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

1st Miadowicz Inga
Solar Research
Deutsches Zentrum für Luft- und Raumfahrt (DLR)
Cologne, Germany
0009-0004-9766-8804

2nd Maldonado Quinto Daniel
Solar Research
Deutsches Zentrum für Luft- und Raumfahrt (DLR)
Cologne, Germany
0000-0003-2929-8667

3rd Pitz-Paal Robert
Solar Research
Deutsches Zentrum für Luft- und Raumfahrt (DLR)
Cologne, Germany
0000-0002-3542-3391

4rd Felderer Michael
Software Technologies
Deutsches Zentrum für Luft- und Raumfahrt (DLR)
Cologne, Germany
0000-0003-3818-4442

***Abstract*—This article presents a systematic literature review on how self-x capabilities are used to characterize autonomous and similar system classes, including automated, intelligent, adaptive, organic, and autonomic systems. In light of hundreds of self-x characteristics that have evolved into overused buzzwords with no consistent, cross-domain agreement on their precise meaning, we analyze 324 articles from current research state combining manual and automated search strategies via the quasi-gold standard approach. From these, we automatically extract 304 distinct self-x expressions, filter out uncommon terms, clarifying their meanings and possible synonym relationships, and finally present a condensed set of 35 widely used self-x capabilities. Building on this harmonized list and on a model introduced in our previous article on system class relationships, we propose a conceptual self-x characterization model that assigns self-x properties to autonomous and related self-x system classes at the levels of class goals, major capabilities, and supporting skills. The resulting catalog and alignment model provide a common terminology and conceptual foundation that make self-x characterizations more comparable, support cross-domain reuse of results, and offer a structured basis for future discussions and standardization efforts on autonomy and related self-x system classes.**

***Index Terms*—Self-X, Adaptive Systems, Autonomic Systems, Intelligent Systems, Organic Systems, Autonomous Systems**

I. Introduction

The notion of autonomous systems has been a topic of discussion in society, science, and industry for centuries. With recent technological progress and developments in areas such as self-driving cars, the realization of autonomous systems now seems within reach. Despite the growing discourse on autonomous systems [35], there is still a lack of standards and a missing common understanding of the details of system autonomy across application domains [27].

In research, many authors use self-x capabilities to characterize autonomous systems such as self-management, self-organization, or self-protection [2]. 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 [44]. Because the concept of self-x capabilities is also not yet standardized, the literature has evolved a large number of self-x properties, with the result that authors across domains use different sets of self-x terms 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 systems difficult to compare between scenarios, authors, and domains. As a consequence, it remains challenging to identify a coherent set of self-x properties that can be used to characterize system autonomy and to assess whether a system qualifies as autonomous from a functional perspective.

What intensifies the fuzzy concept of autonomous systems and their self-x capabilities is the description of similar ideas by different classes of systems, such as automated, intelligent, adaptive, autonomic, and organic systems [27]. Analogously to the characterization of autonomous systems, self-x capabilities are also used to describe these system classes, which are often collectively referred to as self-x systems [6], [45]. Although these system classes have different names, their self-x capabilities can be identical, similar but differently worded, partially overlapping, or completely distinct from others depending on the authors and domain. This leads to a multitude of related system concepts with similar characterizations that are hard to distinguish and compare.

This article is motivated by the lack of a consolidated, cross-domain understanding of self-x capabilities and their role in characterizing autonomous and related system classes. To address this gap, we conduct a systematic literature review on self-x characteristics that (1) extracts the most frequently used self-x terms in the recent literature, together with their meanings and synonym relationships, to derive a condensed list of self-x capabilities, and (2) assigns these capabilities to system classes in a consistent characterization model of self-x system classes. The resulting self-x catalog and characterization model provide a common terminology and conceptual foundation that supports more comparable system descriptions, facilitates cross-domain reuse of results, and can inform future standardization efforts around autonomy and self-x systems.

Basically, we address two research questions that are answered in the course of the study:

1) Which self-x properties dominate the literature on system autonomy and related classes, and how can they be harmonized into a condensed list?
2) How can autonomous systems and related system classes be characterized in a consistent model based on these self-x properties?

The outline of this work is divided into background, related work, method, results, and conclusion. The paper starts with a background on self-x characteristics (see Section II). Then related work (see Section III) and the methodology of the study are discussed (see Section IV). Afterwards, the results of the work are presented with respect to a condensed summary of common self-x terms used in the current research state and a conceptual self-x characterization model for self-x system classes (see Section V). The closing chapter summarizes the results and explicitly answers the identified research questions (see Section VI).

II. Background

Self-x capabilities are used to describe system autonomy and the vision of related system concepts such as automated, autonomic, or adaptive systems [45], [46]. Thereby, Horn proposed eight key elements of autonomic computing, which are widely accepted as a common ground for characterizing autonomic self-managing systems as well as other self-x system classes [26]. 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 [44], [45]. The main characteristics are also known as self-CHOP capabilities [44]:

- *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 [26], [44]. 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 [45].

With growing research interest in autonomous systems and related system classes, the number of self-x characteristics grows steadily, as many authors introduce new capabilities tailored to their specific context and requirements. As a result, there are hundreds of self-x terms used in the literature [6], which can be organized hierarchically into major, minor, and supportive subproperties [20], [37]. Thereby, autonomous and other self-x system classes are frequently characterized using identical or at least closely related sets of self-x properties, often centered around self-CHOP [11], [16], [31], [34], [42], [44], [45], [51], [52]. Some authors extend these characteristics with additional capabilities [11], [17], [31], [34], [44], [50] or propose partially overlapping [23], [51] or entirely new self-x collections [12], [13], [32], [35], leading to a mixture of shared, overlapping, and distinct characterizations between system classes.

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 stems from Singh and Kumar, who examine autonomous systems in general, treating the terms autonomic and autonomous systems as synonymous in their article [44]. The other is concerned with an autonomous industrial system and has been suggested by Müller et al. [35]. While Sing and Kumar use the main self-CHOP capabilities inspired by Horn [26], 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 Fig. 1):

- *Different wording*: 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:* Some terms can be found in 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:* Some attributes cannot be matched with the other model at all, such as open, anticipatory, or learning abilities.

In summary, there is still no consent in the literature on a dedicated set of self-x capabilities that generally describe autonomous systems and 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 [8], [21]. As

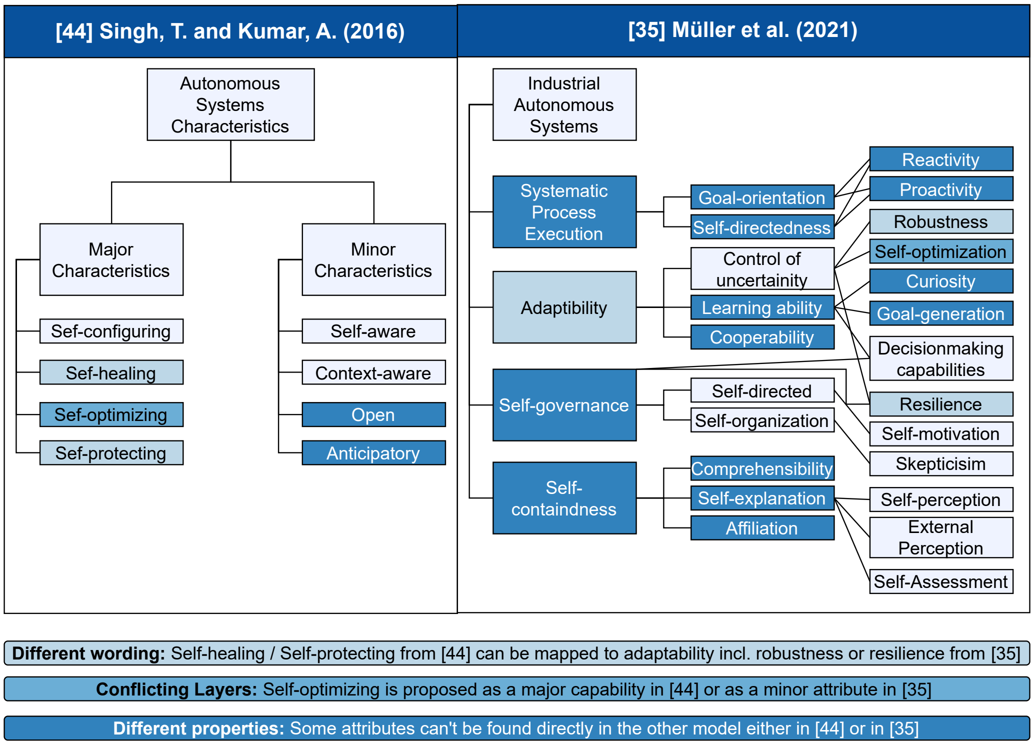

Fig. 1. An example of conflicts between autonomous self-x characterization models on the basis of [35] and [44].

a consequence, proposed self-x characterizations are difficult to compare and use, because they rely on heterogeneous terminology, conflicting structuring into major and minor properties, and divergent skill sets in general. This complicates the characterization of autonomous and other self-x system classes and makes it difficult both to distinguish system classes based on their capabilities and to unambiguously categorize individual system scenarios with one or more system classes.

III. Related Work

Prior work related to it falls into two categories: (i) self-x models and prior reviews informing our catalog of properties, and (ii) studies on system class relationships structuring our characterization model.

With respect to self-x capability models, Horn has proposed one of the first collections of self-x capabilities to characterize autonomic systems (see Section II), which has since become a de facto reference model for many follow-up approaches. Most subsequent papers suggest self-x capabilities to characterize a certain system class either by drawing on general system theory [12], [13], [23], [42], [50], by deducing skills from a specific application domain such as automotive, networking, or industrial automation [32], [35], [37], [47], or by extending and refining previously proposed self-x characterization models, in particular Horn's autonomic computing vision [11], [16], [31], [34], [44], [45], [51], [52].

Only one article from Brooks and Roy deals with a systematic review of the self-x requirements of self-engineering systems [8]. 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 a single system class limited to material and hardware systems and leaves out other system classes and types of system, such as software, control, or operating systems.

Two other studies by Berns and Ghosh [6] and by Nuno et al. [37] investigate the relationship and meaning of different self-x properties. Berns and Ghosh introduce a general template for defining self-x properties, use it to offer more precise definitions of existing self-x terms, and analyze the inter-relationships among these properties, thereby addressing ambiguities in the interpretation of self-x concepts. Nuno et al. study self-x in multimedia communication overlays and derive a taxonomy of self-management characteristics and subproperties implemented in such systems, which helps to structure and relate different self-x mechanisms in a concrete application domain. Both works provide a valuable basis for clarifying the meanings of self-x terms and recognizing synonymous or closely related expressions, but they focus either on formalization in a specific technical model or on a single domain rather than on a broad, cross-domain view of self-x properties, and neither map these properties to different types of system classes.

Related work to our study also includes articles that interrelate self-x system classes, providing essential knowledge of their relationships that enables us to map self-x terms to specific classes and develop a conceptual model characterizing each class based on self-x properties. A recent systematic literature review by Miadowicz et al. [33] examines the intersection of system classes including autonomous, adaptive, autonomic, organic, intelligent and automated systems, clarifying their relationships and overlaps. Unlike that study, which focuses on comparing and interrelating the system classes themselves, our research targets their characterization based on self-x capabilities. However, we use the work of Miadowicz et al. as a baseline for our conceptual model and build upon their model of system class relationships to characterize each class with self-x properties.

In summary, existing research provides important foundations, including early autonomic computing models, domain- and class-specific self-x taxonomies, domain-focused reviews of self-x requirements, and analyzes of self-x meanings and system class relationships. However, there is still no systematic, cross-domain literature review that consolidates self-x terminology across autonomous and related system classes. Consequently, 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, domain-agnostic review of self-x capabilities is still missing that identifies popular terms, specifies their meanings, harmonizes their wording, and maps them consistently to autonomous and related system classes to obtain coherent self-x characterizations.

IV. METHOD

We conduct a systematic literature review to develop fundamental theories for the characterization of autonomous and related self-x systems, following the five-phase framework of Brocke et al. [28]: review scope definition, topic conceptualization, literature search, analysis and synthesis, and research agenda construction. The review scope is specified using Cooper's taxonomy [14], focusing on theoretical work about how self-x capabilities are used to characterize autonomous and related system classes, with the goal of integrating prior literature and clarifying divergent terminologies and conceptualizations. To conceptualize the topic, we derive a search space with two building blocks: (1) names of the system classes of interest including automated, intelligent, adaptive, organic, autonomic and autonomous systems and (2) general terms for self-x capabilities such as *self-x* or another word form for the term such as *self-properties* or *selfware* as we are interested in 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. Putting all together, the search term that we defined and reference here in the advanced search syntax of the bibliographic database Scopus can be found in equation (1).

$$\begin{aligned} &TITLE\text{–}ABS\text{–}KEY((\text{"}autonomous\text{"}\ OR\\ &\text{"}autonomic\text{"}\ OR\ \text{"}adaptive\text{"}\ OR\ \text{"}automated\text{"}\\ &OR\ \text{"}automatic\text{"}\ OR\ \text{"}intelligent\text{"}\ OR\ \text{"}organic\text{"})\\ &AND\ (\{self\text{–}x\}\ OR\ \{self\text{–}properties\}\\ &OR\ \{self\text{–}star\}\ OR\ \{self\text{–}system\}\ OR\\ &\{selfware\}\ OR\ \{self\text{–}*\}\ OR\ \{self\text{–}\star\})) \end{aligned} \tag{1}$$

The literature search combines manual and automated strategies using a quasi-gold standard approach by first identifying a balanced reference set of key papers via venue-based search and snowballing, and then iteratively refining an advanced search string until at least 80% of these reference papers are retrieved across the selected databases [53]. Automated searches are performed in ACM Digital Library, Web of Science, and Scopus, which together provide broad coverage of relevant literature [48]; database-specific syntax and an additional open-source post-processing tool[1] are used to properly capture literal self-x expressions in exported BibTeX entries based on regular expressions. We apply exclusion criteria requiring English language, article or proceeding type, publication year greater than 2000, computer science domain, and final publication status, generating 431 automatically retrieved records that are merged with manually identified references. All records are imported into Citavi, where duplicates and near-duplicates, for example, conference and journal versions of essentially the same work, are removed in a manual screening step, resulting in a final corpus of 324 full-text articles that are downloaded via Citavi's full-text feature.

Based on this corpus, we proceed with the research agenda in several steps. First, we automatically extract self-x terms from all articles using an open-source software tool that we specifically developed to search downloaded full-text articles with regular expressions[2]. Next, we manually define the meaning of each extracted self-x term based on its usage in the linked references. Furthermore, we identify synonym relationships where different terms describe essentially the same function, as well as self-x terms to exclude from further consideration, since they describe narrowly domain-specific capabilities that do not generalize across autonomous systems. We then relate each remaining term to one or more system classes and, where appropriate, to other self-x terms, classifying them as goals, major capabilities, or minor skills following common hierarchical structures in self-x models (see Section III). Finally, we present a condensed set of common self-x terms with clearly specified meanings and propose a conceptual self-x characterization model for self-x system classes, which provides a unified, theory-based foundation for characterizing autonomous and related self-x systems across domains.

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

V. RESULTS

In addition to the current state of research, this section answers the identified research questions. Therefore, the self-x characteristics of autonomous and related system classes are extracted, defined, and harmonized, 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 self-x characterization model for autonomous systems and related system classes.

A. Harmonization of self-x characteristics

To answer the first 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.

First, the software tool referenced in Section IV is used to extract and count all self-x usages in the selected studies. Of the 324 articles, 315 papers contain at least one explicit self-x expression, and overall self-x properties appear more than 12,000 times, confirming the widespread use of self-x terminology to characterize autonomous and related system classes. After unifying different word forms with their nominal variants, for example self-calibrating and self-calibration, and ensuring that each distinct expression is counted only once per paper, we obtain a total of 304 different self-x capabilities mentioned in the corpus. To focus the analysis on terminology with broad support, we restrict the harmonized list to characteristics that are mentioned by at least five articles, which reduces the set to 39 self-x properties; these 39 terms, together with their synonyms, meanings and file counts, are provided in tabular form in Table I, while additional information on tool configuration, original results, and further details such as the full list of references is available in the supplementary material of this work.

The most frequently occurring self-x capability in the corpus is self-healing. It characterizes systems that detect faults or abnormal conditions, diagnose underlying causes, and autonomously recover, for example, by reconfiguring themselves or replacing malfunctioning components. In many articles, self-healing is presented together with the other self-CHOP capabilities as part of the canonical self-management portfolio of autonomic systems. Since these self-CHOP skills are widely used beyond autonomic computing, in particular to describe adaptive, organic, and autonomous systems, both self-management and individual CHOP capabilities are among the most popular self-x descriptors in our dataset.

In terms of the number of articles in which the terms appear, self-healing is closely followed by self-organization and self-adaptation. Self-organization describes systems that autonomously rearrange or adapt their internal structure based on local interactions to achieve global objectives, a concept that is especially prominent in organic systems, where it is frequently stated as a central system goal for large ensembles of cooperating computing units. Self-adaptation refers to systems that adjust themselves in response to changing conditions, evolving goals, or performance deviations to maintain or improve their functionality. While self-adaptation is a defining objective for self-adaptive systems, it is also often attributed to autonomic, organic, and autonomous systems, underlining its role as a cross-cutting self-x capability across several system classes [33].

During the manual consolidation of the extracted terms, we identified three pairs of genuine synonyms: self-optimization and self-tuning, self-recovery and self-stabilizing, as well as self-improving and self-evolving. The first pair, self-optimization and self-tuning, is explicitly described in the literature as a capability that continuously monitors system behavior against a performance baseline and autonomously fine-tunes parameters, redistributes workloads, or reconfigures components at runtime to maximize efficiency, resource utilization, and quality of service without human intervention.

Similarly, self-recovery and self-stabilizing are treated interchangeably in the literature as the automatic response of a system facing failures to achieve a steady state. Both terms describe a system's inherent ability to naturally converge to a safe, legal execution state after the occurrence of transient faults, without necessarily diagnosing the specific root cause. At this point, it is important to clarify why self-repairing is not considered a synonym for self-stabilizing and its equivalent term self-recovery. Self-stabilization is an underlying algorithmic property guaranteeing that a system will automatically converge to a safe, steady state from any arbitrary, corrupted, or unexpected initial configuration after transient faults cease. In contrast, self-repairing acts as the active execution or 'repair' phase of the broader self-healing process, directed at specific, anticipated faults. Instead of naturally settling into a steady state regardless of the initial configuration, a self-repairing system executes precise physical or software corrections such as isolating a broken component which may purposefully result in reduced functionality or graceful degradation if the system cannot fully recover.

Furthermore, the categorization of the third pair, self-improving and self-evolving, requires a transparent distinction. Self-improvement is frequently discussed as a direct synonym for self-optimization and self-tuning in the autonomic computing literature, referring simply to the continuous adjustment of system parameters to increase operational efficiency. However, based on the related references that define this term within our review, we adopt a more advanced, cognitive-driven interpretation. Consequently, we categorize self-improving as a synonym for self-evolving rather than self-optimization. In this context, both terms describe the exact same advanced capability of intelligent systems to continuously learn and enhance their own cognitive functionality over time. They encompass a system's ability to use machine learning, specifically meta-learning or 'learning of learning', and introspective reflection to progressively transform its internal organizational structure, refine its decision-making logic, and adapt its learning schemes to satisfy emerging functional goals and maintain high non-functional quality standards in complex or previously unknown environments without human intervention.

All other terms in the harmonized list show distinguishable

TABLE I

CONDENSED LIST OF SELF-X CAPABILITIES EXTRACTED FROM UNDERLYING LITERATURE TOGETHER WITH THEIR COUNTS OF ARTICLES, MEANINGS OF THE TERMS, AND POSSIBLY SYNONYM RELATIONSHIPS TO OTHER SELF-X TERMS

ID	Self-X Term	Article Count	Meaning
1	self-healing	211	Autonomous detection, diagnosis, and recovery from hardware or software disruptions to maintain system functionality and safety
2	self-organization	210	Autonomous emergence of global structure, behavior, and collaboration patterns driven by local component interactions without external control
3	self-adaption	201	Autonomous evaluation and adjustment of system behavior, operating models, or control data to maintain goals despite internal and external changes
4	self-optimization, self-tuning	198	Autonomous, continuous tuning of internal parameters, resources, and workloads to maximize operational efficiency and system performance
5	self-management	171	Autonomous administration of a system's own knowledge, resources, and self-* properties to fully take care of its internal state without external human intervention
6	self-protection	132	Autonomous anticipation, detection, and mitigation of malicious attacks, security breaches, and cascading failures to maintain system safety and integrity
7	self-configuration	130	Autonomous setup, integration, and dynamic adjustment of system components, topologies, or parameters in accordance with high-level policies
8	self-awareness	85	Autonomous acquisition and maintenance of knowledge regarding a system's internal state, capabilities, and operational environment to enable intelligent decision-making
9	self-monitoring	52	Continuous, autonomous observation of a system's internal states and external environmental conditions via sensors to detect changes
10	self-repairing	33	Autonomous recovery from hardware or software failures to maintain or re-establish system functionality
11	self-stabilizing, self-recovery	31	Autonomous convergence to a legal, stable state from any arbitrary initial configuration following failures or perturbations
12	self-learning	30	Autonomous improvement of system performance and decision-making by independently exploring, generating, or valuing training data through machine learning and meta-learning techniques
13	self-diagnosis	24	Autonomous monitoring, localization, and root-cause identification of internal system errors or failures
14	self-government	21	Autonomous assumption of responsibility to make decisions and perform tasks in uncertain environments independently of external interventions
15	self-maintenance	18	Autonomous prediction of problems, execution of corrective actions, and scheduling of upkeep tasks to reduce human intervention
16	self-assembly	18	Autonomous formation of complex physical structures or collective morphologies by individual components using local interactions
17	self-description	17	Autonomous provisioning and exposure of a component's capabilities, requirements, and operational roles to enable dynamic discovery and interoperability
18	self-evolving, self-improving	16	Autonomous runtime transformation of organizational structures, interaction patterns, and management logic to accommodate new goals or unknown environments
19	self-regulating	14	Modulated autonomous governance and control of internal states and interactions via feedback
20	self-directedness	12	Autonomous generation of goals and independent pursuit of internal agendas without external control
21	self-calibration	10	Autonomous tuning of system parameters and hardware configurations to maintain model accuracy and data fidelity
22	self-destruction	10	Autonomous termination of a component or system to prevent undesirable emergent behavior, safety violations, or security compromises
23	self-similarity	10	Recursive application of similar structural or functional patterns across different scales, granularities, or tasks
24	self-replication	8	Autonomous duplication of a system or its constituent components to facilitate growth, redundancy, or assembly
25	self-explanation	8	Autonomous provision of understandable descriptions and rationales for system states, capabilities, and decisions
26	self-expression	6	Autonomous runtime modification of an ensemble's organizational structure or coordination patterns to maintain objectives
27	self-knowledge	6	Autonomous acquisition and possession of information regarding a system's own internal state, structure, and capabilities
28	self-anticipating	5	Autonomous prediction of future system states, environmental changes, or potential failures to preemptively adapt behavior or carry out management tasks
29	self-critical	5	Capability to examine whether its own goals and policies are being met and to question the adequacy of its behavior
30	self-installing	5	Autonomous installation of software patches, new components, or operating systems
31	self-simulation	5	Ability to generate and run "what-if" scenarios on an internal model without affecting the live system
32	self-modeling	5	Autonomous generation and continuous refinement of explicit models representing a system's own internal behavior, state, and operational environment
33	self-assessment	5	Autonomous, proactive evaluation of a system's own internal state, resilience, and operational risks to anticipate and prevent threatening issues
34	self-testing	5	Autonomous execution and evaluation of a system's own tests to assess and maintain its correct operation
35	self-planning	5	Autonomous generation and coordination of execution paths, operational settings, and task schedules based on goals and context

or only partially overlapping meanings when their definitions and usage contexts are considered, so they cannot be treated as interchangeable synonyms and are instead placed at different levels of our characterization model as system class goals, major capabilities, or more fine-grained supporting skills (see Section V-B). An example of this is the distinction between the term self-configuration and the terms self-calibration and self-installation, which are closely related and therefore might be conflated. Nevertheless, our review makes a strict distinction between them based on their scope and hierarchical level within the autonomic control loop. We classify self-configuration as a comprehensive fundamental major capability that encompasses the broad, macro-level process of automatically setting up, integrating, and adjusting system components to achieve overall system behavior according to high-level goals without human intervention. In contrast, both self-calibration and self-installation are classified as minor capabilities acting as granular, operational execution mechanisms that directly support the broader self-configuration process. In this execution phase, the system actively focuses on distinct, low-level tasks. Specifically, self-installation executes the physical or logical addition of new components and software, or the re-installation of programs after crashes or degradations. Meanwhile, self-calibration performs precise tuning, such as removing systematic errors in sensor readings or adjusting specific parameters to map raw outputs to standardized units. With the same line of argumentation, other closely related terms such as self-healing and self-repairing, self-awareness and self-knowledge, self-management and self-maintenance, or self-directedness and self-planning are not considered as synonyms.

Finally, only the term self-driving was excluded from the initial list of self-x characteristics. In the analyzed articles, self-driving appears only as an illustrative example, referring to the popularity of autonomous cars. It does not denote a reusable, domain-independent self-x capability in the same sense as the other properties. Consequently, self-driving is omitted from the harmonized list.

Extracting and harmonizing the self-x list as described above, this paper presents 35 self-x properties as common terms from the literature to describe system autonomy and related self-x system classes. In total, 304 distinct self-x terms were identified, which were then filtered by occurrence in at least five papers, and consolidated, including resolving the synonym pairs and excluding the domain-specific label self-driving. Table I provides an overview of the resulting list of self-x capabilities, their mapped synonyms, counts of articles, and derived meanings.

B. *Self-x characterization model*

To answer the second research question, we introduce a conceptual alignment model that structures common self-x properties of autonomous systems by means of system classes, building on the system-class relationships described in Miadowicz et al. (see Section III) and the harmonized list of self-x properties presented in Section V-A.

Fig. 2 gives an overview of the resulting model. Its baseline structure follows the relationships between system classes described by Miadowicz et al. [33]. In summary, their results infer that system automation is a basic capability that aims to execute tasks without continuous human intervention, typically in stable, well-defined environments using rule-based control. Intelligent system approaches, in contrast, determine how to perform a task, make a decision, or predict future states based on experience, enabling systems to learn how to solve particular challenges without being explicitly pre-programmed in complex or uncertain scenarios. Organic systems are a special form of autonomic systems that employ bio-inspired mechanisms to organize distributed and decentralized system scenarios with many interacting components. Autonomic systems, in turn, aim to self-manage complex systems with the goal of achieving low-maintenance, highly reliable operation by embedding management complexity into the system itself. Both organic and autonomic systems can generally be classified as adaptive systems as they realize adaptation through self-management and self-organization mechanisms. Adaptive systems form a more general class which modifies system structure or behavior to adapt to an uncertain environment or find solutions to new or modified goals. Adaptivity can, for example, be realized based on autonomic, organic, or other solution approaches, to provide dependable systems with improved functional or non-functional requirements. Finally, autonomous systems represent a superclass of all other classes, as they might need system capabilities from all classes to be fully independent in all kinds of system scenarios. They need to manage themselves, adapt to an uncertain environment or new requirements, and organize themselves in case of distributed system environments with many components while building on automated and intelligent techniques as foundational building blocks. Therefore, they can be seen as a more general superclass that can be accomplished by one or more approaches of the other types of system.

Overall, the basic structure of the model follows the inclusion hierarchy automated $\subseteq$ intelligent $\subseteq$ organic $\subseteq$ autonomic $\subseteq$ adaptive $\subseteq$ autonomous. In addition, the model introduces a foundational class that groups perception and analysis-related self-x capabilities as universal prerequisites used by all other system classes. Using this structure, each self-x capability is assigned to at least one capability domain by placing it in the circle of the corresponding system class. According to the principle separation of concerns, the characterization of each system class is 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 model, the color of each self-x capability indicates its categorization as a system goal (dark gray), major characteristic (medium gray), or minor characteristic (light gray). In the end, the individual self-x characterization of each system class is used to derive an overall self-x characterization of autonomous systems, which can combine characteristics of the underlying self-x system classes depending on the system goals as well as the technical and organizational environment. Since the system classes are nested as subsets and supersets, self-x capabilities associated with a subset can automatically apply to its superclasses.

As a next step, each system class can be characterized by

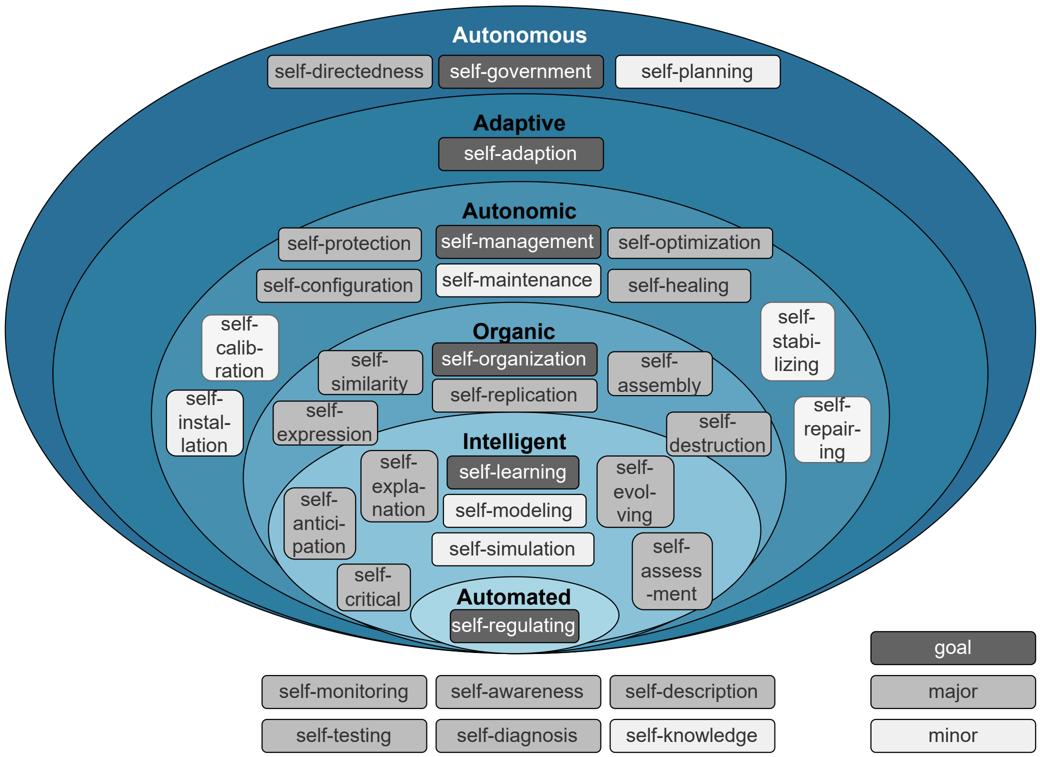

Fig. 2. Self-x characterization model based on system class relationships studied in [33] and the harmonized list of self-x properties in Table I.

the use of self-x capabilities. Starting with the system class goals, these can be determined as self-regulation, self-learning, self-organization, self-management, self-adoption, and self-governing. In our model, self-regulating acts as the baseline objective for automated systems, representing basic closed-loop control without higher-order intelligence. For intelligent systems, self-learning is the defining objective, allowing systems to solve a task based on experience without being pre-programmed. Organic systems target self-organization as their overarching goal, focusing on achieving global, robust system order driven entirely by local, bottom-up interactions among constituent entities. Autonomic systems rely on self-management as their primary goal, aiming to free human administrators from low-level operational complexities, while self-adaption is the logical macro-goal for adaptive systems responding to environmental uncertainty. Ultimately, autonomous systems strive for self-government as their highest-level objective, representing full operational independence and authority.

To continue with the characterization, we derive major characteristics from the system class goals, which serve as the primary functional pillars for achieving these high-level objectives. The foundational class encapsulates the reactive and perceptual baseline required for automated and intelligent adaptation, including self-monitoring, self-awareness, self-description, self-testing, and self-diagnosis. Within this foundational layer, a clear distinction must be made regarding how a system evaluates its current state and environment. As the primary interface for perception, self-monitoring serves as the initial data collection phase, continuously observing and gathering raw information regarding the system's internal resources and changing external conditions. Building upon this collected data, self-awareness broadly perceives and synthesizes these internal and external states into a cohesive understanding. To make this internal understanding available for external interactions and interoperability, self-description formulates an explicit, standardized representation of the system's current state, capabilities, and constraints. To actively verify this understanding, self-testing operates as a functional verification mechanism that executes actual test cases during runtime or idle phases to detect anomalies and ensure quality assurance. If an error or anomaly is indeed detected, self-diagnosis is then employed to determine the specific root cause of that failure.

Building on this foundation, intelligent systems employ advanced proactive and cognitive layers represented by self-evolving, self-explanation, self-anticipation, self-assessment, and self-critical. While the foundational layer is concerned with current states and immediate faults, this intelligent layer introduces proactive evaluation and metacognition. To clearly distinguish these closely related analytical capabilities, it is important to note that while foundational self-diagnosis retrospectively addresses physical or software faults that have already occurred, self-assessment specifically serves as a proactive risk evaluator that analyzes current data to anticipate future vulnerabilities and avoid impending threats. Differentiating even further, self-critical behavior operates at a higher metacognitive level by introspectively evaluating the adequacy and goal-satisfaction of the system's own management policies and adaptation rules rather than just checking if the system is broken or vulnerable. Organic computing utilizes biologically inspired structural capabilities such as self-assembly, self-similarity, self-expression, self-replication, and self-destruction, all concerned with the organization of system ensembles. Autonomic systems encompass the universally accepted self-CHOP properties of self-configuration, self-healing, self-optimization or self-tuning, and self-protection, with broad acceptance in the literature. For autonomous systems, the ability to pursue goals independently is captured by the major capability of self-directedness, which describes the system's ability to generate goals and independently pursue internal agendas without external control.

While major capabilities define what the system does, minor capabilities act as the granular operational execution mechanisms that directly support the major capabilities above them. In the foundational layer, self-knowledge functions as a specific subset of self-awareness, often termed private self-awareness, and deals purely with the internal data repository of the system's own structural information and runtime behavior. To physically implement foundational self-knowledge and self-awareness within the context of intelligent systems, these systems employ self-modeling as a minor capability, serving as the structural and computational mechanism to build, maintain, and refine explicit internal representations for intelligent reasoning. These advanced analytical and cognitive processes are further supported by another minor intelligent capability, self-simulation, which provides the capacity to mentally simulate "what-if" scenarios, experiment with potential adaptations, and predict future outcomes in a virtual sandbox before physically executing them. To physically execute their functional pillars, autonomic computing systems utilize a specialized set of minor capabilities. As detailed in Section V-A, self-calibration and self-installation serve as the granular execution mechanisms for self-configuration, self-repairing and self-stabilizing or self-recovery physically enact the broader self-healing process, and self-maintenance operates as a granular, low-level execution mechanism dedicated to preserving routine operations and preventing long-term system degradation for the overarching macro-level objective of autonomic computing. In the context of the adaptive system class, there is no specific minor self-x capability directly assigned to it; however, this fully aligns with the model's structure, as autonomic and organic computing inherently serve as the underlying functional approaches to realize general adaptivity [33]. Finally, autonomous systems rely on self-planning as a minor capability, which provides the essential computational mechanism required to generate execution paths and action sequences, enabling the system to achieve its overarching major capability of self-directedness.

In summary, we propose a conceptual model that structures harmonized self-x capabilities across an inclusion hierarchy from automated to autonomous systems. By categorizing terms into system goals, major capabilities, and minor execution enablers built upon a foundational class, the model synthesizes fragmented terminology into a cohesive taxonomy. This provides a clear baseline for engineering and evaluating the autonomy and adaptivity of self-x systems.

VI. Conclusion

This article aims to clarify how self-x capabilities are used to characterize autonomous and related system classes and to provide a consistent basis for discussing and comparing such characterizations.

Regarding the first research question, our systematic review of 324 articles identified 304 distinct self-x expressions, which we harmonized into a condensed list of 35 widely used self-x capabilities, including three synonym pairs, each with a clearly specified meaning. This catalog reveals that a small core of properties, most prominently the self-managing self-CHOP properties as well as self-organization and self-adaptation, dominates the reviewed literature across multiple system classes. Other self-x capabilities are less popular, but they are frequently used to describe specific capabilities that are needed to achieve system class goals and major abilities.

Addressing the second research question, we proposed a conceptual self-x characterization model that assigns the harmonized self-x capabilities to automated, intelligent, organic, autonomic, adaptive, and autonomous systems along an inclusion hierarchy from automation up to autonomy, complemented by a foundational class of perception and analysis mechanisms. For each system class, we derived a structured characterization consisting of system-class goals, major capabilities, and minor execution enablers. Combining these class-specific characterizations via the subset relations between system classes yields an overall self-x profile of autonomous systems that can flexibly integrate capabilities from underlying self-x system classes depending on the system's goals and technical or organizational environment.

Taken together, the harmonized catalog of self-x capabilities and the corresponding characterization model provide a common terminology and conceptual foundation for more comparable descriptions of autonomy, for cross-domain reuse of results, and for the future discussion and standardization efforts concerning autonomy and self-x systems.

References

[1] D. B. Abeywickrama, N. Serbedzija, and M. Loreti, "ADSEng: A model-based methodology for autonomous digital services," in *Proceedings of the 11th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS 2016)*, New York, NY: ACM, 2016, pp. 58–69, doi: 10.1145/3012071.3012072.

[2] A. Aniculaesei, J. Grieser, A. Rausch, and K. Rehfeldt, "Towards a holistic software systems engineering approach for dependable autonomous systems," in *2018 IEEE Int. Conf. Autonomic Computing (ICAC)*, 2018, pp. 131–136, doi: 10.1145/3194085.3194091.

[3] P. J. Antsaklis, "Autonomy and metrics of autonomy," *Annu. Rev. Control*, vol. 49, pp. 15–26, 2020, doi: 10.1016/j.arcontrol.2020.05.001.

[4] N. Ashraf and A. Tahir, “Toward autonomic Internet of Things: Recent advances, evaluation criteria, and future research directions,” *IEEE Internet Things J.*, vol. 10, no. 18, pp. 15984–16001, 2023, doi: 10.1109/JIOT.2023.3285359.
[5] C. Johnen, L. Hélouët, and S. Tixeuil, “Self-stabilizing computation and preservation of knowledge of neighbor clusters,” in *2011 IEEE Fifth International Conference on Self-Adaptive and Self-Organizing Systems (SASO 2011)*, Piscataway, NJ: IEEE, 2011, pp. 41–50, doi: 10.1109/SASO.2011.15.
[6] K. Berns and S. Ghosh, “Dissecting self-* properties,” in *2009 3rd IEEE International Conference on Self-Adaptive and Self-Organizing Systems (SASO)*, 2009, pp. 13–22, doi: 10.1109/SASO.2009.25.
[7] F. Borghesan, M. Zagorowska, and M. Mercangöz, “Unmanned and autonomous systems: Future of automation in process and energy industries,” *IFAC-PapersOnLine*, vol. 55, no. 7, pp. 875–882, 2022, doi: 10.1016/j.ifacol.2022.07.555.
[8] S. Brooks and R. Roy, “An overview of self-engineering systems,” *J. Eng. Des.*, vol. 32, no. 8, pp. 397–447, 2021, doi: 10.1080/09544828.2021.1914323.
[9] B. Brukman and R. Yagel, “Self-stabilization as a foundation for autonomic computing,” in *Proceedings of the Second International Conference on Availability, Reliability and Security (ARES 2007)*, Piscataway, NJ: IEEE, 2007, pp. 991–998, doi: 10.1109/ARES.2007.141.
[10] R. Bruni, A. Corradini, F. Gadducci, A. Lluch Lafuente, and A. Vandin, “A white box perspective on behavioural adaptation,” in *Software, Services, and Systems: Essays Dedicated to Martin Wirsing*, M. Hölzl, R. Grotjahn, and M. Wirsing, Eds. Cham, Switzerland: Springer, 2015, pp. 552–581, doi: 10.1007/978-3-319-15545-6_32.
[11] D. K. Chauhan and A. Sharma, “Autonomic computing: A long term vision in computing,” *J. Global Res. Comput. Sci.*, vol. 3, pp. 65–67, 2012.
[12] J. Chen, M. Abbod, and J.-S. Shieh, “Integrations between autonomous systems and modern computing technology: A survey,” *Sensors*, vol. 19, no. 18, p. 3897, 2019, doi: 10.3390/s19183897.
[13] H. Chen, Y. Wen, M. Zhu, Y. Huang, C. Xiao, T. Wei, and A. Hahn, “From automation system to autonomous system: An architecture perspective,” *J. Mar. Sci. Eng.*, vol. 9, no. 6, p. 645, 2021, doi: 10.3390/jmse9060645.
[14] H. M. Cooper, “Organizing knowledge syntheses: A taxonomy of literature reviews,” *Knowledge in Society*, vol. 1, no. 1, pp. 104–126, 1988, doi: 10.1007/BF03177550.
[15] W. Damm and R. Kalmar, “Autonome Systeme – Fähigkeiten und Anforderungen,” *Informatik-Spektrum*, vol. 40, no. 5, pp. 400–408, 2017, doi: 10.1007/s00287-017-1063-0.
[16] F. Duan, X. Li, Y. Liu, and Y. Fang, “Towards autonomic computing: A new self-management method,” in *Artificial Intelligence and Computational Intelligence*, H. Deng *et al.*, Eds. Berlin, Heidelberg: Springer, 2011, pp. 292–299.
[17] L. Duan and L. D. Xu, “Data analytics in Industry 4.0: A survey,” *Inf. Syst. Front.*, vol. 26, no. 6, pp. 2809–2825, 2024, doi: 10.1007/s10796-021-10190-0.
[18] L. A. Estrada-Jimenez, T. Pulikottil, S. Nikghadam-Hojjati, and J. Barata, “Self-organization in smart manufacturing—background, systematic review, challenges and outlook,” *IEEE Access*, vol. 11, pp. 10107–10136, 2023, doi: 10.1109/ACCESS.2023.3240433.
[19] F. Flammini, A. De Benedictis, and A. D’Onofrio, “Digital twins for anomaly detection in the industrial Internet of Things: A survey,” *IEEE Trans. Emerg. Top. Comput.*, vol. 12, no. 1, pp. 197–211, 2024, doi: 10.1109/TETC.2022.3227113.
[20] R. Frei and J. Barata, “Distributed systems – from natural to engineered: Three phases of inspiration by nature,” *Int. J. Bio-Inspired Comput.*, vol. 2, nos. 3–4, pp. 258–270, 2010, doi: 10.1504/IJBIC.2010.033094.
[21] R. Frei and G. Di Marzo Serugendo, “Advances in complexity engineering,” *Int. J. Bio-Inspired Comput.*, vol. 3, no. 4, pp. 199–212, 2011, doi: 10.1504/IJBIC.2011.041144.
[22] T. Gamer and M. Hörnicke, “The autonomous industrial plant – future of process engineering, operations and maintenance,” *IFAC-PapersOnLine*, vol. 52, no. 7, pp. 454–460, 2019, doi: 10.1016/j.ifacol.2019.06.104.
[23] O. Gheibi, D. Weyns, and F. Quin, “Applying machine learning in self-adaptive systems: A systematic literature review,” *ACM Trans. Auton. Adapt. Syst.*, vol. 15, no. 4, Art. 22, 2021, doi: 10.1145/3469440.
[24] M. Güdemann, C. von der Malsburg, U. Brinkschulte, and C. Hochberger, “A specification and construction paradigm for organic computing systems,” in *2008 Second IEEE International Conference on Self-Adaptive and Self-Organizing Systems (SASO 2008)*, Piscataway, NJ: IEEE, 2008, pp. 36–45, doi: 10.1109/SASO.2008.66.
[25] M. Hölzl, N. Koch, M. Puviani, M. Wirsing, and F. Zambonelli, “The ensemble development life cycle and best practices for collective autonomic systems,” in *Software Engineering for Collective Autonomic Systems*, D. Weyns, S. Malek, and R. Mirandola, Eds. Cham, Switzerland: Springer, 2015, pp. 329–356, doi: 10.1007/978-3-319-16310-9_9.
[26] P. Horn, *Autonomic Computing: IBM’s Perspective on the State of Information Technology*. Armonk, NY: International Business Machines Corporation, 2001.
[27] C.-E. Hrabia, N. Masuch, and S. Albayrak, “A metrics framework for quantifying autonomy in complex systems: Chapter 2,” in *Multiagent System Technologies*, LNCS 8732, Cham: Springer, 2015, pp. 22–41, doi: 10.1007/978-3-319-27343-3_2.
[28] J. vom Brocke, A. Simons, B. Niehaves, K. Riemer, R. Plattfaut, and A. Cleven, “Reconstructing the giant: On the importance of rigour in documenting the literature search process,” in *European Conference on Information Systems (ECIS)*, 2009.
[29] E. Kaddoum, C. Raibulet, J.-P. Georgé, G. Picard, and M.-P. Gleizes, “Criteria for the evaluation of self-* systems,” in *Proceedings of the 32nd International Conference on Software Engineering (ICSE 2010), Volume 2*, New York, NY: ACM, 2010, pp. 13–22, doi: 10.1145/1808984.1808988.
[30] H. Hirsch-Kreinsen and D. Gerst, Eds., *Autonome Systeme und Arbeit: Perspektiven, Herausforderungen und Grenzen der künstlichen Intelligenz in der Arbeitswelt*. Bielefeld, Germany: transcript Verlag, 2019, doi: 10.1515/9783839443958.
[31] F. D. Macías-Escrivá, R. Haber, R. Del Toro, and V. Hernandez, “Self-adaptive systems: A survey of current approaches, research challenges and applications,” *Expert Syst. Appl.*, vol. 40, no. 18, pp. 7267–7279, 2013, doi: 10.1016/j.eswa.2013.07.033.
[32] A. Mateen, B. Raza, M. Sher, M. M. Awais, and N. Mustapha, “Workload management: A technology perspective with respect to self-* characteristics,” *Artif. Intell. Rev.*, vol. 41, no. 4, pp. 463–489, 2014, doi: 10.1007/s10462-012-9320-8.[file:26]
[33] I. Miadowicz, D. Maldonado Quinto, and M. Felderer, “A systematic literature review on the intersection of self-X system classes,” *ACM Comput. Surv.*, vol. 58, no. 7, Art. 171, 39 pp., Jan. 2026, doi: 10.1145/3778859.
[34] F. Mo *et al.*, “A maturity model for the autonomy of manufacturing systems,” *Int. J. Adv. Manuf. Technol.*, vol. 126, nos. 1–2, pp. 405–428, 2023, doi: 10.1007/s00170-023-10910-7.
[35] M. Müller, T. Müller, B. Ashtari Talkhestani, P. Marks, N. Jazdi, and M. Weyrich, “Industrial autonomous systems: A survey on definitions, characteristics and abilities,” *at - Automatisierungstechnik*, vol. 69, no. 1, pp. 3–13, 2021, doi: 10.1515/auto-2020-0131.
[36] M. Nickschas and U. Brinkschulte, “Guiding organic management in a service-oriented real-time middleware architecture,” in *Organic Computing – From Neural to Pervasive Systems*, C. Müller-Schloer, H. Schmeck, and T. Ungerer, Eds. Berlin, Germany: Springer, 2008, pp. 135–149, doi: 10.1007/978-3-540-87785-1_9.
[37] P. Nuno, J. C. Granda, F. J. Suarez, and D. F. Garcia, “Self-* in multimedia communication overlays,” *Comput. Commun.*, vol. 36, no. 7, pp. 817–833, 2013, doi: 10.1016/j.comcom.2012.12.009.
[38] M. Onori, D. Semere, and B. Lindberg, “Evolvable systems: An approach to self-X production,” *Int. J. Comput. Integr. Manuf.*, vol. 26, no. 8, pp. 711–727, 2013, doi: 10.1080/0951192X.2011.566282.
[39] A. Pinto, “An open and modular architecture for autonomous and intelligent systems,” in *2019 IEEE International Conference on Embedded Software and Systems (ICESS)*, 2019, pp. 1–8, doi: 10.1109/ICESS.2019.8782512.
[40] M. Salehie and L. Tahvildari, “Autonomic computing,” *ACM SIGSOFT Softw. Eng. Notes*, vol. 30, no. 4, pp. 1–7, 2005, doi: 10.1145/1082983.1083082.
[41] M. Salehie and L. Tahvildari, “Self-adaptive software: Landscape and research challenges,” *ACM Trans. Auton. Adapt. Syst.*, vol. 4, no. 2, Art. 14, pp. 1–42, 2009, doi: 10.1145/1516533.1516538.
[42] M. Salehie and L. Tahvildari, “Towards a goal–driven approach to action selection in self–adaptive software,” *Softw. Pract. Exper.*, vol. 42, no. 2, pp. 211–233, 2012, doi: 10.1002/spe.1066.
[43] H. Schmeck, “Organic computing – a new vision for distributed embedded systems,” in *Proceedings of the Eighth IEEE International Symposium on Object-Oriented Real-Time Distributed Comput-*

ing (ISORC 2005), Piscataway, NJ: IEEE, 2005, pp. 201–203, doi: 10.1109/ISORC.2005.42.

[44] T. Singh and A. Kumar, "Survey on characteristics of autonomous system," *Int. J. Comput. Sci. Inf. Technol.*, vol. 8, no. 2, pp. 121–128, 2016, doi: 10.5121/ijcsit.2016.8210.

[45] R. Sterritt, "Autonomic computing," *Innov. Syst. Softw. Eng.*, vol. 1, no. 1, pp. 79–88, 2005, doi: 10.1007/s11334-005-0001-5.

[46] D. Stock, C. Kuhn, N. Gruschka, A. Karmakar, and N. Pohlmann, "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)*, 2020, pp. 1789–1796, doi: 10.1109/ETFA46521.2020.9212182.

[47] C. Szabo, B. Sims, T. Mcatee, R. Lodge, and R. Hunjet, "Self-adaptive software systems in contested and resource-constrained environments: Overview and challenges," *IEEE Access*, vol. 9, pp. 10711–10728, 2021, doi: 10.1109/ACCESS.2020.3043440.

[48] A. Valente, M. Holanda, A. M. Mariano, R. Furuta, and D. Da Silva, "Analysis of academic databases for literature review in the computer science education field," in *2022 IEEE Frontiers in Education Conference (FIE)*, Piscataway, NJ: IEEE, 2022, pp. 1–7, doi: 10.1109/FIE56618.2022.9962393.

[49] K. G. Vamvoudakis, P. J. Antsaklis, W. E. Dixon, J. P. Hespanha, F. L. Lewis, H. Modares, and B. Kiumarsi, "Autonomy and machine intelligence in complex systems: A tutorial," in *Proc. 2015 American Control Conference (ACC)*, 2015, pp. 5062–5079, doi: 10.1109/ACC.2015.7172127.

[50] N. M. Villegas, G. Tamura, and H. A. Müller, "Architecting software systems for runtime self-adaptation: Concepts, models, and challenges," in *Managing Trade-Offs in Adaptable Software Architectures*, I. Mistrik *et al.*, Eds. Boston, MA: Morgan Kaufmann, 2017, pp. 17–43, doi: 10.1016/B978-0-12-802855-1.00002-2.

[51] D. Weyns, M. U. Iftikhar, S. Malek, and J. Andersson, "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)*, Piscataway, NJ: IEEE, 2012, pp. 89–98, doi: 10.1109/SEAMS.2012.6224395.

[52] M. A. Yahya, M. A. Yahya, and A. Dahanayake, "Autonomic computing: A framework to identify autonomy requirements," *Procedia Comput. Sci.*, vol. 20, pp. 235–241, 2013, doi: 10.1016/j.procs.2013.09.267.

[53] H. Zhang, M. A. Babar, and P. Tell, "Identifying relevant studies in software engineering," *Inf. Softw. Technol.*, vol. 53, no. 6, pp. 625–637, 2011, doi: 10.1016/j.infsof.2010.12.010.