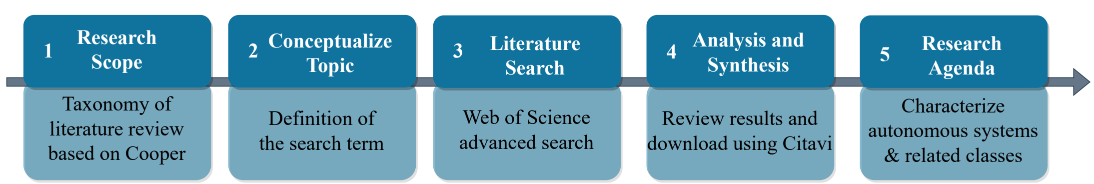

Fig. 2. Framework for literature reviews on the basis of Brocke et al. [80]

4.2.1 Review Scope. In order to clearly define the scope of the review, Brocke et al. suggest using an established taxonomy for literature reviews as a basis [80]. Therefore, this study summarizes the scope of the review based on the general taxonomy for categorizing reviews presented by Cooper [45]. This taxonomy consists of six characteristics that can be found with the impliciations for this study in Table 1.

The focus of the study is a review of fundamental theories of the characterization of autonomous systems (1). Thereby, the goal is to integrate past literature to identify self-x terms and harmonize their meanings to resolute different understandings. Furthermore, the goal is to gain new knowledge about the relation between comparable system concepts and deduct a general applicable self-x characterization model based on the results of the review (2). The perspective of the study is espousal, because the literature is accumulated in the service of synthesizing terms and derive new knowledge (3). Next, the coverage is exhaustive based on a search term that allows to include most of the literature with relevance for the topic. Special attention is thereby paid to the self-x terms and their meanings, while other aspects of the papers are not discussed in detail (4). The gathered papers are grouped according to their self-x terms so that papers with similar ideas and goals with respect to self-x characterization appear together (5). Finally, the results of the study encompass a fundamental theory of system autonomy with a general scope, so that the findings can be applied to various research domains (6).

Table 1. Taxonomy of literature review based on Cooper [45]

No.	Characteristics	Categories
1	Focus	Theories
2	Goal	Integration (Generalization and Conflict Resolution)
3	Perspective	Espousal
4	Coverage	Exhaustive
5	Organization	Conceptual
6	Audience	General Scholars

4.2.2 Conceptualization of the Topic. After the delimitation of the review scope, the first step is to define a search term for the systematic literature review. Basically, we are interested in the literature on autonomous systems and related system classes that discusses their characteristics based on self-x capabilities. Thereby, special attention is directed towards studies that discuss self-x characteristics in general rather than concentrating on the research of specialized self-x skills such as self-healing or self-optimization. Therefore, the search term comprises three building blocks:

(1) **System class:** The first term that needs to be included is one of the system class names of interest as referenced in chapters 2.1 and 2.2. These are *autonomous* systems or one of the comparable system concepts such as *automated*, *autonomic*, *intelligent*, *adaptive* or *organic* systems.
(2) **Computer system:** In this study we are concerned with computer systems or techniques rather than, for example, material or social science. Therefore, the search term also needs to include the words *computing* or *system* to focus on literature with special interest on computing systems.
(3) **Self-x capabilities:** Next the literature contribution needs to deal with self-x characteristics in general and, therefore, needs to name the generic term rather than using only specialized characteristics. Therefore, the article should include the term *self-x* or another word form for the term such as *self-properties*, *self-star*, *self-systems* or *selfware*.

4.2.3 Literature search. Web of Science has been used as a bibliographic database to conduct the literature search. It offers an advanced search query builder that can be used to find research articles based on a search term using logical operators and filter the results on the basis of various exclusion criteria. Therefore, the defined search term has been specified in the query language of Web of Science and has been applied to the literature fields title (TI), abstract (AB), and author keywords (AK):

$$
\begin{aligned}
(\; autonomous \; OR \; autonomic \; OR \; adaptive \; OR \; automated \; OR \; intelligent \; OR \; organic \;) \\
AND \; (\; computing \; OR \; system \;) \\
AND \; (\; self{-}x \; OR \; self{-}properties \; OR \; self{-}star \; OR \; self{-}system \; OR \; selfware \;)
\end{aligned}
\tag{1}
$$

The search query initially resulted in 159 papers. Afterwards, titles have been filtered based on their research area to sort out papers from psychology, neuroscience, chemistry, educational research, and polymer science, so that the result set finally includes papers with a focus on various types of computation systems only. The application of further exclusion criteria has not been necessary, as the results include only English papers, cover years from 2005 - 2023 and encompass paper types including articles, proceedings, or review papers anyway. In the end, the collection of search results comprised 155 hits.

4.2.4 Literature analysis and Synthesis. As a next step, the list of search results has been exported from Web of Science and imported into Citavi. Thereby, we manually reviewed the results and sorted out five papers with duplicate titles that have been published as a journal and as a proceeding paper in the same year. The final list therefore comprises 150 titles that have been automatically downloaded using Citavi full-text search.

4.2.5 Research Agenda. The research agenda covers the main contributions of the study to answer the identified research questions. First, a literature review is conducted to examine the relations of autonomous and similar system classes (see Section 5.1). Second, self-x terms are extracted and harmonized based on a Python script and manual review. The script is used to automate the analysis of the self-x terms from the literature. Thus, the tool parses each PDF file, extracts self-x terms, filters the selection, unifies the word form, and creates a list with self-x terms, their counts of papers, and their references[1]. The resulting list is then used to manually identify common self-x terms used in the underlying literature, define their meanings based on related references, and sort out synonyms (see Section 5.2). Finally, the authors deduce a conceptual self-x characterization model using the harmonized list of self-x terms and the relations of autonomous and similar system classes as a foundation (see Section 5.3).

5 RESULTS

In addition to the current state of research, this chapter answers the identified research questions. Therefore, the relation of comparable system classes is examined as a first step. Furthermore, the self-x characteristics of autonomous and related system classes are extracted, harmonized, and defined, so that a list of common self-x terms from the literature can be presented. On the basis of that, autonomous systems and similar system classes are characterized by the use of the extracted self-x characteristics to finally deduce a conceptual alignment model of autonomous self-x capabilities.

5.1 Relation of similar system classes

Based on the definitions of autonomous, autonomic, automated, intelligent, adaptive, and organic systems, this chapter describes the relation of system classes to answer the first research question.

Autonomous systems are often described as automation systems with higher capabilities for complex or uncertain scenarios [183]. Advanced automated systems can have the appearance of autonomy as they work independently of humans as long as they operate in a defined and limited scope [40]. Therefore, automated systems are often described

[1] Link to Python tool: https://github.com/DLR-SF/Extract_Selfx_Terms

as a subclass of autonomous systems [77]. They also focus on executing processes independently of humans to replace manual labor, increase productivity, or improve quality, but unlike autonomous systems, they do so only in a stable and well-defined environment with clearly defined state spaces [39, 149]. Thereby, the necessary process steps can be programmed based on a set of rules, so that the behavior is fully determined by its external input and internal state [110, 149]. They are limited to repeated tasks, do not make complex decisions, and are unable to manage new situations [118]. Their behavior can generally be described on the basis of a classical control loop with the phases sense, analyze, and act. Autonomous systems can have higher capabilities and extend classical control loops with functionalities to perceive sensed information according to the context, understand current circumstances as a baseline for decision making, and the ability to also solve uncertain or new situations [65].

Autonomic systems are another subclass of autonomous systems [77]. These systems also focus on automation, but in contrast to autonomous systems, they concentrate on automation of the management tasks of hardware and software systems, as well as entire computing networks, to cope with the ever-growing complexity of IT infrastructures [76, 157]. Therefore, they differ from autonomous systems primarily in terms of the system objectives and the environment. Autonomic computing systems try to reduce human effort in the underlying management tasks in complex technical infrastructures so that human users can concentrate on their main exercises [53, 76]. In contrast to that, the goal of autonomous systems is self-governance and therefore the ability to operate independently from external influences [183]. They manage themselves but can have an expanded responsibility to act autonomously, handle uncertainties, or solve new exercises without human interference [65]. Thus, autonomy focuses on self-sufficient and goal-directed action, while autonomicity ensures that maintenance-free operation can happen [169].

Next, there are intelligent systems that emulate intelligent behavior to solve some sort of exercise. They typically use AI technologies to deal with challenges that cannot be effectively solved with rule-based programming techniques [66]. To draw a line between intelligent and autonomous systems, it can be concluded that intelligent systems do not have general intelligence [77]. They use AI to solve a specific sort of challenge that cannot be programmed by a set of rules but do not act fully autonomously in an uncertain environment. System intelligence can therefore be seen as a way to deal with uncertain or new situations in complex and dynamic environments, but, unlike autonomous systems, they do not have the goal of operating independently.

A subclass of intelligent systems are self-adaptive systems that use some sort of intelligence to learn and evolve from experiences. This enables them to fulfill new objectives and adjust to an uncertain environment that is not completely known at design time [77]. While autonomous system goals are self-governance and independence, adaptive systems are concerned with adapting to internal or external disturbances or changes. They typically concentrate on the fulfillment of functionality or the improvement of non-functional requirements like robustness, efficiency, or availability. Therefore, their main focus is to reduce human effort in system interaction, provide dependable systems, and improve system performance [189] rather than replacing human labor with machines. Autonomous systems can be adaptive, but do not necessarily need to depending on the specific use case and scenario. Self-adoption capabilities are especially needed in uncertain or new scenarios where the system needs to adopt to a changing environment or handle new situations. In stable environments or complex computing infrastructures, automated or autonomic capabilities can be sufficient to realize an autonomous system that executes tasks and manages itself independently [77].

Self-adaptive systems are strongly related to other types of system, especially autonomic systems. Many researchers use the terms interchangeably [17, 192], because it is difficult to differentiate the system classes [151]. Researchers who draw a line between both concepts do so in different ways. The first group of authors describes that self-adaptive systems are a subcategory of autonomic systems because they basically adapt to complex environments to manage themselves [46, 63]. The second group of authors describes it the other way around in the sense that self-adaptive systems are a broader category than autonomic computing systems that can address more characteristics than specified in self-managing self-CHOP capabilities [46, 51, 61, 99, 117, 141, 151, 159, 184]. We share the opinion with the second larger group of authors that understand self-adaption as a more general term based on the similarities and differences of the system class characteristics described in Chapter 2.2. Autonomic systems focus on self-management to provide a highly reliable and maintenance-free system with a transparent interface to end users. Adaptive systems modify their own structure or behavior in uncertain environments or attempt to find solutions to new or modified goals to reduce human effort, improve functional or non-functional requirements, or handle unknown situations. Therefore, both types of system have another focus and differentiate according to their main goals, benefits, and environment. On the other hand, the term self-adaption can be mapped to one or more self-managing properties of autonomic computing. They can be seen as specific adaption skills that address healing, protection, and optimization capabilities and adjust on the basis of configuration changes [130]. Therefore, self-adaption can be achieved on the basis of autonomic capabilities, but can also realize additional skills to preserve functional requirements, regulate non-functional requirements, or the use of intelligence to cope with uncertainties or new situations [184].

Another related system class is organic computing. It especially has similarities with autonomic and adaptive systems. Many authors state that they have the same self-x properties as requested in autonomic systems [119, 157, 163, 190]. They also follow autonomic system goals to manage themselves and provide simple interfaces to complex infrastructures for human users [190]. Similarly to adaptive systems, organic systems are also able to dynamically adapt themselves to current requirements of their execution environment and new situations based on intelligence [70, 190]. Therefore, organic systems are described as systems with autonomic and adaptive capabilities [157]. The essential difference from the classes of automatic and adaptive systems is that organic computing systems focus on the organization of large collections of devices that provide services to humans [157]. In comparison, autonomic and adaptive systems are concerned with self-management and self-adaption skills to reduce human effort in system interaction and do not necessarily need to consist of many components [76, 189]. Moreover, organic systems are structured according to biological paradigms [157]. Automatic and adaptive systems can be inspired by nature, but do not necessarily need to. Therefore, an organic system can be seen as a way to realize autonomic and adaptive systems based on nature-inspired design and the cooperation of computing units. However, autonomous and adaptive systems are separate system classes that can be designed independently of organic computing principles. In contrast to autonomous systems, the goal of organic systems is not to transfer human labor to machines and be independent but to provide self-managed and self-adaptive services to human end users so that they can concentrate on their actual tasks.

Figure 3 summarizes the relations of similar system classes. It shows that an automated or autonomic system can be categorized as a subtype of an autonomous system. In contrast to autonomous systems, their capabilities are limited to achieve independence in a stable environment or to cope with complexity by providing self-managing infrastructures. Adaptive systems are a subclass of intelligent systems because they use intelligent algorithms to adapt to internal or external changes that cannot be foreseen at design time. Autonomous systems can be adaptive, but they do not necessarily have to depending on the stability, uncertainty, and novelty of the environment. Nevertheless, autonomous systems would not be able to solve real-world problems in dynamic or unknown environments without intelligent behavior and the ability to self-learn. Therefore, autonomous systems are often described as an intersection of autonomic, automated, and adaptive systems to be able to cope with uncertain environments or modified goals [77]. As a result, it can be concluded that the concept of autonomous systems is directly related to automated, autonomic, and adaptive

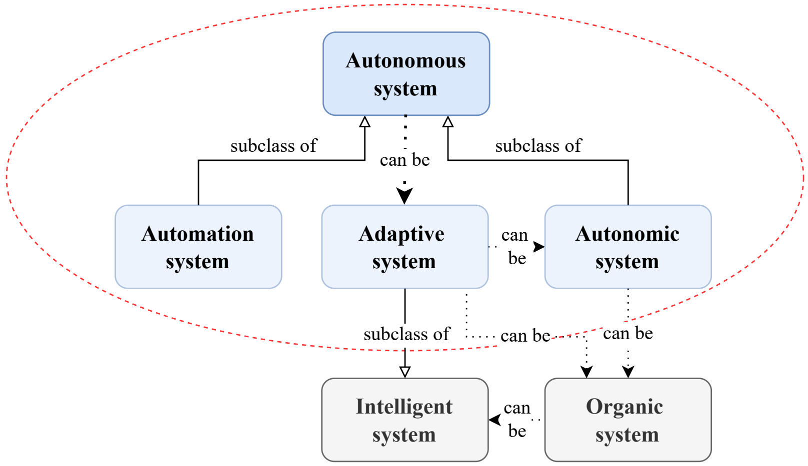

Fig. 3. Relation of autonomous systems and similar system classes

systems. The authors of this work use this relation as a basis for further analysis and as the foundation for the conceptual alignment model of autonomous self-x characteristics.

5.2 Harmonization of self-x characteristics

To answer the second research question, we will collect and harmonize the self-x capabilities of the underlying literature script-based and by manual review as a basis for further usage. Therefore, a Python tool is being used in the first step to identify popular self-x characteristics from papers automatically. In a second step, the resulting list of self-x capabilities is reviewed manually to define their meanings and reduce the list size by identifying possible synonyms.

First, the Python tool is used to extract and count all self-x usages in the 150 selected research papers (see Section 4.2). The statistics of the tool show that 142 of 150 papers use self-x terms (see Figure 4). Overall, the articles mention self-x properties more than 6,000 times. When the terms are unified to nouns and each word is only counted once, the papers mention 184 different self-x capabilities in total. To further break down the collection of self-x characteristics, the tool has been programmed to sort out unusual self-x terms by considering only those characteristics that have been named by at least three papers. This further reduces the list size to 34 self-x properties that can be found together with the number of papers that named them in the Attachment A.2.

```
13-May-24 07:38:42 - root - [ Searched in 150 files. ]
13-May-24 07:38:42 - root - [ Found: ]
13-May-24 07:38:42 - root - [ - 6175 all self-x matches (without excluded or too long words) ]
13-May-24 07:38:42 - root - [ - 184 self-x matches unified and without duplicates ]
13-May-24 07:38:42 - root - [ - 142 papers had matches ]
13-May-24 07:38:42 - root - [ --- Done --- ]
```

Fig. 4. Logged statistics of the use of self-x terms in the underlying literature as a result of the execution of the Python tool

The most common self-x term used in the underlying literature is *self-healing*. It describes a system that detects system errors or faulty components, diagnoses and corrects them [51]. In many papers, the term is named together with the other autonomic self-CHOP capabilities *self-configuration*, *self-optimization*, and *self-protecting* to describe the capabilities of a self-managing system (see Section 2.3). As these skills are not only used to characterize autonomic systems, but also related system classes, *self-management* and self-CHOP are very popular self-x characteristics in the underlying references.

In terms of popularity in papers, the results show that the term self-healing is closely followed by the terms *self-organization* and *self-adaption*. Self-organization describes a system that rearranges or adapts its own structure to achieve a global goal based on local actions [17]. It is used in papers across all system classes, but is predominantly described as a system goal of organic computing to self-organize large collections of devices with shared objectives [157]. Self-adaption describes a system that adjusts itself in response to changing conditions or to improve its function [20]. The term is used as a goal for self-adaptive systems [68, 108, 152, 184, 189], but is also mentioned in the field of autonomic [39, 53, 109, 168] or autonomous systems [40, 41, 110, 118].

The rest of the list has been manually reviewed to identify synonyms and closely related terms with comparable meanings. Self-healing has, for example, *self-repair* and *self-recovery* as synonyms, since the terms basically describe the same characteristics [60, 165]. Another example of comparable terms are *self-optimization* and *self-balancing*. Self-optimization describes the system ability to improve itself with respect to certain goals [17]. Self-balancing is described as the ability of the system to distribute or optimize the load between the resources of a system in the underlying references [195, 196]. It can therefore be seen as a special form of self-optimization which is being used in the context of load balancing of interactive network elements. Following the same pattern, further self-x capabilities have been harmonized. Details about this are covered in the Attachment A.3.

Extracting and harmonizing the self-x list as described above, this paper presents 21 self-x properties as common terms from the literature to describe system autonomy and related system classes. Thereby, 34 terms have been extracted as popular terms by the Python tool and 13 have been identified as synonyms or comparable concepts of another term by manual review. Appendix A.4 provides an overview of the resulting list of self-x capabilities, their mapped synonyms, counts of papers, derived meanings, and references for the meaning.

5.3 Self-x characterization model

To answer the third research question, we want to introduce a conceptual alignment model to structure common self-x properties of autonomous systems using system classes. Therefore, the definition and relation of similar system classes described in Section 5.1 is being used as an underlying resource together with the harmonized list of self-x properties shown in Appendix A.4.

The idea of the alignment approach is to first create a baseline structure of the model that shows the relation of system classes and allows assigning self-x capabilities to autonomous systems or one or more directly related system concepts. According to the principle separation of concerns, the characterization of automated, autonomic, and adaptive systems is then derived individually based on gathered self-x terms. Each characterization begins with determining the system class goals and derives major and minor capabilities from them. In the end, the individual self-x characterization of each system class is used to finally present an overall self-x characterization of autonomous systems that can have autonomic, automated and adaptive self-x properties.

Figure 5 gives an overview of the resulting model. As shown in sections 2.3 and 5.1, different system classes are related to each other and can have similar or overlapping self-x capabilities. The baseline structure of the model captures the relation of different system classes and self-x terms and is therefore composed out of three overlapping circles which visualize capability domains of automated, autonomic and adaptive systems. As autonomous systems can be automated, autonomic, and adaptive systems at the same time, their capability domain is visualized in the center as an intersection of other system classes. Another circle visualizes the capability domain of shared capabilities that are related to all other capability domains. Therefore, the circle has been placed in the background and encloses the others. Using the structure, each self-x capability can be related to at least one capability domain by adding it to a circle or overlapping area. The color of the self-x capabilities highlights the categorization of each skill into system goal, major, or minor characteristic.

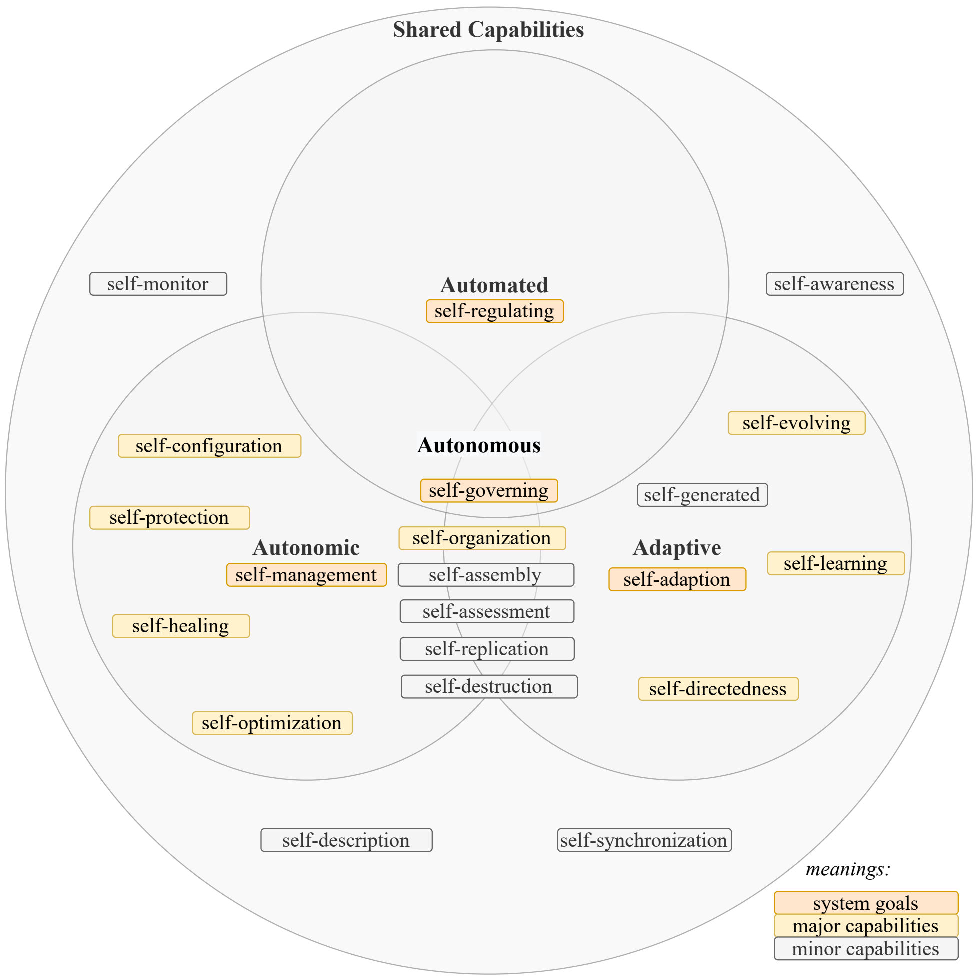

Fig. 5. Conceptual alignment model for autonomous self-x capabilities structured by system class

As a next step, each system class can be characterized by the use of self-x capabilities. Starting with the system class goals, these can be determined as self-regulation, self-adoption, self-management, and self-governing. As described in chapter 2.1, the overall goal of an autonomous system is *self-governing* [118, 183]. Automated systems automate the execution of processes or the fulfillment of tasks independently and therefore have *self-regulation* as their main goal [40, 134]. Adaptive systems have the main focus of adjusting in response to an uncertain environment, maintaining or improving function, or finding solutions to modified goals and therefore have *self-adaption* as the major objective [20]. As autonomic systems attempt to relieve end users from IT management tasks, their overall intent is *self-management* [20].

To continue with the characterization, we derive major characteristics from the system class goals. Self-management has been defined by self-CHOP properties *self-configuration*, *self-healing*, *self-optimization*, and *self-protection* with broad acceptance in the literature (see Section 2.3). Self-adaptive systems adapt their structure or behavior in response to a changing environment and plan actions to pursue modified or new goals independently. Therefore, they need the ability to plan tasks independently to achieve an overall goal despite an uncertain environment [118]. They also need the ability to learn and evolve from experience to gain knowledge, predict forthcoming events, and improve their future operations [8, 133]. Therefore, self-adaptive systems can be characterized by the major capabilities *self-directedness*, *self-learning*, and *self-evolving*. A major goal of organic computing is *self-organization* [157, 190]. As self-adaptive and autonomic systems can be organic at the same time, the term self-organization is added to the shared capability domain of autonomic and adaptive systems as a major system capability by placing it in the overlapping area.

As highlighted in Section 5.1 autonomic and adaptive systems are highly related. From our perspective, self-adoption can be seen as a more general term with an extended meaning, which can be mapped to one or more self-x properties of autonomic computing. While autonomic capabilities support self-management goals, self-adoption can additionally focus on preserving functional requirements, regulating non-functional requirements, or the use of intelligence to cope with uncertainties or new situations [184]. Therefore, the model includes autonomic capabilities as well as self-adoption, but categorizes self-adoption as a main goal of adaptive systems and self-CHOP as minor characteristics of autonomic systems. In this way, the more specific skills of self-managing systems are captured along with self-adaptive systems to indicate that autonomous systems can have autonomic and adaptive system capabilities but do not necessarily need to.

After characterizing system goals and major capabilities of all system classes with self-x terms, there are ten remaining self-x capabilities from the harmonized list that cannot directly be categorized into a capability domain. One term is *self-generated*, which describes systems that are able to generate a cognitive model, input parameters, or knowledge by themselves [4, 106, 185]. The term is directly related to the self-learning ability and is therefore categorized as a minor property of adaptive systems.

Four other attributes, *self-assembly*, *self-assessment*, *self-replication*, and *self-destruction*, have been characterized as minor capabilities of autonomic and adaptive systems. The ability self-assembly describes system environments in which individual elements can self-assemble into complex forms to achieve higher-level objectives [75]. This has been inspired by nature where molecules can form crystals or ants can build bridges [57]. Transferred to computer systems, the term is often used in the context of robots that assemble a group to accomplish tasks that a single robot would not be able to perform [75]. Therefore, the term can be seen as a special form of self-organization or, in broader terms, structural self-adoption that might be beneficial in a specialized field [34]. Self-assessment is needed as a basis for self-healing, self-protecting, self-optimization or in broader sense self-adaption. The effectiveness of the system is being assessed to identify failures, evaluate performance degradation or risks and inform other components to heal, protect, optimize or adapt [60, 118, 149]. Self-replication is the ability to duplicate oneself or individual components of a system [175]. In computer systems, service resources can, for example, be replicated to achieve horizontal scaling, which is typically done by a configuration change of the underlying orchestration system or simply by making a copy of resources. Thus, a self-replicating system must either be able to self-configure, for example, to self-optimize load or, in wider terms, self-adopt its overall structure by adding a new component [20]. Finally, self-destruction refers to the ability of a system to abort execution or exclude components when an unsolvable safety problem that could damage the system or the environment is detected [130, 169]. It is a skill that is needed as a basis for self-managing systems in order to protect, heal, or optimize itself and, in broader sense, for structural adaption in self-adaptive systems.

The last four remaining self-x attributes, *self-monitoring*, *self-awareness*, *self-synchronization* and *self-description*, can contribute to all other capabilities and system classes and therefore relate to the shared domain. Self-monitoring provides the basis for almost all other properties by the ability to observe the system state and detect changing circumstances [168]. Self-awareness is also needed as a basis for all other system classes, as it describes that a system is aware of its internal state which is directly dependent on self-monitoring [168]. Self-synchronization describes the ability to synchronize the behavior of components in a group with a collective goal. In the references it is described as an important basis for systems with a self-organizing or self-assembly ability to synchronize behavior in a group or collection [19, 181, 195, 196]. In general, it is a necessary basis for parallel computing to constrain the ordering of instructions performed by different threads, processes, or collaborating computing systems [162]. Therefore, it is an important basis for different distributed computing scenarios in all types of related system classes. Furthermore, self-descriptive systems are capable of describing their own structure and services in a dynamic or static environment [175]. Therefore, this capability can also be useful for autonomic, automated, and adaptive systems to some extent.

Overall, 21 self-x capabilities have been used to characterize autonomous systems based on their relation to autonomic, adaptive, and automated system classes. Thereby, four characteristics have been identified as system goals, eight as major capabilities, and nine as related or shared minor traits. Autonomous systems can be automated, autonomic, or adaptive at the same time, and therefore inherit necessary capabilities from each related system class to realize the vision of self-governing systems. As use cases across all domains can have different goals and environments, a different type of system, and consequently another set of self-x capabilities might be required. Therefore, the proposed self-x model characterizes each system class and allows to define the vision of autonomy based on a single or combined view of different system classes and self-x requirements depending on individual scenarios. The resulting alignment model is accordingly not supposed to list all requirements an autonomous system needs to provide, but suggests a conceptual model for a more unified view on self-x capabilities and their relations to different system concepts.

6 CONCLUSION

Overall, it becomes clear that there is still a lot of ambiguity between the differences of autonomous and comparable system classes such as automated, intelligent, organic, adaptive, and autonomic systems in the current research state. The system class names are sometimes used interchangeably or their relationship is described in conflicting ways. Accordingly, it is especially difficult to draw a line between autonomous systems and similar system concepts and classify existing or future system scenarios with a dedicated system class (see Sections 2.1 and 2.2).

To describe the vision of autonomy and decide whether a system is autonomous or not, self-x capabilities are being used throughout the literature. Because there is currently no standard that defines the characteristics of autonomous systems, there are many self-x characterizations today. As emphasized by the authors of the paper, this makes self-x characteristics hard to compare, because of a different wording for similar concepts, distinct skill sets, and conflicting structures of self-x characterization models. The result is a still fuzzy characterization of autonomous systems and similar system classes that further complicates the differentiation of system concepts and a common understanding of system autonomy (see Section 2.3).

To create an underlying basis for the discussion of autonomous system concepts and their self-x capabilities, this research provides a systematic literature review on the characterization of autonomous systems and their relation to similar system classes to finally present a conceptual alignment model of autonomous self-x characteristics. Thereby, three research questions have been identified that will be answered explicitly in the following paragraphs (see Section 4).

The first research question addresses the relationship between autonomous systems and comparable system classes. As outlined in Chapter 5.1 the concept of autonomous systems is directly related to automated, adaptive, and autonomic systems. An autonomous system can be an automated system if the system goal is to realize a process from start to end without human intervention in a stable and clearly defined environment. As the scenario is deterministic, a self-regulated system that acts according to rules can be sufficient to achieve system autonomy. If the environment is complex and provides a lot of management effort, autonomic system capabilities could also be needed to improve reliability, lower the effort of maintenance tasks, and thereby decrease the necessary degree of human intervention in system management. If the environment is uncertain additionally or the system has the responsibility to cope with new or modified goals, the system needs to provide self-adaptive skills in order to satisfy functional and non-functional requirements. Therefore, we argue that an autonomous system needs to provide different capabilities according to the scenario and can therefore be classified as autonomic, automated, and adaptive at the same time. Overall, we believe that autonomous systems often occur in a mixed form of similar system classes when applied in real-life domains depending on the specific environment, system goals, and required self-x capabilities.

The second research question addresses the large amount of non-standardized self-x capabilities. As described in Chapter 5.2, there are hundreds of terms that are used in the literature to define the characteristics of autonomous systems and related system classes. In this work, the authors of the study reviewed 150 research papers automatically and manually to present a list of 21 self-x characteristics together with their meanings to the reader. Thereby, the popularity of the terms with respect to the number of papers which named them has been used as a basis to identify relevant terms. In addition, synonyms and closely related concepts have been identified to provide a condensed list of terms as a basis for further usage.

The third research question addresses the need for a holistic self-x capability model to provide a characterization of autonomous systems and related system classes. Thereby, the use case and the specific scenario create a need for a dedicated set of self-x characteristics of an autonomous system from the viewpoint of the authors. These can be automated, autonomic, adaptive, or related to multiple system classes. Because of that, we propose a conceptual alignment model that structures the self-x characterization of system autonomy based on autonomous systems and related system classes, as described in Chapter 5.3. Thereby, the model can be used to define the vision of autonomy based on a popular set of self-x goals, major and minor capabilities, but is not limited to identified terms. In addition, it can help to categorize individual system scenarios with system classes according to their system environment, goals, and self-x capabilities.

Overall, this work emphasizes a clear distinction of different system classes from the literature, but also points out similarities of system concepts, shared self-x capabilities, and mixed system classes in real-life scenarios. As a result, this paper contributes to the clarification of autonomous system concepts by aligning the definition of different system classes and their self-x capabilities as a baseline for future work.

REFERENCES

[1] Hamam Abd and Andreas Koenig. 2020. A Compact Four Transistor CMOS-Design of a Floating Memristor for Adaptive Spiking Neural Networks and Corresponding Self-X Sensor Electronics to Industry 4.0. *TM - Technisches Messen* 87, 1 (2020), S91–S96. https://doi.org/10.1515/teme-2020-0024

[2] Hamam Abd and Andreas Koenig. 2023. Characterization of Adaptive Implementation of Neuromorphic Spiking Sensory Systems On-Chip with Self-X Abilities. *TM - Technisches Messen* 90, 1 (2023), 126–131. https://doi.org/10.1515/teme-2023-0084

[3] Dhaminda B. Abeywickrama and Eila Ovaska. 2016. ADSEng: A Model-based Methodology for Autonomous Digital Service Engineering. In *Proceedings of the 8th International Conference on Management of Digital Ecosystems (MEDES 2016)*. Association for Computing Machinery, New York, NY, USA, 34–42. https://doi.org/10.1145/3012071.3012072

[4] Ursula Addison. 2024. Human-inspired goal reasoning implementations: A survey. *Cognitive Systems Research* 83 (2024), 101181. https://doi.org/10.1016/j.cogsys.2023.101181

[5] Nancy Alonistioti, Andreas Merentitis, Makis Stamatelatos, Egon Schulz, Chan Zhou, Georgios Koudouridis, Bernd Bochow, Mario Schuster, Piet Demeester, Pieter Ballon, Simon Delaere, Markus Mueck, Christian Drewes, Liesbet van der Perre, Jeroen Declerck, Tim Lewis, and Ioannis Chochliouros. 2010. Towards Self-Adaptable, Scalable, Dependable and Energy Efficient Networks: The Self-Growing Concept. In *UBICOMM 2010: The Fourth International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies*, J. L. Mauri, S. Balandin, and C. Dini (Eds.). IARIA XPS Press, Florence, Italy, 324–327. https://doi.org/10.1109/ISGT-Asia.2014.6873884

[6] Eduardo Alonso and Esther Mondragón. 2004. Agency, Learning and Animal-Based Reinforcement Learning. In *Agents and Computational Autonomy*, Matthias Nickles, Michael Rovatsos, and Gerhard Weiss (Eds.), Vol. 2969. Springer, Berlin Heidelberg, 1–6. https://doi.org/10.1007/978-3-540-25928-2_1

[7] Senan Alraho, Qummar Zaman, and Andreas Konig. 2020. Reconfigurable Wide Input Range, Fully-Differential Indirect Current-Feedback Instrumentation Amplifier with Digital Offset Calibration for Self-X Measurement Systems. *TM - Technisches Messen* 87, 1 (2020), S85–S90. https://doi.org/10.1515/teme-2020-0021

[8] Adina Aniculaesei, Jörg Grieser, Andreas Rausch, Karina Rehfeldt, and Tim Warnecke. 2018. Towards a holistic software systems engineering approach for dependable autonomous systems. https://doi.org/10.1145/3194085.3194091

[9] Bjoern Annighoefer, Johannes Reinhart, Matthias Brunner, and Bernd Schulz. 2021. The Concept of an Autonomic Avionics Platform and the Resulting Software Engineering Challenges. In *2021 International Symposium on Software Engineering for Adaptive and Self-managing Systems (SEAMS 2021)*. IEEE, Madrid, Spain, 179–185. https://doi.org/10.1109/SEAMS51251.2021.00031

[10] Panos Antsaklis. 2020. Autonomy and metrics of autonomy. *Annual Reviews in Control* 49 (2020), 15–26. https://doi.org/10.1016/j.arcontrol.2020.05.001

[11] Qazi Mamoon Ashraf, Mohamed Hadi Habaebi, Gopinath Rao Sinniah, Musse Mohamud Ahmed, Sheroz Khan, and Shihab Hameed. 2014. Autonomic Protocol and Architecture for Devices in Internet of Things. In *2014 IEEE Innovative Smart Grid Technologies - ASIA (ISGT ASIA)*. IEEE Explore, Kuala Lumpur, 737–742. https://doi.org/10.1109/ISGT-Asia.2014.6873884

[12] Mehdi Bahrami, Ahmad Faraahi, and Amir Masoud Rahmani. 2010. AGC4ISR, New Software Architecture for Autonomic Grid Computing. In *UKSIM-AMSS First International Conference on Intelligent Systems, Modelling and Simulation (UKSim International Conference on Computer Modelling and Simulation)*. IEEE, Liverpool, UK, 318–321. https://doi.org/10.1109/ISMS.2010.65

[13] Christopher Bailey, Lionel Montrieux, Rogerio de Lemos, Yijun Yu, and Michel Wermelinger. 2014. Run-Time Generation, Transformation, and Verification of Access Control Models for Self-Protection. In *9TH International Symposium on Software Engineering for Adaptive and Self-managing Systems (SEAMS 2014)*. Association for Computing Machinery, New York, NY, USA, 135–144. https://doi.org/10.1145/2593929.2593945

[14] Davide B. Bartolini, Riccardo Cattaneo, Gianluca C. Durelli, Martina Maggio, Marco D. Santambrogio, and Filippo Sironi. 2013. The Autonomic Operating System Research Project - Achievements and Future Directions. In *2013 50TH ACM / EDAC / IEEE Design Automation Conference (DAC) (Design Automation Conference DAC)*. IEEE, Austin, TX, USA, 1–10.

[15] Idilia Batchkova and Tsvetelina Ivanova. 2019. Model-driven development of agent-based cyber-physical systems. *IFAC PAPERSONLINE* 52, 25 (2019), 258–263. https://doi.org/10.1016/j.ifacol.2019.12.492

[16] Nelly Bencomo. 2021. Modeling Autonomic Systems in the time of ML, DevOps and Microservices. In *24TH ACM/IEEE International Conference on Model-driven Engineering Languages and Systems Companion (MODELS-C 2021)*. IEEE, Fukuoka, Japan, 732. https://doi.org/10.1109/MODELS-C53483.2021.00118

[17] Andrew Berns and Sukumar Ghosh. 2009. Dissecting Self-* Properties. In *SASO: 2009 3rd IEEE International Conference on Self-adaptive and Self-organizing Systems (International Conference on Self-Adaptive and Self-Organizing Systems)*. IEEE, San Francisco, CA, USA, 10–19. https://doi.org/10.1109/SASO.2009.25

[18] Victor Pazmino Betancourt, Bo Liu, and Juergen Becker. 2021. Towards Policy-based Task Self-Reallocation in Dynamic Edge Computing Systems. In *2021 IEEE 19th International Conference on Industrial Informatics (INDIN) (IEEE International Conference on Industrial Informatics INDIN)*. IEEE, Palma de Mallorca, Spain, 1–6. https://doi.org/10.1109/INDIN45523.2021.9557374

[19] Benjamin Betting, Julius von Rosen, Lars Hedrich, and Uwe Brinkschulte. 2013. A Highly Dependable Self-adaptive Mixed-Signal Multi-core System-on-Chip. In *Architecture of Computing Systems - ARCS 2013 (Lecture Notes in Computer Science, Vol. 7767)*, H. Kubatova, C. Hochberger, M. Danek, and B. Sick (Eds.). Springer, Berlin, Heidelberg, 122–133. https://doi.org/10.1016/j.vlsi.2014.04.001

[20] Sam Brooks and Rajkumar Roy. 2021. An overview of self-engineering systems. *Journal of Engineering Design* 32, 8 (2021), 397–447. https://doi.org/10.1080/09544828.2021.1914323

[21] Yuriy Brun, Giovanna Di Marzo Serugendo, Cristina Gacek, Holger Giese, Holger Kienle, Marin Litoiu, Hausi Müller, Mauro Pezzè, and Mary Shaw. 2009. Engineering Self-Adaptive Systems through Feedback Loops. In *Software engineering for self-adaptive systems*, Betty H. C. Cheng, Rogério de Lemos, Holger Giese, Paola Inverardi, Jeff Magee, and Betty H.C. Cheng (Eds.). State-of-the-art survey, Vol. 5525. Springer, Berlin, 48–70. https://doi.org/10.1007/978-3-642-02161-9_3

[22] Matthias Brunner, Darbaz Nawzad Darwesh, and Bjoern Annighoefer. 2021. A Safety Process for Self-adaptive Safety-Critical Plug&Fly Avionics. In *2021 IEEE/AIAA 40th Digital Avionics Systems Conference (DASC) (IEEE-AIAA Digital Avionics Systems Conference)*. IEEE, San Antonio, TX, USA, 1–10. https://doi.org/10.1109/DASC52595.2021.9594388

[23] Rainer Buchty, David Kramer, and Wolfgang Karl. 2008. An organic computing approach to sustained real-time monitoring. In *Biologically-inspired Collaborative Computing (International Federation for Information Processing, Vol. 268)*, M. Hinchey, A. Pagnoni, F. J. Rammig, and H. Schmeck (Eds.). Springer, Boston, MA, USA, 151–162. https://doi.org/10.1007/978-0-387-09655-1_14

[24] Rainer Buchty, Jie Tao, and Wolfgang Karl. 2006. Automatic data locality optimization through self-optimization. In *Self-organizing Systems, Proceedings (Lecture Notes in Computer Science, Vol. 4124)*, H. DeMeer and J. P.G. Sterbenz (Eds.). Springer, Berlin, Heidelberg, 187–201. https://doi.org/10.1007/11822035_16

[25] Chad M. Byers and Betty H. C. Cheng. 2015. An Approach to Mitigating Unwanted Interactions between Search Operators in Multi-Objective Optimization. In *GECCO'15: Proceedings of the 2015 Genetic and Evolutionary Computation Conference*, S. Silva (Ed.). Association for Computing Machinery, New York, NY, USA, 655–662. https://doi.org/10.1145/2739480.2754698

[26] Giacomo Cabri and Nicola Capodieci. 2013. Runtime Change of Collaboration Patterns in Autonomic Systems: Motivations and Perspectives. In *2013 IEEE 27TH International Conference on Advanced Information Networking and Applications Workshops (WAINA)*, L. Barolli, F. Xhafa, M. Takizawa, T. Enokido, and H. H. Hsu (Eds.). IEEE, Barcelona, Spain, 1038–1043. https://doi.org/10.1109/WAINA.2013.82

[27] Giacomo Cabri, Nicola Capodieci, Luca Cesari, Rocco de Nicola, Rosario Pugliese, Francesco Tiezzi, and Franco Zambonelli. 2014. Self-expression and Dynamic Attribute-Based Ensembles in SCEL. In *Leveraging Applications of Formal Methods, Verification and Validation: Technologies for Mastering Change, PT I (Lecture Notes in Computer Science, Vol. 8802)*, T. Margaria and B. Steffen (Eds.). Springer, Berlin, Heidelberg, 147–163. https://doi.org/10.1007/978-3-662-45234-9_11

[28] Emre Cakar, Moez Mnif, Christian Mueller-Schloer, Urban Richter, and Hartmut Schmeck. 2007. Towards a quantitative notion of self-organisation. In *2007 IEEE Congress on Evolutionary Computation, Vols 1-10, Proceedings (IEEE Congress on Evolutionary Computation)*. IEEE, Singapore, 4222–4229. https://doi.org/10.1109/CEC.2007.4425022

[29] Radu Calinescu and Marta Kwiatkowska. 2009. CADS*: Computer-Aided Development of Self-* Systems. In *FUNDAMENTAL APPROACHES TO SOFTWARE ENGINEERING, PROCEEDINGS (Lecture Notes in Computer Science, Vol. 5503)*, M. Chechik and M. Wirsing (Eds.). Springer, Berlin, Heidelberg, 421–424. https://doi.org/10.1007/978-3-642-00593-0_29

[30] Nicola Capodieci and Giacomo Cabri. 2013. Managing Deregulated Energy Markets: an adaptive and autonomous Multi-Agent System application. In *2013 IEEE International Conference on Systems, Man, and Cybernetics (SMC 2013) (IEEE International Conference on Systems Man and Cybernetics Conference Proceedings)*. IEEE, Manchester, UK, 758–763. https://doi.org/10.1109/SMC.2013.134

[31] Nicola Capodieci, Emma Hart, and Giacomo Cabri. 2014. Artificial Immune System driven evolution in Swarm Chemistry. In *2014 IEEE Eighth International Conference on Self-adaptive and Self-organizing Systems (SASO) (International Conference on Self-Adaptive and Self-Organizing Systems)*. IEEE, London, UK, 40–49. https://doi.org/10.1109/SASO.2014.16

[32] Cosmin Carabelea, Olivier Boissier, and Adina Florea. 2004. Autonomy in Multi-agent Systems: A Classification Attempt. In *Agents and Computational Autonomy*, Matthias Nickles, Michael Rovatsos, and Gerhard Weiss (Eds.). Springer, Berlin Heidelberg, 103–113. https://doi.org/10.1007/978-3-540-25928-2_9

[33] Roberto Casadei, Mirko Viroli, and Alessandro Ricci. 2020. Collective Adaptive Systems as Coordination Media: The Case of Tuples in Space-Time. In *2020 IEEE International Conference on Autonomic Computing and Self-organizing Systems Companion (ACSOS-C 2020)*, E. ElAraby, S. Tomforde, T. Wood, P. Kumar, C. Raibulet, I. Petri, G. Valentini, P. Nelson, and B. Porter (Eds.). IEEE, Washington, DC, USA, 139–144. https://doi.org/10.1109/ACSOS-C51401.2020.00045

[34] Andre L. D. Cavalcante, Carlos E. Pereira, and José Barata. 2010. Component-Based Approach to the Development of Self-X Automation Systems*. *IFAC Proceedings Volumes* 43, 4 (2010), 222–227. https://doi.org/10.3182/20100701-2-PT-4011.00039

[35] Tarak Chaari and Kaouthar Fakhfakh. 2012. Semantic Modeling and Reasoning at Runtime for Autonomous Systems Engineering. In *2012 9TH International Conference on Ubiquitous Intelligence & Computing and 9th International Conference on Autonomic & Trusted Computing (UIC/ATC)*, B. O. Apduhan, C. H. Hsu, T. Dohi, K. Ishida, L. T. Yang, and J. Ma (Eds.). IEEE, Fukuoka, Japan, 415–422. https://doi.org/10.1109/UIC-ATC.2012.82

[36] Walid Chainbi, Haithem Mezni, and Khaled Ghedira. 2010. An Autonomic Computing Architecture for Self-* Web Services. In *Autonomic Computing and Communications Systems (Lecture Notes of the Institute for Computer Sciences, Social Informatics, and Telecommunications Engineering, Vol. 23)*, A. V. Vasilakos, R. Beraldi, R. Friedman, and M. Mamei (Eds.). Springer, Berlin, Heidelberg, 252—-267. https://doi.org/10.1007/978-3-642-11482-3_17

[37] Jian-Ming Chang, Ching-Chi Hsu, Jiann-Liang Chen, and Han-Chieh Chao. 2009. A Self-Optimization Path-Finding for Geographic Forwarding to Avoid Dead-End in Mobile Ad hoc Networks. In *2009 First International Conference on Future Information Networks*, H. Zhang, E. J. Coyle, S. Y. Kuo, and M. Zukerman (Eds.). IEEE, Beijing, China, 126–130. https://doi.org/10.1109/ICFIN.2009.5339558

[38] Ranganai Chaparadza, Abdelaali Chaoub, Baw Chng, Nigel Davis, Ashutosh Dutta, Muslim Elkotob, Dilip Krishnaswamy, Kaniz Mahdi, Aarne Mammela, Pedro Martinez-Julia, N. Kishor Narang, Lyndon Ong, Mohammad Patwary, Meryem Simsek, Jens Voigt, Brad Kloza, and Matthew Borst. 2022. Systems Optimization. In *2022 IEEE Future Networks World Forum, FNWF*. IEEE, Montreal, QC, Canada,, 1–79. https://doi.org/10.1109/FNWF55208.2022.00143

[39] eep Kumar Chauhan and Amit Sharma. 2012. Autonomic Computing: A long term vision in computing. *Journal of Global Research in Computer Sciences* 3, 5 (2012), 65–67. https://doi.org/10.1109/CIMA.2005.1662304

[40] Hualong Chen, Yuanqiao Wen, Man Zhu, Yamin Huang, Changshi Xiao, Tao Wei, and Axel Hahn. 2021. From Automation System to Autonomous System: An Architecture Perspective. *Journal of Marine Science and Engineering* 9, 6 (2021), 645. https://doi.org/10.3390/jmse9060645

[41] Jerry Chen, Maysam Abbod, and Jiann-Shing Shieh. 2019. Integrations between Autonomous Systems and Modern Computing Techniques: A Mini Review. *Sensors (Basel)* 19, 18 (2019), 3897 pages. https://doi.org/10.3390/s19183897

[42] Qiqi Chen, Ming Li, Gangyan Xu, and George Q. Huang. 2023. Cyber-physical spare parts intralogistics system for aviation MRO. *Advanced Engineering Informatics* 56 (2023), 13 pages. https://doi.org/10.1016/j.aei.2023.101919

[43] Mohan Baruwal Chhetri, Anton V. Uzunov, Quoc Bao Vo, Ryszard Kowalczyk, Michael Docking, Hien Luong, Isuru Rajapakse, and Surya Nepal. 2018. AWaRE - Towards Distributed Self-Management for Resilient Cyber Systems. In *2018 23RD International Conference on Engineering of Complex Computer Systems (ICECCS) (IEEE International Conference on Engineering of Complex Computer Systems-ICECCS)*. IEEE, Melbourne, VIC, Australia, 185–188. https://doi.org/10.1109/ICECCS2018.2018.00028

[44] Frank Chiang and Robin Braun. 2005. A nature inspired multi-agent framework for autonomic service management in ubiquitous computing environments. In *2005 ICSC Congress on Computational Intelligence Methods and Applications (CIMA 2005)*. IEEE, Istanbul, Turkey, 44–48. https://doi.org/10.1109/CIMA.2005.1662304

[45] Harris M. Cooper. 1988. Organizing knowledge syntheses: A taxonomy of literature reviews. *Knowledge in Society* 1, 1 (1988), 104–126. https://doi.org/10.1007/BF03177550

[46] Wenbin Dai, Valeriy Vyatkin, Victor Dubini, and James H. Christensen. 2014. Enhancing Distributed Automation Systems with Efficiency and Reliability by Applying Autonomic Service Management. In *2014 IEEE Emerging Technology and Factory Automation (ETFA)*. IEEE, Barcelona, Spain, 1–8. https://doi.org/10.1109/ETFA.2014.7005123

[47] Werner Damm and Ralf Kalmar. 2017. Autonome Systeme. *Informatik-Spektrum* 40, 5 (2017), 400–408. https://doi.org/10.1007/s00287-017-1063-0

[48] Kai Ding, Xudong Zhang, Felix T. S. Chan, Ching-Yuen Chan, and Chuang Wang. 2019. Training a Hidden Markov Model-Based Knowledge Model for Autonomous Manufacturing Resources Allocation in Smart Shop Floors. *IEEE ACCESS* 7 (2019), 47366–47378. https://doi.org/10.1109/ACCESS.2019.2909306

[49] Mauro Dragone. 2012. A BDI Model for Component and Service-Based Systems: Self-OSGi. In *Advances on Practical Applications of Agents and Multi-agent Systems (Advances in Intelligent and Soft Computing, Vol. 155)*, Y. Demazeau, J. P. Muller, J. M.C. Rodriguez, and J. B. Perez (Eds.). Springer, Berlin, Heidelberg, 67–72. https://doi.org/10.1007/978-3-642-28786-2_7

[50] Mauro Dragone. 2012. Component & Service-based Agent Systems: Self-OSGI. In *ICAART: Proceedings of the 4th International Conference on Agents and Artificial Intelligence, Vol 1*, J. Filipe and A. Fred (Eds.). SciTePress, Vilamoura, Algarve, Portugal, 200–210. https://doi.org/10.5220/0003890302000210

[51] Meenakshi D'Souza and Rajanikanth N. Kashi. 2019. Avionics Self-adaptive Software: Towards Formal Verification and Validation. In *Distributed Computing and Internet Technology, ICDCIT 2019 (Lecture Notes in Computer Science, Vol. 11319)*, G. Fahrnberger, S. Gopinathan, and L. Parida (Eds.). Springer, Cham, 3–23. https://doi.org/10.1007/978-3-030-05366-6_1

[52] Fu Duan, Xiaoli Li, Yi Liu, and Yun Fang. 2011. Towards Autonomic Computing: A New Self-Management Method. In *Artificial Intelligence and Computational Intelligence*, Hepu Deng, Duoqian Miao, Jingsheng Lei, and Fu Lee Wang (Eds.), Vol. 7002. Springer, Berlin, Heidelberg, 292–299. https://doi.org/10.1007/978-3-642-23881-9_39

[53] Lian Duan and Li Da Xu. 2021. Data Analytics in Industry 4.0: A Survey. *Information Systems Frontiers* (2021), 1572–9419. https://doi.org/10.1007/s10796-021-10190-0

[54] Benedikt Eberhardinger, Hella Seebach, Alexander Knapp, and Wolfgang Reif. 2014. Towards Testing Self-organizing, Adaptive Systems. In *Testing Software and Systems (ICTSS 2014) (Lecture Notes in Computer Science, Vol. 8763)*, M. G. Merayo and E. M. DeOca (Eds.). Springer, Berlin, Heidelberg, 180–185. https://doi.org/10.1007/978-3-662-44857-1_13

[55] Mica R. Endsley and David B. Kaber. 1999. Level of automation effects on performance, situation awareness and workload in a dynamic control task. *Ergonomics* 42, 3 (1999), 462–492. https://doi.org/10.1080/001401399185595

[56] Faramarz Safi Esfahani, Masrah Azrifah Azmi Murad, Md. Nasir B. Sulaiman, and Nur Izura Udzir. 2011. Adaptable Decentralized Service Oriented Architecture. *Journal of Systems and Software* 84, 10 (2011), 1591–1617. https://doi.org/10.1016/j.jss.2011.03.031

[57] Luis A. Estrada-Jimenez, Terrin Pulikottil, Sanaz Nikghadam-Hojjati, and Jose Barata. 2023. Self-Organization in Smart Manufacturing— Background, Systematic Review, Challenges and Outlook. *IEEE Access* 11 (2023), 10107–10136. https://doi.org/10.1109/ACCESS.2023.3240433

[58] Ali Farahani, Eslam Nazemi, and Giacomo Cabri. 2016. Self-* Properties in Collective Adaptive Systems. In *UBICOMP'16 Adjunct: Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. Association for Computing Machinery, New York, NY, USA, 1309–1314. https://doi.org/10.1145/2968219.2979127

[59] Luis M. Fernandez-Carrasco, Hugo Terashima-Marin, and Manuel Valenzuela-Rendon. 2008. On the Path Towards Autonomic Computing: Combining Swarm Intelligence and Excitable Media Models. In *Proceedings of the Special Session of the Seventh Mexican International Conference on Artificial Intelligence - Micai 2008 (Mexican International Conference on Artificial Intelligence-MICAI)*, A. Gelbukh and E. F. Morales (Eds.). IEEE, Atizapan de Zaragoza, Mexico, 192–198. https://doi.org/10.1109/MICAI.2008.51

[60] Francesco Flammini, Cristina Alcaraz, Emanuele Bellini, Stefano Marrone, Javier Lopez, and Andrea Bondavalli. 2022. Towards Trustworthy Autonomous Systems: Taxonomies and Future Perspectives. *IEEE Transactions on Emerging Topics in Computing* (2022), 1–13. https://doi.org/10.1109/TETC.2022.3227113

[61] Regina Frei and Jose Barata. 2010. Distributed systems - from natural to engineered: three phases of inspiration by nature. *International Journal of Bio-inspired Computation* 2, 3-4 (2010), 258–270. https://doi.org/10.1504/IJBIC.2010.033094

[62] Regina Frei and Giovanna Marzo Di Serugendo. 2011. Advances in complexity engineering. *International Journal of Bio-inspired Computation* 3, 4 (2011), 199–212. https://doi.org/10.1504/IJBIC.2011.041144

[63] Regina Frei and Giovanna Marzo Di Serugendo. 2011. Concepts in complexity engineering. *International Journal of Bio-inspired Computation* 3, 2 (2011), 123–139. https://doi.org/10.1504/IJBIC.2011.039911

[64] Maxim Friesen, Lukasz Wisniewski, and Jurgen Jasperneite. 2022. Machine Learning for Zero-Touch Management in Heterogeneous Industrial Networks - A Review. In *18th IEEE International Workshop on Factory Communication Systems 2022 (Wfcs 2022)*. IEEE, Pavia, Italy, 33–40. https://doi.org/10.1109/WFCS53837.2022.9779183

[65] Thomas Gamer, Mario Hoernicke, Benjamin Kloepper, Reinhard Bauer, and Alf J. Isaksson. 2019. The Autonomous Industrial Plant -Future of Process Engineering, Operations and Maintenance. *IFAC-PapersOnLine* 52, 1 (2019), 454–460. https://doi.org/10.1016/j.ifacol.2019.06.104

[66] Rinku Garg, Ashish Sharma, Aditi Sexana, Harpreet Kaur, and Sheela Bijlwan. 2023. Growth and Application of Intelligent Systems in Various Fields. In *2023 3rd Asian Conference on Innovation in Technology (ASIANCON)*. IEEE, Ravet IN, India, 1–6. https://doi.org/10.1109/ASIANCON58793.2023.10270116

[67] Elena Gerken, Qummar Zaman, Senan Alraho, and Andreas Koenig. 2023. Development of a Self-X Sensory Electronics for Anomaly Detection and its Conceptual Implementation on an XMR-based Angular Decoder Prototype. *TM - Technisches Messen* 90, 1 (2023), 20–26. https://doi.org/10.1515/teme-2023-0086

[68] Omid Gheibi, Danny Weyns, and Federico Quin. 2021. Applying Machine Learning in Self-adaptive Systems: A Systematic Literature Review. *ACM Trans. Auton. Adapt. Syst.* 15, 3 (2021), Article 9. https://doi.org/10.1145/3469440

[69] Anastasios Gounaris, Christos Yfoulis, Rizos Sakellariou, and Marios D. Dikaiakos. 2008. Robust runtime optimization of data transfer in queries over Web Services. In *2008 IEEE 24th International Conference on Data Engineering, vols 1-3 (IEEE International Conference on Data Engineering)*. IEEE, Cancun, Mexico, 596–605. https://doi.org/10.1109/ICDE.2008.4497468

[70] Matthias Güdemann, Florian Nafz, Frank Ortmeier, Hella Seebach, and Wolfgang Reif. 2008. A Specification and Construction Paradigm for Organic Computing Systems. In *Proceedings / Second IEEE International Conference on Self-Adaptive and Self-Organizing Systems, SASO 2008*, Sven Brueckner (Ed.). IEEE, Piscataway, NJ, 233–242. https://doi.org/10.1109/SASO.2008.66

[71] Matthias Guedemann, Florian Nafz, Frank Ortmeier, Hella Seebach, and Wolfgang Reif. 2008. A Specification and Construction Paradigm for Organic Computing Systems. In *SASO 2008: second IEEE international conference on self-adaptive and self-organizing systems, proceedings*, S. Brueckner, P. Robertson, and U. Bellur (Eds.). Venezia, Italy, IEEE, 233–242. https://doi.org/10.1109/SASO.2008.66

[72] Matthias Guedemann, Frank Ortmeier, and Wolfgang Reif. 2006. Formal modeling and verification of systems with self-x properties. In *Autonomic and Trusted Computing, Proceedings (Lecture Notes in Computer Science, Vol. 4158)*, L. T. Yang, H. Jin, J. Ma, and T. Ungerer (Eds.). Springer, Berlin, Heidelberg, 38–47. https://doi.org/10.1007/11839569_4

[73] Robrecht Haesevoets, Danny Weyns, Tom Holvoet, and Wouter Joosen. 2009. A Formal Model for Self-Adaptive and Self-Healing Organizations. In *2009 ICSE Workshop on Software Engineering for Adaptive and Self-managing Systems*. IEEE, Vancouver, BC, Canada, 116–125. https://doi.org/10.1109/SEAMS.2009.5069080

[74] Saad Sajid Hashmi, Hoa Khanh Dam, Peter Smet, and Mohan Baruwal Chhetri. 2022. Towards Antifragility in Contested Environments: Using Adversarial Search to Learn, Predict, and Counter Open-Ended Threats. In *2022 IEEE International Conference on Autonomic Computing and Self-organizing Systems (Acsos 2022)*, R. Casadei, E. DiNitto, I. Gerostathopoulos, D. Pianini, I. Dusparic, T. Wood, P. Nelson, E. Pournaras, N. Bencomo, S. Gotz, C. Krupitzer, and C. Raibulet (Eds.). IEEE, CA, USA, 141–146. https://doi.org/10.1109/ACSOS55765.2022.00032

[75] Mary Katherine Heinrich, Mohammad Divband Soorati, Tanja Katharina Kaiser, Mostafa Wahby, and Heiko Hamann. 2019. Swarm robotics: Robustness, scalability, and self-X features in industrial applications. *it - Information Technology* 61, 4 (2019), 159–167. https://doi.org/10.1515/itit-2019-0003

[76] Paul Horn. 2001. *Autonomic Computing: IBM's Perspective on the State of Information Technology.* International Business Machines Corporation, Armonk, New York.

[77] Christopher-Eyk Hrabia, Nils Masuch, and Sahin Albayrak. 2015. A Metrics Framework for Quantifying Autonomy in Complex Systems. In *Multiagent System Technologies*. Springer, Cham, 22–41. https://doi.org/10.1007/978-3-319-27343-3_2

[78] Pawel Idziak and Siobhan Clarke. 2014. 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 (SASOW) (International Conference on Self-Adaptive and Self-Organizing Systems)*. IEEE, London, UK, 137–143. https://doi.org/10.1109/SASOW.2014.23

[79] Tomoko Izumi, Taisuke Izumi, Fukuhito Ooshita, Hirotsugu Kakugawa, and Toshimitsu Masuzawa. 2009. A Biologically Inspired Self-Adaptation of Replica Density Control. *IEICE Transactions on Information and Systems* E92D, 5 (2009), 1125–1136. https://doi.org/10.1587/transinf.E92.D.1125

[80] Jan vom Brocke, Alexander Simons, Björn Niehaves, Kai Riemer, Ralf Plattfaut, and Anne Cleven. 2009. Reconstructing the giant: On the importance of rigour in documenting the literature search process. In *European Conference on Information Systems*, Susan Newell, Edgar A. Whitley, Nancy Pouloudi, Jonathan Wareham, and Lars Mathiassen (Eds.). AIS eLibrary, Verona, Italy, 2206–2217.

[81] Fahad Javed, Naveed Arshad, Fredrik Wallin, Iana Vassileva, and Erik Dahlquist. 2010. An Adaptive Optimization Model for Power Conservation in the Smart Grid. In *IEEE International Conference on Systems, Man and Cybernetics (SMC 2010) (IEEE International Conference on Systems Man and Cybernetics Conference Proceedings)*. IEEE, Istanbul, Turkey, 1563–1570. https://doi.org/10.1109/ICSMC.2010.5642343

[82] Muhammad Akmal Johar and Andreas Koenig. 2011. Case Study of an Intelligent AMR Sensor System with Self-x Properties. In *Soft Computing in Industrial Applications (Advances in Intelligent and Soft Computing, Vol. 96)*, A. GasparCunha, R. Takahashi, G. Schaefer, and L. Costa (Eds.). Springer, Berlin, Heidelberg, 337–346. https://doi.org/10.1007/978-3-642-20505-7_30

[83] Elsy Kaddoum, Marie-Pierre Gleizes, Jean-Pierre George, and Gauthier Picard. 2009. Characterizing and Evaluating Problem Solving Self-* Systems. In *2009 Computation World: Future Computing, Service Computation, Cognitive, Adaptive, Content, Patterns*. IEEE, Athens, Greece, 137–145. https://doi.org/10.1109/ComputationWorld.2009.100

[84] Samurdhi Karunaratne, Ramy Atawia, Erma Perenda, and Haris Gacanin. 2018. Artificial Intelligence Driven Optimization of Channel and Location in Wireless Networks. In *2018 IEEE Globecom Workshops (GC WKSHPS) (IEEE Globecom Workshops)*. IEEE, Abu Dhabi, United Arab Emirates, 1–6. https://doi.org/10.1109/GLOCOMW.2018.8644283

[85] Rajanikanth N. Kashi, Meenakshi D'Souza, and Koyalkar Raman Kishore. 2017. Incorporating Formal Methods and Measures Obtained through Analysis, Simulation Testing for Dependable Self-Adaptive Software in Avionics Systems. In *COMPUTE'17: Proceedings of the 10th Annual Acm India Compute Conference*. Association for Computing Machinery, New York, NY, USA, 137–142. https://doi.org/10.1145/3140107.3140128

[86] Holger Kasinger, Bernhard Bauer, and Joerg Denzinger. 2009. Design Pattern for Self-Organizing Emergent Systems Based on Digital Infochemicals. In *SIXTH IEEE International Conference and Workshop on the Engineering of Autonomic and Autonomous Systems: EASE 2009*, R. Sterritt (Ed.). IEEE, San Francisco, CA, USA, 45–55. https://doi.org/10.1109/EASe.2009.8

[87] Florian Kluge, Joerg Mische, Sascha Uhrig, and Theo Ungerer. 2008. An operating system architecture for organic computing in embedded real-time systems. In *Autonomic and Trusted Computing, Proceedings (Lecture Notes in Computer Science, Vol. 5060)*, C. Rong, M. G. Jaatun, F. E. Sandnes, L. Yang, and J. Ma (Eds.). Springer, Berlin, Heidelberg, 343–357. https://doi.org/10.1007/978-3-540-69295-9_28

[88] Andreas Koenig. 2009. Automated and Holistic Design of Intelligent and Distributed Integrated Sensor Systems with Self-x Properties for Applications in Vision, Robotics, Smart Environments, and Culinary Assistance Systems. In *Advances In Neuro-information Processing, PT II (Lecture Notes in Computer Science, Vol. 5507)*, M. Koppen, N. Kasbov, and G. Coghill (Eds.). Springer, Berlin, Heidelberg, 827–834. https://doi.org/10.1007/978-3-642-03040-6_101

[89] Marcus Komann and Dietmar Fey. 2006. Marching pixels - Using organic computing principles in embedded parallel hardware. In *PAR ELEC 2006: International Symposium On Parallel Computing In Electrical Engineering, Proceedings*, R. S. Bilof (Ed.). IEEE, Bialystok, Poland, 369–373. https://doi.org/10.1109/PARELEC.2006.52

[90] Alberto Arkader Kopiler, Ines de Castro Dutra, and Felipe M. G. Franca. 2008. Personal Autonomic Desktop Manager with a circulatory computing approach. In *Proceedings Of The Fifth IEEE International Workshop On Engineering Of Autonomic & Autonomous Systems (EASE 2008)*, R. Sterritt, D. W. Bustard, D. Lewis, and J. J. Fleck (Eds.). IEEE, Belfast, UK, 119–127. https://doi.org/10.1109/EASe.2008.20

[91] Apostolos Kotsialos and Adam Poole. 2013. Autonomic Systems Design for ITS Applications. In *2013 16TH International IEEE Conference on Intelligent Transportation Systems - (ITSC) (IEEE International Conference on Intelligent Transportation Systems-ITSC)*. IEEE, The Hague, Netherlands, 178–183. https://doi.org/10.1109/ITSC.2013.6728230

[92] David Kramer, Rainer Buchty, and Wolfgang Karl. 2011. A Light-Weight Approach for Online State Classification of Self-organizing Parallel Systems. In *Architecture of Computing Systems - ARCS 2011 (Lecture Notes in Computer Science, Vol. 6566)*, M. Berekovic, W. Fornaciari, U. Brinkschulte, and C. Silvano (Eds.). Springer, Berlin, Heidelberg, 183–194. https://doi.org/10.1007/978-3-642-19137-4_16

[93] Christian Kroeher, Klaus Schmid, Simon Paasche, and Christian Sauer. 2021. Combining Central Control with Collective Adaptive Systems. In *2021 IEEE International Conference on Autonomic Computing and Self-organizing Systems Companion (ACSOS-C 2021)*, E. ElAraby, V. Kalogeraki, D. Pianini, F. Lassabe, B. Porter, S. Ghahremani, I. Nunes, M. Bakhouya, and S. Tomforde (Eds.). IEEE, DC, USA, 56–61. https://doi.org/10.1109/ACSOS-C52956.2021.00035

[94] Christian Krupitzer, Julian Otto, Felix Maximilian Roth, Alexander Frommgen, and Christian Becker. 2017. Adding Self-Improvement to an Autonomic Traffic Management System. In *ICAC 2017*, Xiaorui Wang (Ed.). IEEE, Piscataway, NJ, 209–214. https://doi.org/10.1109/ICAC.2017.16

[95] Heng Kuang, Olga Ormandjieva, Stan Klasa, Noorulain Khurshid, and Jamal Benthar. 2009. Towards Specifying Reactive Autonomic Systems with a Categorical Approach: A Case Study. In *SOFTWARE ENGINEERING RESEARCH, MANAGEMENT AND APPLICATIONS 2009 (Studies in Computational Intelligence, Vol. 253)*, R. Lee and N. Ishii (Eds.). Springer, Berlin, Heidelberg, 119–134. https://doi.org/10.1007/978-3-642-05441-9_11

[96] Thomas Kuhn and Peter Liggesmeyer. 2019. *Autonome Systeme: Potenziale und Herausforderungen*. transcript Verlag, Bielefeld. https://doi.org/10.1515/9783839443958

[97] Paulo Leitao, Paul Valckenaers, and Emmanuel Adam. 2009. Self-Adaptation for Robustness and Cooperation in Holonic Multi-Agent Systems. In *Transactions on Large-scale Data- and Knowledge-centered Systems I (Lecture Notes in Computer Science, Vol. 5740)*, A. Hameurlain, J. Kung, and R. Wagner (Eds.). Springer, Berlin, Heidelberg, 267–288. https://doi.org/10.1007/978-3-642-03722-1_11

[98] Pierre Leone, Marina Papatriantafilou, Elad M. Schiller, and Gongxi Zhu. 2010. Chameleon-MAC: Adaptive and Self-Star Algorithms for Media Access Control in Mobile Ad Hoc Networks. In *Stabilization, Safety, and Security of Distributed Systems (Lecture Notes in Computer Science, Vol. 6366)*, S. Dolev, J. Cobb, M. Fischer, and M. Yung (Eds.). Springer, Berlin, Heidelberg, 468–488. https://doi.org/10.1007/978-3-642-16023-3_37

[99] Lima Ribeiro, Admilson de Ribamar, Fernando Mendonca de Almeida, Edward David Moreno, and Carlos A. E. Montesco. 2016. A Management Architectural Pattern for Adaptation System in Internet of Things. In *2016 International Wireless Communications and Mobile Computing Conference (IWCMC) (International Wireless Communications and Mobile Computing Conference)*. IEEE, Paphos, Cyprus, 576–581. https://doi.org/10.1109/IWCMC.2016.7577121

[100] Cheng-Li Liu and Sheue-Ling Hwang. 2000. A performance measuring model for dynamic quality characteristics of human decision-making in automation. *Theoretical Issues in Ergonomics Science* 1, 3 (2000), 231–247. https://doi.org/10.1080/14639220110037470

[101] Josep Lluis Arcos, Juan A. Rodriguez-Aguilar, and Bruno Rosell. 2008. Engineering autonomic electronic institutions. In *Engineering Environment-mediated Multi-agent Systems (Lecture Notes in Artificial Intelligence, Vol. 5049)*, D. Weyns, S. A. Brueckner, and Y. Demazeau (Eds.). Springer, Berlin, Heidelberg, 76–87. https://doi.org/10.1007/978-3-540-85029-8_6

[102] Inga Loeser, Martin Braun, Christian Gruhl, Jan-Hendrik Menke, Bernhard Sick, and Sven Tomforde. 2022. The Vision of Self-Management in Cognitive Organic Power Distribution Systems. *Energies* 15, 3 (2022), 1–20. https://doi.org/10.3390/en15030881

[103] Georgios Loukas and Gulay Oke. 2007. A biologically inspired denial of service detector using the random neural network. In *2007 IEEE International Conference on Mobile Ad-hoc and Sensor Systems, Vols 1-3 (IEEE International Conference on Mobile Ad-hoc and Sensor Systems)*. IEEE, Pisa, Italy, 723–728. https://doi.org/10.1109/MOBHOC.2007.4428683

[104] Andreas Lund, Benjamin Betting, and Uwe Brinkschulte. 2015. Design and Evaluation of a bio-inspired, distributed Middleware for a Multiple Mixed-Core System on Chip. In *2015 IEEE 18th International Symposium on Real-Time Distributed Computing Workshops*. IEEE, Auckland, New Zealand, 80–88. https://doi.org/10.1109/ISORCW.2015.46

[105] Andreas Lund and Uwe Brinkschulte. 2021. Task-allocation in a large-scaled hierarchical many-core topology. *Concurrency and Computation-practice & Experience* 33, 14, SI (2021), 1–8. https://doi.org/10.1002/cpe.5731

[106] Holger Lyre. 2020. The State Space of Artificial Intelligence. *Minds and Machines* 30, 3 (2020), 325–347. https://doi.org/10.1007/s11023-020-09538-3

[107] Raphael Maas, Erik Maehle, and Karl-Erwin Grosspietsch. 2012. Applying the Organic Robot Control Architecture ORCA to Cyber-Physical Systems. In *2012 38TH Euromicro Conference on Software Engineering and Advanced Applications (SEAA) (EUROMICRO Conference Proceedings)*, V. Cortellessa, H. Muccini, and O. Demirors (Eds.). IEEE, Cesme, Turkey, 250–257. https://doi.org/10.1109/SEAA.2012.74

[108] Frank D. Macías-Escrivá, Rodolfo Haber, Raul Del Toro, and Vicente Hernandez. 2013. Self-adaptive systems: A survey of current approaches, research challenges and applications. *Expert Systems with Applications* 40, 18 (2013), 7267–7279. https://doi.org/10.1016/j.eswa.2013.07.033

[109] Abdul Mateen, Basit Raza, Muhammad Sher, M. M. Awais, and Norwatti Mustapha. 2014. Workload management: a technology perspective with respect to self-* characteristics. *Artificial Intelligence Review* 41, 4 (2014), 463–489. https://doi.org/10.1007/s10462-012-9320-8

[110] Fan Mo, Fabio Marco Monetti, Agajan Torayev, Hamood Ur Rehman, Jose A. Mulet Alberola, Nathaly Rea Minango, Hien Ngoc Nguyen, Antonio Maffei, and Jack C. Chaplin. 2023. A maturity model for the autonomy of manufacturing systems. *The International Journal of Advanced Manufacturing Technology* 126, 1-2 (2023), 405–428. https://doi.org/10.1007/s00170-023-10910-7

[111] Mischa Moestl, Johannes Schlatow, Rolf Ernst, Henry Hoffmann, Arif Merchant, and Alexander Shraer. 2016. Self-aware systems for the Internet-of-Things. In *2016 International Conference on Hardware/software Codesign and System Synthesis (CODES+ISSS)*. IEEE, Pittsburgh, PA, USA, 1–9. https://doi.org/10.1145/2968456.2974043

[112] Dimitris Mourtzis. 2021. Towards the 5th industrial revolution: A literature review and a framework for process optimization based on big data analytics and semantics. *Journal of Machine Engineering* 21, 3 (2021), 5–39. https://doi.org/10.36897/jme/141834

[113] Nizar Msadek, Rolf Kiefhaber, and Theo Ungerer. 2014. A Trust- and Load-Based Self-Optimization Algorithm for Organic Computing Systems. In *2014 IEEE Eighth International Conference on Self-adaptive and Self-organizing Systems (SASO) (International Conference on Self-Adaptive and Self-Organizing Systems)*. IEEE, London, UK, 177–178. https://doi.org/10.1109/SASO.2014.32

[114] Nizar Msadek, Rolf Kiefhaber, and Theo Ungerer. 2015. A trustworthy, fault-tolerant and scalable self-configuration algorithm for Organic Computing systems. *Journal of Systems Architecture* 61, 10, SI (2015), 511–519. https://doi.org/10.1016/j.sysarc.2015.07.012

[115] Nizar Msadek, Rolf Kiefhaber, and Theo Ungerer. 2015. Trustworthy Self-optimization in Organic Computing Environments. In *Architecture of Computing Systems - ARCS 2015 (Lecture Notes in Computer Science, Vol. 9017)*, L. M. Pinho, W. Karl, A. Cohen, and U. Brinkschulte (Eds.). Springer, Cham, 123–134. https://doi.org/10.1007/978-3-319-16086-3_10

[116] Nizar Msadek and Theo Ungerer. 2017. Trustworthy self-optimization for organic computing environments using multiple simultaneous requests. *Journal of Systems Architecture* 75, SI (2017), 26–34. https://doi.org/10.1016/j.sysarc.2017.03.003

[117] Gero Muehl, Matthias Werner, Michael A. Jaeger, Klaus Herrmann, and Helge Parzyjegla. 2007. On the Definitions of Self-Managing and Self-Organizing Systems. In *Communication in Distributed Systems - 15. ITG/GI Symposium*. VDE-Verl., Bern, Switzerland, 1–11.

[118] Manuel Müller, Timo Müller, Behrang Ashtari Talkhestani, Philipp Marks, Nasser Jazdi, and Michael Weyrich. 2021. Industrial autonomous systems: a survey on definitions, characteristics and abilities. *at - Automatisierungstechnik* 69, 1 (2021), 3–13. https://doi.org/10.1515/auto-2020-0131

[119] Christian Muller-Schloer. 2004. Organic computing - on the feasibility of controlled emergence. In *Proceedings of the 2nd IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS '04)*. Association for Computing Machinery, New York, NY, USA, 2–5. https://doi.org/10.1145/1016720.1016724

[120] Mohamed Mustafa, Marina Papatriantafilou, Elad M. Schiller, Amir Tohidi, and Philippas Tsigas. 2012. Autonomous TDMA Alignment for VANETs. In *2012 IEEE Vehicular Technology Conference (VTC FALL) (IEEE Vehicular Technology Conference Proceedings)*. IEEE, Quebec City, QC, Canada, 1–5. https://doi.org/10.1109/VTCFall.2012.6399373

[121] Florian Nafz, Frank Ortmeier, Hella Seebach, Jan-Philipp Steghoefer, and Wolfgang Reif. 2009. A generic software framework for role-based Organic Computing systems. In *2009 ICSE Workshop on Software Engineering for Adaptive and Self-managing Systems*. IEEE, Vancouver, BC, Canada, 96–105. https://doi.org/10.1109/SEAMS.2009.5069078

[122] Florian Nafz, Hella Seebach, Jan-Philipp Steghoefer, Simon Baeumler, and Wolfgang Reif. 2010. A Formal Framework for Compositional Verification of Organic Computing Systems. In *Autonomic and Trusted Computing (Lecture Notes in Computer Science, Vol. 6407)*, B. Xie, J. Branke, S. M. Sadjadi,

D. Q. Zhang, and X. S. Zhou (Eds.). Springer, Berlin, Heidelberg, 17–31. https://doi.org/10.1007/978-3-642-16576-4_2

[123] Rajanikanth Kashi Nagaraj and Meenakshi D'Souza. 2021. A verifiable multi-agent framework for dependable and adaptable avionics. *Sadhana-Academy Proceedings in Engineering Sciences* 46, 1 (2021), 46:1–46:27. https://doi.org/10.1007/s12046-020-01538-4

[124] Hiroyuki Nakagawa, Akihiko Ohsuga, and Shinichi Honiden. 2010. Cooperative Behaviors Description for Self-* Systems Implementation. In *Advances in Practical Applications of Agents and Multiagent Systems (Advances in Intelligent and Soft Computing, Vol. 70)*, Y. Demazeau, F. Dignum, J. M. Corchado, and J. B. Perez (Eds.). Springer, Berlin, Heidelberg, 69–74. https://doi.org/10.1007/978-3-642-12384-9_9

[125] Leandro Navarro. 2009. Managing Democratic Grids: Architecture and Lessons Learnt. In *Grids Meet Autonomic Computing Workshop - GMAC 2009*. Association for Computing Machinery, New York, NY, USA, 51–52. https://doi.org/10.1145/1555301.1555309

[126] Manuel Nickschas and Uwe Brinkschulte. 2008. Guiding Organic Management in a Service-Oriented Real-Time Middleware Architecture. In *Software Technologies for Embedded and Ubiquitous Systems, Proceedings (Lecture Notes in Computer Science, Vol. 5287)*, U. Brinkschulte, T. Givargis, and S. Russo (Eds.). Springer, Berlin, Heidelberg, 90–101. https://doi.org/10.1007/978-3-540-87785-1_9

[127] Rafael Nogueras and Carlos Cotta. 2018. On the Use of Self-* Island-based Evolutionary Computation Methods on Complex Environments. *Computer Science and Information Systems* 15, 3 (2018), 733–750. https://doi.org/10.2298/CSIS180115032N

[128] Steve Nordstrom, Abhishek Dubey, Turker Keskinpala, Sandeep Neema, and Theodore Bapty. 2011. Autonomic Healing of Model-Based Systems. *Journal of Aerospace Computing Information and Communication* 8, 4 (2011), 87–99. https://doi.org/10.2514/1.31940

[129] Tomasz Nowicki, Mark S. Squillante, and Chai W. Wu. 2005. Fundamentals of dynamic decentralized optimization in autonomic computing systems. In *Self-star Properties in Complex Information Systems: Conceptual and Practical Foundations*. Lecture Notes in Computer Science, Vol. 3460. Springer, Berlin, Heidelberg, 204–218. https://doi.org/10.1007/11428589_14

[130] Pelayo Nuno, Juan C. Granda, Francisco J. Suarez, and Daniel F. Garcia. 2013. Self-* in Multimedia Communication Overlays. *Computer Communications* 36, 7 (2013), 817–833. https://doi.org/10.1016/j.comcom.2012.12.009

[131] Andrea Omicini. 2013. Nature-Inspired Coordination for Complex Distributed Systems. In *Intelligent Distributed Computing VI (Studies in Computational Intelligence, Vol. 446)*, G. Fortino, C. Badica, M. Malgeri, and R. Unland (Eds.). Springer, Berlin, Heidelberg, 1–6. https://doi.org/10.1007/978-3-642-32524-3_1

[132] Mauro Onori, Daniel Semere, and Bengt Lindberg. 2011. Evolvable systems: an approach to self-X production. *International Journal of Computer Integrated Manufacturing* 24, 5, SI (2011), 506–516. https://doi.org/10.1080/0951192X.2011.566282

[133] Mauro Onori, Daniel T. Semere, and Bengt Lindberg. 2009. Evolvable Systems: An Approach to Self-X Production. In *Proceedings of the 6th CIRP-Sponsored International Conference on Digital Enterprise Technology*, George Q. Huang, K. L. Mak, and Paul G. Maropoulos (Eds.). Advances in Intelligent and Soft Computing, Vol. 66. Springer Berlin Heidelberg, Berlin, Heidelberg, 789–802. https://doi.org/10.1007/978-3-642-10430-5_61

[134] Olga Pacheco. 2004. Autonomy in an Organizational Context. In *Agents and computational autonomy*, Matthias Nickles (Ed.). Hot topics, Vol. 2969. Springer, Berlin and Heidelberg, 195–208. https://doi.org/10.1007/978-3-540-25928-2_16

[135] Raja Parasuraman. 2000. Designing automation for human use: empirical studies and quantitative models. *Ergonomics* 43, 7 (2000), 931–951. https://doi.org/10.1080/001401300409125

[136] Joao Alvarez Peixoto, Carlos Eduardo Pereira, and Jose Barata de Oliveira. 2016. Functionalities of multi-agent systems in Programmable Logic Controllers. *IFAC Papersonline* 49, 30 (2016), 60–64. https://doi.org/10.1016/j.ifacol.2016.11.126

[137] Mariusz Pelc. 2016. Analysis of Policy-Based Systems with AGILE Policies Using Petri Nets. *International Journal of Software Engineering and Knowledge Engineering* 26, 8 (2016), 1255–1283. https://doi.org/10.1142/S0218194016500443

[138] Alessandro Pinto. 2019. An Open and Modular Architecture for Autonomous and Intelligent Systems. In *2019 IEEE International Conference on Embedded Software and Systems (ICESS)*. IEEE, Las Vegas, NV, USA, 1–8. https://doi.org/10.1109/ICESS.2019.8782512

[139] Alessandro Pinto. 2021. Requirement Specification, Analysis and Verification for Autonomous Systems. In *58th ACM/IEEE Design Automation Conference (DAC)*. IEEE, San Francisco, CA, USA, 1315–1318. https://doi.org/10.1109/dac18074.2021.9586208

[140] Jesus M. T. Portocarrero, Flavia C. Delicato, Paulo F. Pires, Nadia Gamez, Lidia Fuentes, David Ludovino, and Paulo Ferreira. 2014. Autonomic Wireless Sensor Networks: A Systematic Literature Review. *Journal of Sensors* 2014 (2014), 1–13. https://doi.org/10.1155/2014/782789

[141] Harald Psaier and Schahram Dustdar. 2011. A survey on self-healing systems: approaches and systems. *Computing* 91, 1 (2011), 43–73. https://doi.org/10.1007/s00607-010-0107-y

[142] H. P. Raghunandan and Smita Bedekar. 2011. Multiple Linear Regression based Parameter Influencer model of a Self-Healing Network. In *Proceedings of the 2011 3rd International Workshop on Security and Communication Networks (IWSCN 2011)*, P. Bours (Ed.). IEEE, Gjovik, Norway, 45–52. https://doi.org/10.1109/IWSCN.2011.6827716

[143] Binu Rajan and Vinod Chandra. 2017. ABC Metaheuristic Based Optimized Adaptation Planning Logic for Decision Making Intelligent Agents in Self Adaptive Software System. In *Data Mining and Big Data, DMBD 2017 (Lecture Notes in Computer Science, Vol. 10387)*, Y. Tan and H. Takagi (Eds.). Springer, Cham, 496–504. https://doi.org/10.1007/978-3-319-61845-6_49

[144] Edgar Chacon Ramirez, Juan Cardillo Albarran, and Luis Alberto Cruz Salazar. 2020. The Control of Water Distribution Systems as a Holonic System. In *Service Oriented, Holonic and Multi-agent Manufacturing Systems for Industry of the Future (Studies in Computational Intelligence, Vol. 853)*, T. Borangiu, D. Trentesaux, P. Leitao, Boggino, AG, and V. Botti (Eds.). Springer, Cham, 352–365. https://doi.org/10.1007/978-3-030-27477-1_27

[145] Franz Rammig and Katharina Stahl. 2016. Online behavior classification for anomaly detection in self-x real-time systems. *Concurrency and Computation-practice & Experience* 28, 14 (2016), 3751–3772. https://doi.org/10.1002/cpe.3567

[146] Franz Rammig, Katharina Stahl, and Gavin Vaz. 2013. A Framework for Enhancing Dependability in Self-x Systems by Artificial Immune Systems. In *2013 IEEE 16th International Symposium on Object/component/service-oriented Real-time Distributed Computing (ISORC) (International Symposium on Object Component Service-Oriented Real-Time Distributed Computing)*. IEEE, Paderborn, Germany, 1–10. https://doi.org/10.1109/ISORC.2013.6913240

[147] Luis Ribeiro, Jose Barata, and Armando Colombo. 2008. MAS and SOA: A Case Study Exploring Principles and Technologies to Support Self-Properties in Assembly Systems. In *SASOW 2008: Second IEEE International Conference on Self-adaptive and Self-organizing Systems Workshops, Proceedings*, G. D. Serugendo (Ed.). IEEE, Venice, Italy, 192–197. https://doi.org/10.1109/SASOW.2008.41

[148] Imre Rudas and Janos Fodor. 2008. Intelligent systems. *International Journal of Computers, Communications & Control (IJCCC)* 3 (2008), 132–138.

[149] S. Saidi, D. Ziegenbein, J. V. Deshmukh, and R. Ernst. 2022. Autonomous Systems Design: Charting a New Discipline. *IEEE Design & Test* 39, 1 (2022), 8–23. https://doi.org/10.1109/MDAT.2021.3128434

[150] Mazeiar Salehie and Ladan Tahvildari. 2005. Autonomic computing. *ACM SIGSOFT Software Engineering Notes* 30, 4 (2005), 1–7. https://doi.org/10.1145/1082983.1083082

[151] Mazeiar Salehie and Ladan Tahvildari. 2009. Self-Adaptive Software: Landscape and Research Challenges. *ACM Transactions on Autonomous and Adaptive Systems* 4, 2 (2009), 1—-42. https://doi.org/10.1145/1516533.1516538

[152] Mazeiar Salehie and Ladan Tahvildari. 2012. Towards a goal–driven approach to action selection in self–adaptive software. *Software: Practice and Experience* 42, 2 (2012), 211–233. https://doi.org/10.1002/spe.1066

[153] Saraiva de Sousa, Nathan Franklin and Christian Esteve Rothenberg. 2021. CLARA: Closed Loop-based Zero-touch Network Management Framework. In *2021 IEEE Conference on Network Function Virtualization and Software Defined Networks (IEEE NFV-SDN)*, L. Horner, K. Tutschku, M. Gharbaoui, A. DeLaOliva, C. Contoli, and H. Parzyjegla (Eds.). IEEE, Heraklion, Greece, 110–115. https://doi.org/10.1109/NFV-SDN53031.2021.9665048

[154] Benjamin Satzger, Florian Mutschelknaus, Faruk Bagci, Florian Kluge, and Theo Ungerer. 2009. Towards Trustworthy Self-optimization for Distributed Systems. In *Software Technologies for Embedded and Ubiquitous Systems, Proceedings (Lecture Notes in Computer Science, Vol. 5860)*, S. Lee and P. Narasimhan (Eds.). Springer, Berlin, Heidelberg, 58–68. https://doi.org/10.1007/978-3-642-10265-3_6

[155] Benjamin Satzger, Andreas Pietzowski, Wolfgang Trumler, and Theo Ungerer. 2007. Variations and evaluations of an adaptive accrual failure detector to enable self-healing properties in distributed systems. In *Architecture of Computing Systems - ARCS 2007, Proceedings (Lecture Notes in Computer Science, Vol. 4415)*, P. Lukowicz, L. Thiele, and G. Troster (Eds.). Springer, Berlin, Heidelberg, 171–184. https://doi.org/10.1007/978-3-540-71270-1_13

[156] Julia Schaumeier, Jeremy Pitt, and Giacomo Cabri. 2012. A Tripartite Analytic Framework for Characterising Awareness and Self-Awareness in Autonomic Systems Research. In *2012 IEEE Sixth International Conference on Self-adaptive and Self-organizing Systems Workshops (SASOW) (International Conference on Self-Adaptive and Self-Organizing Systems)*. IEEE, Lyon, France, 157–162. https://doi.org/10.1109/SASOW.2012.35

[157] H. Schmeck. 2005. Organic computing - a new vision for distributed embedded systems. In *Proceedings / Eighth IEEE International Symposium on Object-Oriented Real-Time Distributed Computing, ISORC 2005*, Arif Ghafoor (Ed.). IEEE Computer Society, Los Alamitos, Calif., 201–203. https://doi.org/10.1109/ISORC.2005.42

[158] Hartmut Schmeck and Christian Mueller-Schloer. 2008. A characterization of key properties of environment-mediated multiagent systems. In *Engineering Environment-mediated Multi-agent Systems (Lecture Notes in Artificial Intelligence, Vol. 5049)*, D. Weyns, S. A. Brueckner, and Y. Demazeau (Eds.). Springer, Berlin, Heidelberg, 17–38. https://doi.org/10.1007/978-3-540-85029-8_3

[159] Hartmut Schmeck, Christian Mueller-Schloer, Emre Cakar, Moez Mnif, and Urban Richter. 2010. Adaptivity and Self-Organization in Organic Computing Systems. *ACM Transactions on Autonomous and Adaptive Systems* 5, 3, SI (2010), 5–37. https://doi.org/10.1145/1837909.1837911

[160] Connor Schoenberner and Sven Tomforde. 2022. Deep Reinforcement Learning with a Classifier System - First Steps. In *Architecture of Computing Systems, ARCS 2022 (Lecture Notes in Computer Science, Vol. 13642)*, M. Schulz, C. Trinitis, N. Papadopoulou, and T. Pionteck (Eds.). Springer, Cham, 256–270. https://doi.org/10.1007/978-3-031-21867-5_17

[161] Christian Schuck, Stefan Lamparth, and Juergen Becker. 2007. artNoC - A novel multi-functional router architecture for organic computing. In *2007 International Conference on Field Programmable Logic and Applications, Proceedings, Vols 1 and 2 (International Conference on Field Programmable and Logic Applications)*, K. Bertels, W. Najjar, A. VanGenderen, and S. Vassiliadis (Eds.). IEEE, Amsterdam, Netherlands, 371–376. https://doi.org/10.1109/FPL.2007.4380674

[162] Michael L. Scott. 2011. Synchronization. In *Encyclopedia of Parallel Computing*, David Padua (Ed.). Springer US, Boston, MA, 1989–1996. https://doi.org/10.1007/978-0-387-09766-4_252

[163] Hella Seebach, Frank Ortmeier, and Wolfgang Reif. 2007. Design and construction of organic computing systems. In *2007 IEEE Congress on Evolutionary Computation*. IEEE Service Center, Piscataway, NJ, 4215–4221. https://doi.org/10.1109/CEC.2007.4425021

[164] Manolis Sifalakis, Michael Fry, and David Hutchison. 2010. Event Detection and Correlation for Network Environments. *IEEE Journal on Selected Areas in Communications* 28, 1 (2010), 60–69. https://doi.org/10.1109/JSAC.2010.100107

[165] Taranjeet Singh and Avadhesh Kumar. 2016. Survey on Characteristics of Autonomous System. *International Journal of Computer Science and Information Technology* 8, 2 (2016), 121–128. https://doi.org/10.5121/ijcsit.2016.8210

[166] Sebastian Smolorz and Bernardo Wagner. 2011. Self-organised Distribution of Tasks Inside a Networked Robotic System. In *ICINCO 2011: Proceedings of the 8th International Conference on Informatics in Control, Automation and Robotics, Vol 1*, J. L. Ferrier, A. Bernard, O. Gusikhin, and K. Madani (Eds.). SciTePress, Noordwijkerhout, Netherlands, 235–238. https://doi.org/10.5220/0003572402350238

[167] Jens Steiner and Matthias Hagner. 2007. Runtime analysis of a self-adaptive hard real-time robotic control system. In *Fourth IEEE International Workshop on Engineering of Autonomic & Autonomous Systems, Proceedings*, R. Sterritt, M. G. Hinchey, and T. Bapty (Eds.). IEEE, Tucson, AZ, USA,

53–60. https://doi.org/10.1109/EASE.2007.23

[168] Roy Sterritt. 2005. Autonomic computing. *Innovations in Systems and Software Engineering* 1, 1 (2005), 79–88. https://doi.org/10.1007/s11334-005-0001-5

[169] Roy Sterritt. 2011. Next Generation Autonomic System Environment. *Journal of Aerospace Computing Information and Communication* 8, 4 (2011), 115–126. https://doi.org/10.2514/1.54467

[170] Roy Sterritt, Grainne Garrity, Edward Hanna, and Patricia O'Hagan. 2006. Survivable security systems through autonomicity. In *Innovative concepts for autonomic and agent-based systems (Lecture Notes in Artificial Intelligence, Vol. 3825)*, M. G. Hinchey, P. Rago, J. L. Rash, C. A. Rouff, R. Sterritt, and W. Truszkowski (Eds.). Springer, Berlin, Heidelberg, 379–389. https://doi.org/10.1007/11964995_34

[171] Roy Sterritt and Michael Hinchey. 2005. SPAACE:: Self-properties for an autonomous & autonomic computing environment. In *SERP 2005: Proceedings of the 2005 International Conference on Software Engineering Research and Practice, Vols 1 and 2*, H. R. Arabnia and H. Reza (Eds.). CSREA Press 2005, Las Vegas, Nevada, USA, 3–8.

[172] Roy Sterritt, Christopher Rouff, James Rash, Walter Truszkowski, and Michael Hinchey. 2005. Self-* properties in NASA missions. In *SERP 2005: Proceedings of the 2005 International Conference on Software Engineering Research and Practice, Vols 1 and 2*, H. R. Arabnia and H. Reza (Eds.). CSREA Press 2005, Las Vegas, Nevada, USA, 66–72.

[173] Roy Sterritt, Christopher A. Rouff, Michael G. Hinchey, James L. Rash, and Walt Truszkowski. 2006. Next generation system and software architectures - Challenges from future NASA exploration missions. *Science of Computer Programming* 61, 1 (2006), 48–57. https://doi.org/10.1016/j.scico.2005.11.005

[174] Roy Sterritt, Barry Smyth, and Martin Bradley. 2005. PACT: Personal autonomic computing tools. In *12TH IEEE International Conference and Workshops on the Engineering of Computer-based Systems, Proceedings*, J. Rozenblit, T. ONeill, and J. Peng (Eds.). IEEE, Greenbelt, MD, USA, 519–527. https://doi.org/10.1109/ECBS.2005.54

[175] Daniel Stock, Thomas Bauernhansl, Michael Weyrich, Matthias Feurer, and Rolf Wutzke. 2020. System Architectures for Cyber-Physical Production Systems enabling Self-X and Autonomy. In *2020 IEEE 25th International Conference on Emerging Technologies and Factory Automation (ETFA)*. IEEE, Piscataway, NJ, 148–155. https://doi.org/10.1109/ETFA46521.2020.9212182

[176] Poly Z. H. Sun, Yuguang Bao, Xinguo Ming, and Tongtong Zhou. 2023. Knowledge-Driven Industrial Intelligent System: Concept, Reference Model, and Application Direction. *IEEE Transactions on Computational Social Systems* 10, 4 (2023), 1465–1478. https://doi.org/10.1109/TCSS.2022.3188295

[177] Claudia Szabo, Brendan Sims, Thomas Mcatee, Riley Lodge, and Robert Hunjet. 2021. Self-Adaptive Software Systems in Contested and Resource-Constrained Environments: Overview and Challenges. *IEEE Access* 9 (2021), 10711–10728. https://doi.org/10.1109/ACCESS.2020.3043440

[178] Veronika Szaboova, Csaba Szabo, Zdenek Havlice, and Tomas Guzlej. 2015. A Case Study Comparing Naive Approaches to Self-Reflecting Information System Architecture and Implementation. In *2015 IEEE 13th International Scientific Conference on Informatics*, V. Novitzka, S. Korecko, and A. Szakal (Eds.). IEEE, Poprad, Slovakia, 263–268. https://doi.org/10.1109/Informatics.2015.7377844

[179] Behrang Ashtari Talkhestani, Tobias Jung, Benjamin Lindemann, Nada Sahlab, Nasser Jazdi, Wolfgang Schloegl, and Michael Weyrich. 2019. An architecture of an Intelligent Digital Twin in a Cyber-Physical Production System. *AT-Automatisierungstechnik* 67, 9 (2019), 762–782. https://doi.org/10.1515/auto-2019-0039

[180] Kittikhun Thongpull, Dennis Groben, and Andreas Koenig. 2015. A design automation approach for task-specific intelligent multi-sensory systems - Lab-on-spoon in food applications. *TM - Technisches Messen* 82, 4 (2015), 196–208. https://doi.org/10.1515/teme-2014-0009

[181] Kittikhun Thongpull and Andreas Koenig. 2016. Advance and case studies of the DAICOX framework for automated design of multi-sensor intelligent measurement systems. *TM - Technisches Messen* 83, 4 (2016), 234–243. https://doi.org/10.1515/teme-2015-0117

[182] Marialena Vagia, Aksel A. Transeth, and Sigurd A. Fjerdingen. 2016. A literature review on the levels of automation during the years. What are the different taxonomies that have been proposed? *Applied Ergonomics* 53 (2016), 190–202. https://doi.org/10.1016/j.apergo.2015.09.013

[183] Kyriakos G. Vamvoudakis, Panos J. Antsaklis, Warren E. Dixon, João P. Hespanha, Frank L. Lewis, Hamidreza Modares, and Bahare Kiumarsi. 2015. Autonomy and machine intelligence in complex systems: A tutorial. In *2015 American Control Conference (ACC)*. IEEE, Chicago, IL, USA, 5062–5079. https://doi.org/10.1109/ACC.2015.7172127

[184] Norha M. Villegas, Gabriel Tamura, and Hausi A. Müller. 2017. Chapter 2 - Architecting Software Systems for Runtime Self-Adaptation: Concepts, Models, and Challenges. In *Managing Trade-Offs in Adaptable Software Architectures*, Ivan Mistrik, Nour Ali, Rick Kazman, John Grundy, and Bradley Schmerl (Eds.). Morgan Kaufmann, Boston, 17–43. https://doi.org/10.1016/B978-0-12-802855-1.00002-2

[185] Luis Eduardo Villela Zavala, Alfonso Ordonez Garcia, and Mario Siller. 2019. Architecture and Algorithm for IoT Autonomic Network Management. In *2019 International Conference on Internet of Things (Ithings) and Ieee Green Computing and Communications (Greencom) and Ieee Cyber, Physical and Social Computing (Cpscom) and IEEE Smart Data (SMARTDATA)*. IEEE, Atlanta, GA, USA, 861–867. https://doi.org/10.1109/iThings/GreenCom/CPSCom/SmartData.2019.00155

[186] Alexander von Renteln and Uwe Brinkschulte. 2011. Implementing and evaluating the AHS organic middleware. *Computer Systems Science and Engineering* 26, 6, SI (2011), 519–526.

[187] David Watson and David Scheidt. 2005. Autonomous systems. *Johns Hopkins APL Technical Digest* 26, 4 (2005), 368–376.

[188] Gereon Weiss, Klaus Becker, Benjamin Kamphausen, Ansgar Radermacher, and Sebastien Gerard. 2011. Model-Driven Development of Self-describing Components for Self-Adaptive Distributed Embedded Systems. In *2011 37TH Euromicro Conference on Software Engineering and Advanced Applications (SEAA 2011) (EUROMICRO Conference Proceedings)*, S. Biffl, M. Koivuluoma, P. Abrahamsson, and M. Oivo (Eds.). IEEE, Oulu, Finland, 477–484. https://doi.org/10.1109/SEAA.2011.78

[189] Danny Weyns, M. Usman Iftikhar, Sam Malek, and Jesper Andersson. 2012. Claims and supporting evidence for self-adaptive systems: A literature study. In *2012 ICSE Workshop on Software Engineering for Adaptive and Self-Managing Systems (SEAMS 2012)*, Hausi A. Müller (Ed.). IEEE, Piscataway, NJ, 89–98. https://doi.org/10.1109/SEAMS.2012.6224395

[190] Rolf P. Würtz. 2008. Introduction: Organic Computing. In *Organic Computing*, J. ScottA. Kelso, P. Érdi, K. Friston, H. Haken, J. Kacprzyk, J. Kurths, L. Reichl, P. Schuster, F. Schweitzer, D. Sornette, and Rolf P. Würtz (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 1–6. https://doi.org/10.1007/978-3-540-77657-4_1

[191] Kamaleddin Yaghoobirafi and Eslam Nazemi. 2016. A Self-Adaptive Middleware for Attaining Semantic Self-Interoperation Property. In *2016 IEEE 1st International Workshops on Foundations and Applications of Self* Systems (FAS*W)*, S. Elnikety, P. R. Lewis, and C. Muller-Schloer (Eds.). IEEE, Augsburg, Germany, 293–298. https://doi.org/10.1109/FAS-W.2016.70

[192] Mona A. Yahya, Manal A. Yahya, and Ajantha Dahanayake. 2013. Autonomic Computing: A Framework to Identify Autonomy Requirements. *Procedia Computer Science* 20 (2013), 235–241. https://doi.org/10.1016/j.procs.2013.09.267

[193] Eric Yuan, Naeem Esfahani, and Sam Malek. 2014. A Systematic Survey of Self-Protecting Software Systems. *ACM Transactions on Autonomous and Adaptive Systems* 8, 4 (2014), 17:1–17:41. https://doi.org/10.1145/2555611

[194] Kening Zhang, Navid Khoshavi, Jaafar M. Alghazo, and Ronald F. DeMara. 2015. Organic Embedded Architecture for Sustainable FPGA Soft-Core Processors. In *2015 61st Annual Reliability and Maintainability Symposium (RAMS 2015) (Reliability and Maintainability Symposium)*. IEEE, Palm Harbor, FL, USA, 1–6. https://doi.org/10.1109/RAMS.2015.7105065

[195] Zhongshan Zhang, Keping Long, Jianping Wang, and Falko Dressler. 2014. On Swarm Intelligence Inspired Self-Organized Networking: Its Bionic Mechanisms, Designing Principles and Optimization Approaches. *IEEE Communications Surveys and Tutorials* 16, 1 (2014), 513–537. https://doi.org/10.1109/SURV.2013.062613.00014

[196] Zhang ZhongShan, Huangfu Wei, Long KePing, Zhang Xu, Liu XiaoYuan, and Zhong Bin. 2013. On the designing principles and optimization approaches of bio-inspired self-organized network: a survey. *Science China-information Sciences* 56, 7 (2013), 071301:1–071301:28. https://doi.org/10.1007/s11432-013-4894-6

[197] Pai Zheng, Liqiao Xia, Chengxi Li, Xinyu Li, and Bufan Liu. 2021. Towards Self-X cognitive manufacturing network: An industrial knowledge graph-based multi-agent reinforcement learning approach. *Journal of Manufacturing Systems* 61 (2021), 16–26. https://doi.org/10.1016/j.jmsy.2021.08.002

[198] Haibin Zhu. 2008. Role-Based Systems are Autonomic. In *Proceedings of the seventh IEEE international conference on cognitive informatics*, Y. Wang, D. Zhang, J. C. Latombe, and W. Kinsner (Eds.). IEEE, Stanford, CA, USA, 144–152. https://doi.org/10.1109/COGINF.2008.4639162

A APPENDIX

A.1 Self-X comparison between system classes

No	Name	Autonomous					Autonomic					Adaptive				
		[165]	[118]	[110]	[40]	[41]	[192]	[39]	[168]	[109]	[52]	[184]	[68]	[108]	[152]	[189]
1	self-configuration	X		X			X	X	X	X	X			X	X	
2	self-healing	X		X			X	X	X		X	X		X	X	X
3	self-optimizing	X	X	X			X	X	X	X	X	X		X	X	X
4	self-protecting	X		X			X	X	X		X	X	X	X	X	
5	self-awareness	X						X	X					X	X	
6	self-determination				X											
7	self-managing	X		X	X		X	X	X	X	X	X		X		X
8	self-containedness		X					X								
9	self-directedness		X													
10	self-explanation		X													
11	self-motivation		X													
12	self-assessment		X													
13	self-adaptability		X	X	X	X		X	X	X	X	X	X	X	X	X
14	self-evolution				X											
15	self-repairing					X										
16	self-regulating				X	X				X						
17	self-control				X											
18	self-organization		X		X	X				X			X	X		
19	self-governance		X			X										
20	self-perception		X		X											
21	self-guidance				X											
22	self-calibration		X													
23	self-monitoring					X			X					X		
24	self-prediction									X						
25	self-inspection									X						
26	self-representation													X		
27	self-tuning													X		X
28	self-assembly															X
29	self-improve			X												
30	self-analyze			X												
31	self-recovery			X												
32	self-maintenance					X										
33	self-learning				X								X			

A.2 Extracted Self-X Capabilities from underlying literature

No	Self-x term	Papers	References
1	self-healing	99	[1–4, 7, 9, 11, 12, 17, 19, 22, 25–29, 33, 35–38, 43, 46, 50, 51, 54, 58, 59, 61–64, 67, 71, 73, 74, 79, 82, 83, 85–87, 89–92, 95, 99, 102, 104–106, 111, 114–116, 121, 123, 124, 126–128, 130, 140–142, 144, 146, 147, 151, 153–159, 161, 166, 167, 169–174, 177–179, 181, 185, 186, 188, 193–198]
2	self-organization	98	[3, 9, 11, 15, 17, 19, 22–24, 26–28, 30, 31, 33, 35, 37, 38, 42–44, 46, 48, 54, 58, 59, 61–64, 67, 71–74, 78, 81–84, 86, 92, 93, 95, 97–99, 101, 102, 104–107, 111, 113–116, 120–122, 124–127, 129–132, 136, 140, 142–144, 146, 147, 151, 154–159, 161, 164, 166, 167, 169, 171, 177, 178, 185, 186, 191, 193, 195–197]
3	self-adaption	97	[1–5, 9, 11, 13, 14, 16–19, 22–27, 30, 31, 33, 35, 36, 38, 42, 43, 46, 48–51, 54, 56, 58, 61–63, 67, 71–74, 78, 79, 81, 83, 85, 90, 92, 93, 95, 97, 99, 102, 104, 111, 113–116, 121–124, 127, 130, 132, 136, 140–147, 151, 154, 156, 161, 167, 169–174, 177, 178, 185, 188, 191, 193–195, 197]
4	self-optimization	85	[2, 3, 5, 9, 12, 15, 17, 19, 22–24, 26–29, 35–38, 43, 46, 50, 51, 54, 58, 59, 63, 64, 72, 74, 78, 79, 81, 84–87, 90, 91, 93, 95, 102, 104–107, 111, 113–116, 123, 126, 130, 136, 140–142, 144–146, 151, 153–156, 158, 159, 164, 166, 167, 169–174, 177, 185, 186, 193, 195–198]
5	self-management	69	[3, 9, 12–15, 17, 18, 22, 23, 26–28, 35, 36, 38, 43, 46, 49–51, 54, 58, 61–64, 67, 69, 71, 73, 74, 81, 85–87, 90, 91, 95, 99, 101, 102, 124, 125, 127, 129, 130, 140–142, 151, 153, 158, 159, 166, 167, 169–174, 177, 185, 188, 193, 195, 196, 198]
6	self-configuration	62	[3, 4, 7, 9, 11, 12, 15, 17, 22, 25, 28, 35–38, 42, 46, 48, 61–64, 67, 71, 85, 90, 95, 98, 99, 102, 104, 106, 107, 111, 122, 130, 132, 137, 140, 141, 145–147, 151, 156, 159, 161, 166, 167, 169–174, 185, 193–198]
7	self-protection	55	[3, 12, 13, 17, 26, 27, 29, 31, 35–38, 43, 46, 51, 59, 61–63, 74, 85, 87, 90, 91, 95, 99, 102, 104–106, 111, 114–116, 123, 129, 130, 140–142, 151, 154–156, 167, 169–174, 177, 185, 193, 198]
8	self-awareness	36	[3, 9, 14, 17, 23, 24, 26, 27, 30, 31, 35, 36, 38, 42, 46, 81, 85, 92, 99, 102, 103, 111, 131, 141, 151, 153, 156, 166, 169–174, 193, 198]
9	self-repairing	22	[12, 17, 24, 30, 31, 38, 61–63, 67, 82, 85, 101, 106, 128, 129, 141, 151, 169, 171, 178, 194]
10	self-monitoring	18	[3, 17, 25, 38, 67, 78, 82, 90, 95, 147, 151, 169–174, 181]
11	self-diagnosis	13	[15, 36, 38, 61, 63, 67, 85, 101, 102, 130, 132, 141, 151]
12	self-learning	12	[48, 74, 84, 88, 102, 106, 160, 177, 179–181, 195]
13	self-governing	10	[151, 156, 169–173, 185, 195, 196]
14	self-stabilizing	9	[17, 79, 83, 98, 104, 120, 130, 141, 159]
15	self-tuning	7	[12, 24, 69, 83, 132, 151, 174]
16	self-assembly	7	[31, 61–63, 151, 169, 171]
17	self-calibration	7	[1, 2, 67, 91, 130, 180, 181]
18	self-destruction	7	[90, 130, 169, 171–173, 198]
19	self-directedness	6	[62, 164, 169–171, 173]
20	self-regulating	6	[4, 83, 130, 170, 177, 198]
21	self-similarity	6	[62, 63, 95, 103, 169, 171]
22	self-maintenance	6	[63, 67, 141, 151, 195, 196]
23	self-description	5	[22, 56, 99, 106, 188]
24	self-recovery	5	[102, 130, 151, 195, 196]

(Continued next page)

No	Self-x term	Papers	References
25	self-expression	4	[26, 27, 30, 156]
26	self-evolving	4	[5, 132, 195, 196]
27	self-synchronization	4	[19, 181, 195, 196]
28	self-replication	3	[61, 63, 106]
29	self-balancing	3	[127, 195, 196]
30	self-sustaining	3	[128, 141, 144]
31	self-assessment	3	[22, 69, 130]
32	self-representation	3	[43, 141, 156]
33	self-improving	3	[74, 78, 160]
34	self-generated	3	[4, 106, 185]

A.3 Synonyms of self-x capabilities

No	Capabilities	Explanation	Ref.
1	self-healing, self-repairing, self-stabilizing, self-recovery	Self-healing describes the ability of a system to detect system errors or faulty components, diagnose and correct them. Therefore, it has a comparable meaning to the terms self-repairing, self-recovery and self-stabilizing which are also concerned with reaching a safe state after fault.	[51, 61, 63, 130, 141]
2	self-optimization; self-tuning; self-balancing	Self-optimization means that the system can improve itself regarding certain goals. The property has the same meaning than self-tuning, which is actively used as a synonym in the references. A special form of self-optimization is self-balancing which is being used in the context of load balancing of interactive network elements to optimize system load and can therefore be seen as another synonym in specialized fields.	[20, 94, 151, 165, 196]
3	self-management; self-maintenance; self-sustaining	The goal of autonomic computing is self-management and therefore to provide a zero cost maintenance and highly reliable system to end users encompassing self-CHOP properties. The term self-maintenance is being used in more specialized fields such as industry or networking, but is often named together with self-CHOP capabilities and basically follows the same goals as self-management. Self-sustaining is actively used as a synonym for self-managing systems in the related references.	[20, 67, 91, 128, 130, 141, 195]
4	self-assembly; self-similarity;	Self-assembly describes systems where individual elements can self-assemble into complex forms to achieve higher-level objectives. A specialized form of self-assembly is self-similarity, which describes components created from similar components. Both properties are named are especially in the context of swarm intelligence and self-organizing systems.	[95, 169, 171]
5	self-description; self-representation	Both terms describe the ability of a system to describe itself and its components.	[43, 141]
6	self-evolving; self-improving	Self-evolving or self-improving systems assist in the evolution of components that adapt to changing conditions of operation and self-improve function over time. These terms are used as synonyms in the related references.	[74, 133, 160]
7	self-organization; self-expression	self-organization describes systems that are capable of rearranging their own organization or structure to achieve a global goal based on local actions. The property is especially used in the context of organic computing and is closely related to swarm intelligence and emergent behavior. The term self-expression is described as a synonym of self-organization that is used in the field of autonomic computing. It describes components that can dynamically change their coordination pattern during run-time execution and therefore modify the internal organization of a system.	[26, 27, 30]
8	self-configuration; self-calibration	self-configuration describes the system ability to configure and re-configuring automatically, for example, for the purpose of self-optimization or self-healing. Self-calibration is a synonym of self-configuration as it describes the re-configuration of paramters with respect to a calibration standard or optimal performance.	[11, 51, 61, 91, 130]
9	self-assessment; self-diagnosis	Self-assessment describes the ability to automatically identify system failure, risks and problem-solving capacity as a basis for autonomic self-healing and self-protection or in wider terms changes in self-adaptive systems. Therefore, this property includes capabilities to diagnose errors or faults and can be seen as a broader term for self-diagnosis that is used as a basis for self-healing.	[22, 118, 130]

A.4 Harmonized list of self-x characteristics

No	Self-x term	Synonyms	Papers	Meaning	Ref.
1	self-healing	self-repairing, self-stabilizing, self-recovery	135	System detects system errors or faulty components, diagnoses and corrects them	[51, 61, 63]
2	self-organization	self-expression	102	System rearranges or adapts its own structure to achieve a global goal based on local actions	[17, 38, 54]
3	self-adaption		97	Systems adjusts itself in response to changing conditions to maintain or improve function	[20, 61, 143, 151]
4	self-optimization	self-tuning, self-balancing	95	System can improve itself regarding certain goals	[130, 141, 198]
5	self-management	self-maintenance, self-sustaining	78	System manages a substantial amount of computing functions to relieve users from management activities	[17, 20, 61]
6	self-configuration	self-calibration	69	System is able to configure and re-configuring automatically	[46, 130, 169]
7	self-protecting		55	System is capable of defending itself and its components from attacks	[171]
8	self-awareness		36	System is aware of its internal state and behavior	[3, 169, 171]
9	self-monitoring		18	System observes its state and detects changing circumstances	[90, 171, 174]
10	self-assessment	self-diagnosis	16	System automatically assesses system failure, risks and problem-solving capacity	[22, 118, 130]
11	self-assembly	self-similarity	13	System allows individual elements to self-assemble into complex forms to achieve higher-level objectives	[61, 63, 171]
12	self-learning		12	System learns from experiences rather than being pre-programmed	[74, 106, 195]
13	self-governing		10	System takes responsibility, performs tasks despite uncertain environment and makes decisions independently from external interventions	[168, 171]
14	self-description	self-representation	8	System can describe itself and its components	[99, 106, 188]
15	self-evolving	self-improving	7	System assists in the evolution of components or itself to improve its operation over time	[94, 133]
16	self-destruction		7	System can abort execution or exclude components when a risk or safety problem is detected for the system or its environment	[130, 169]
17	self-directedness		6	System is able to plan goals and actions independently to complete a given task	[118, 169, 173]
18	self-regulating		6	Independent execution of actions to fulfill a repetitive task	[41, 76, 130]
19	self-synchronization		4	System is able to synchronize the behavior of resources in a group with a collective goal	[61, 130, 141, 151]
20	self-replication		3	System is able to duplicate oneself or individual system components	[20, 61]
21	self-generated		3	System finds its own representations and goals according to some specific task	[4, 106, 185]

Received 01 June 2024; revised 12 March 2009; accepted 5 June 2009