# Self-improving System Integration: Mastering Continuous Change

Kirstie Bellman[a], Jean Botev[b], Ada Diaconescu[c], Lukas Esterle[d], Christian Gruhl[e], Christopher Landauer[a], Peter R. Lewis[f], Phyllis R. Nelson[a,j], Evangelos Pournaras[g], Anthony Stein[h], Sven Tomforde[i,*]

[a]*Topcy House Consulting, California, USA*
[b]*University of Luxembourg, Luxembourg*
[c]*Telecom Paris, LTCI, IPParis, France*
[d]*Aarhus University, Denmark*
[e]*University of Kassel, Germany*
[f]*Aston University, Birmingham, UK*
[g]*University of Leeds, UK*
[h]*University of Hohenheim, Germany*
[i]*Christian-Albrechts-Universität zu Kiel, Germany*
[j]*California State Polytechnic University Pomona, USA*

**Abstract**

The research initiative "self-improving system integration" (SISSY) was established with the goal to master the ever-changing demands of system organisation in the presence of autonomous subsystems, evolving architectures, and highly-dynamic open environments. It aims to move integration-related decisions from design-time to run-time, implying a further shift of expertise and responsibility from human engineers to autonomous systems. This introduces a qualitative shift from existing self-adaptive and self-organising systems, moving from self-adaptation based on predefined variation types, towards more open contexts involving novel autonomous subsystems, collaborative behaviours, and emerging goals.

In this article, we revisit existing SISSY research efforts and establish a corresponding terminology focusing on how SISSY relates to the broad field of integration sciences. We then investigate SISSY-related research efforts and derive a taxonomy of SISSY technology. This is concluded by establishing a research road-map for developing operational self-improving self-integrating systems.

*Keywords:* Self-Integration, Self-Improvement, Autonomous Systems, Taxonomy, Organic Computing, System Engineering

## 1. Introduction

Over the last few decades, there has been a tremendous increase in the complexity of large computing systems, primarily due to the increasing interconnectivity and interaction between systems, components, and services. These entities are, in turn, ever more autonomous, heterogeneous, and larger in scale, while also operating in increasingly open and dynamic environments. Hence, most of today's systems evolve frequently and continuously over time [1, 2], often interacting with changing and even completely novel systems and execution contexts. This leads to the need for an ongoing requirement for system-of-systems integration. Systems can be interconnected either purposefully, for achieving novel intended goals, opportunistically, by *ad-hoc* interaction, or accidentally, through interference within a shared environment.

Systems engineering experience testifies that even carefully designed and extensively tested solutions are unable to react properly to all sources of problems that might be entailed by such a continuously changing integration status. For example, unforeseen interactions may occur when conventional interfaces, defined at design-time, are bypassed with a connection through the physical environment, including humans. This situation is further complicated by the lack of shared knowledge and/or pooled authority when considering the control of all involved systems [3]. Thus the self-improving system integration community [4, 5] has emphasised the consideration of what are known as *Interwoven Systems* [6, 7, 8] – that is, systems that integrate themselves and adapt as necessary, in response to their own behaviour and each other, within open and unpredictable environments.

These challenges drive the necessity to push system design, development and testing operations into run-time, with the concomitant need for real-time adaptations in knowledge acquisition, representation and processing. Doing so entails more system autonomy, intelligence, and self-awareness, to facilitate suitable reactions at run-time, when human experts may be unable to intervene. Related initiatives such as Organic Computing (OC) [2], Autonomic Computing [9] and Self-Aware Computing [10] have also emphasised the need to tackle ever increasing system complexity, by endowing systems with self-adaptive and self-organising (SASO) functionalities, or "life-like" properties. The underlying concept in each of these initiatives is to design systems with inherent self-adaptive, self-organising, and self-improving capabilities that lead to more flexible, robust and resilient behaviours during run-time. We

*Corresponding author:
*Email address:* st@informatik.uni-kiel.de (Sven Tomforde)

will refer to this class of systems as SASO systems throughout this article.

These initiatives have brought about exciting insights and developments, with considerable success in developing 'stand-alone' SASO systems. At the same time, there continue to be substantial challenges to integrate multiple heterogeneous systems each with different self-adaptive and self-organising capabilities. Further challenges include combining technical and human organisations, which are also potentially unfamiliar to each other, the need to operate at very large scales, and operating in highly unpredictable environments. Several complementary initiatives focusing on systems-of-systems methodologies (e.g., service-oriented architectures [11] and cloud technology [12]) have also seen significant progress relevant to the aforementioned concerns. Still, most of these developments lack scalable methods for rapidly adapting components and sub-systems to their new integrated environments (e.g., optimal or viable configurations or compositions).

As a result of these observations, the concept of "Self-Improving System Integration" (SISSY) was established. The goal of the SISSY initiative is to address the challenges of system-of-system self-organisation and self-adaptation. Specifically, SISSY investigates approaches that allow for continual self-integration among components and systems that are self-improving (i.e. learning or evolving over time) towards effective and stable outcomes. This is considered the solution to the 'Interwoven Systems' problem outlined above.

Of course, variants of the above SISSY challenges have been approached by many different fields, ranging from mathematics and computer science to cognitive sciences. With respect to middleware research, SISSY goes beyond establishing reliable (self-adaptive) communication services, by adapting physical and functional connectivity, and by providing associated services like security, transactions and persistence. SISSY aims to ensure that self-integrated systems-of-systems feature (at least) functional coherence and, ideally, are able to meet their goals, while self-improving and self-adapting in the face of unforeseen challenges.

Concerning to previous efforts in SASO systems research, SISSY represents a qualitative shift rather than a radical different approach. It changes the process from designing adaptive processes within a well-understood problem space to developing what is now needed to negotiate and explore new configurations and collaborations over multiple systems at run-time. The biggest consequence of this shift is that designers need more than straightforward parameter changes or component hot-swapping. Such approaches are based on a known adaptation space, and can thus only respond to predefined types of changes (however useful they are in the short term). Rather, they must now entertain the possibility that there will be not only new usages for currently deployed or available components and processes, but, because of new system collaborations, also new and unforeseen interactions, interfaces, and interferences between behaviours and objectives. Meeting this challenge requires systems to carry out new forms of reasoning and self-modelling.

Furthermore, this kind of integration, once accomplished, is never finished in a SISSY system-of-systems because changes will continue to occur. Such changes may be within partnering systems (intentionally or by corruption), in collective behaviours, in executing environments, and in targeted goals. Thus system integration must not only take into account the systems (and components) that are currently interfaced, but also the frequent insertion of new components and systems as well as their utilised functions and featured behaviours within ever-changing execution environments.

Qualitatively, the SISSY initiative goes beyond conceiving adaptive and maintenance processes for mature systems with persistent predefined goals where the learning and adaptive processes are directed towards retaining the desired state, performance, or set of behaviours. It broadens the scope of systems integration to also include a set of novel, and not currently well-defined, processes that create genuinely novel behavioural patterns and joint goals among a set of simultaneously self-developing systems, through negotiation and what we call "active experimentation" processes [13, 14].

Although the SISSY initiative has produced various contributions towards mastering continuous change, it has not yet fully achieved its goal. This article summarises the current state-of-the-art and establishes an agenda for the years to come. In particular, the article's contributions are as follows:

- Defining the SISSY concept (Section 2);
- Briefly positioning SISSY within the overall field of integration sciences (Section 3);
- Reviewing the research efforts within the SISSY initiative and deriving a taxonomy of different approaches (Section 4);
- Identifying a research road-map for filling the gaps with SISSY research (Section 5).

## 2. Basic terminology: Self-improving System Integration

We characterise the field of "self-improving system integration" (SISSY) as an attempt for purposeful and autonomous composition and inter-operation of complex systems that may have both adaptive and non-adaptive components or subsystems. In the following paragraphs, we approach a more precise definition of the term by initially revisiting the notion of "integration" in traditional engineering, followed by a discussion of the "self" and the "improving" parts.

In engineering, the notion of integration describes a process in which several component (sub-)systems are brought together and interconnected into a unified system. That is, integration is about bringing together disparate elements, often thought of as "parts" or components (even when they do not have very distinguishable boundaries), and combining them into an integral "whole" that is expected to have capabilities that none of the parts does.

Some notable examples of integration are:

- organs in biological systems;

- people in organisations;
- hardware components in other engineered artefacts;
- software components in computing systems.

The first example works because it has already been done (though it seems to have taken millions of years). The second one can work because people have enormous flexibility of interaction, and can iron out or work through any differences or difficulties (which is not to say that they always do that). The third one can work because hardware components have a limited repertoire of behaviours that we generally understand (humans have been building and breaking hardware devices for as long as there have been humans). The fourth one is the most difficult, and the subject of this paper (humans have been building and breaking software systems for only a little over a hundred years).

In particular, integration is more than just inter-operation since the latter means that the participating autonomous systems may act completely independently, while the former requires a purposeful interaction. The essential part of integration is the interlocking dependencies and shared internal capabilities that make it different from inter-operability (which also means that the systems can work together, but implies that they treat each other more like black boxes). This aims to achieve a correctly working unit, where the subsystems work together to provide desired functions, with acceptable performance and dependability properties. In classic engineering, the integration of (sub-)systems is done at *design-time*, with rigid specifications and testing of (sub-)system interfaces and performance based on preconceived use cases. However, with increasingly networked and open systems (e.g., Internet of Things [15], smart homes and grids [16], or smart traffic [17, 18]), we now face the challenge of integrating systems dynamically, and as rapidly as possible. Because of the dynamic nature of contexts — where goals, resources, and knowledge required for integration change rapidly — increasing efforts have been directed towards new processes and computations that allow intelligent systems to do most of the integration themselves at *run-time*.

Hence, we define "self-integration" as an ongoing autonomous process for linking a potentially large set of heterogeneous computing systems, devices, and software applications, so as to accomplish system goals. The linking itself is done physically or functionally. We consider this process to be continual, i.e., it is never finished, as the way in which the integrated elements — software and hardware — act together must adapt to external changes, goal evolution, and autonomous decisions of these elements [8]. The goals to be achieved typically not only cover a single objective — more likely they may comprise several (possibly conflicting) aspects. There may even be a deviation between goals of the overall system and the individual component subsystems, see [19] or [20].

Within this process, two major tasks are performed: (i) connections between subsystems are recognised, turned into a purposeful interaction, and tested; and, (ii) the particular subsystems in combination with their behavioural strategies are configured depending on the operational and functional area. This has a strong overlap with the concept of "self-organisation" [21]: (i) connecting subsystems or elements refers to building the system's structure; and, (ii) configuration is mainly concerned with enabling the desired behaviour. It also relies on concepts from self-aware computing systems [10] that use internal knowledge and learning processes to reason about their resources and state. Multi-agent systems (MAS) also provide relevant concepts in terms of organisational paradigms [22], where agents can cooperate or compete, forming hierarchies, coalitions, teams, federations, societies and so on [23].

Besides self-aware computing systems and MAS, the vision of SISSY is closely related to the concept of 'intention awareness' (IA) which is defined as "[...] the process of integrating actors' intentions into a unified view of the surrounding environment [...]" in [24]. Compared to IA, SISSY aims at keeping the human out of the loop and focusing on on the system engineering and design perspective. The particular difference become more visible in the context of the taxonomy developed later in this article (cf. Section 4).

As mentioned before, self-integration is a continuous and continuing process rather than a single completable task. It is also an open process, i.e., it is not bounded or even well-defined in advance. Since not all conditions can be anticipated, the process needs a "creative component". We call this activity a "process", since the "self-" can either refer to each autonomous subsystem or to centralised instances responsible for these tasks; although in many cases hybrid constellations will exist because they provide a compromise reflecting the different responsibilities and the varying autonomy aspects. In particular, these "creative components" (centralised, decentralised or hybrid) are responsible for inventing solutions where none are otherwise available and for optimising the selection of solutions over time. We refer to this concept as "self-improvement".

Self-improvement implies that self-integrating systems do not settle for the status quo, but rather rely on continuous learning (which may include different types of the evolution of integration that is not necessarily the result of applied machine learning technology, such as experimentation and subsequent adjustment). Here, learning is responsible for optimising the behaviours with respect to the goals, even though these "creative components" may or may not be able to reach optimal states or behaviours. Such a self-improvement process requires a utility function that guides the optimisation process towards desired states. However, this utility is not necessarily a single objective. It may also be composed of several aspects, and these aspects can potentially conflict, as highlighted in [19], requiring negotiation and mediation processes to make the appropriate trade-offs for any given situation. While ideally different constituents negotiate among themselves, we might need a third party or external process to ensure a beneficial outcome for the overall system — this is covered by "mediation" efforts.

Based on the previous discussion, we differentiate between integration within an overall system and integration between individual systems. If competing systems are integrated (i.e., they come into contact) the self-improvement process does not necessarily go towards a "desired state", as there is no such state in the ABSOLUTE; here, we talk about a stable state or

equilibrium instead.

In general, the capability of "self-improvement" relies on reflective, intelligent behaviour. This means we need a computer system that "achieves a certain performance even in time-variant environments and in disturbed situations, which is self-adapting and optimises its behaviour through experiences" [25]. Alternatively, it is at least able to maintain an acceptable goal achievement if unexpected events or other disturbances and uncertainties occur. Both cases typically require that the system can autonomously assess its performance based on a given goal or utility function.

There are different types of self-integration:

- Explicit (or wanted, intended) self-integration: Integration of component sub-systems within a single system or integration of systems within a system-of-systems, but done purposefully, with the intent of achieving more-or-less well-defined objectives, or goals. We refer to this concept as collaboration.

- Implicit (or *ad-hoc*, unintended, even hazardous) self-integration: Integration of systems that share a common environment through which they impact each other directly or indirectly. Here there is no shared purpose or goal, at least not initially. This may include scenarios of unintended cooperation but also covers competition.

Other cases are possible. For example in "attack"-oriented integration a system is introduced on purpose to interfere with an existing system to decrease its utility and correspondingly urge it to dissolve its integration state. Besides using the type of integration process (explicit and implicit) for characterisation, we may further distinguish integration in terms of the trigger (i.e., active and passive), the intention (i.e., benevolent, malevolent, indifferent / neutral), and so on. For this article, however, we stay with a basic characterisation fusing the type of process.

Finally, we want to emphasise that self-integration is not necessarily always desirable. For instance, systems may self-improve, for their utility function, by deciding to disintegrate from others. In addition to improving a system's performance, disintegration may also be a way to cope with or to mitigate a cascading failure. There is evidence from other scientific fields, e.g., network dismantling [26] and industrial settings [27], that this approach can lead to beneficial outcomes. However, there are no examples available in the context of SISSY research yet, where self-disintegration has been investigated.

## 3. Integration Science

As discussed in the previous section, we consider self-integration to be much more than just putting things together. In [28], we introduced the notion of an "Integration Science", and explained why we thought it was necessary for the construction of complex software-intensive systems. It is based on the biological principles discussed in [29] [30] – observing a surprising collision of biological behaviour with mathematics [31] and complex computing system modelling.

As we use it, Integration Science is (still) a hopeful term, based on the appalling historical (and usually intended) rigidity of most complex computing systems. The goal is to take what is currently done as largely a set of *ad-hoc* methods, applied by often very experienced engineers, and develop methods supported by principles and where possible more formal and mathematical foundations. Integration Science was created for use by analysts and developers who would apply the methods to complex systems. What happens when we take such Integration Science methods and have the self-adaptive and self-organising (SASO) systems apply them? What works and what does not work?

For example, when a system makes a trade-off as part of its integration with other systems, these trade-offs will always be based on the partial information and local perspective of the self-integrating system. This implies a less than perfect global solution, as all integration will be "self-centred integration", based on multiple local perspectives. An external third party could conceivably find a better solution than that proffered during negotiation by two agents, although other third party agents will also be limited by lack of full information and full control over the other agents. This effect was studied by Van Moffaert et al [32] in the context of decentralised control of smart camera networks. Thus we can expect no run-time integration to compare to the careful months of study used to investigate trade-offs and feedback loops in carefully controlled but necessarily limited integration scenarios. For critical systems, we may, in fact, want human agents in the loop to ensure better solutions. Hence the negotiated integration will never be perfect; they will always be partial and based on what is known at the time, and what current operational environment the systems are in.

Given that, are there Integration Science principles that should be applied by the SASO system? We, in fact, have three critical principles from biological systems that should be used by SASO mechanisms, which are viability, improvement, and currency. **Viability** means that a system always keeps a vital level of functioning and performance. In terms of self-integration that means that any compromises and trade-offs are strictly limited by the imperative that each partner system must survive to try to improve again. Integration and the negotiation about integration with partners is a continuous process.

The second principle is that the system always strives to **improve** its current situation. In other work, we have pointed out that biological systems change their environments, and modify their goals and behaviours, to place themselves in better positions. This includes always deciding their next responses in parallel time windows, from immediate to longer-term and always deciding their next responses in terms of multiple objectives. [33] In the context of self-improving self-integration, this improvement includes the critical step of working with partnering systems on how the system intends to improve and making that also part of the negotiation.

The third principle is the **currency**. Although biological systems reason and plan over several timescales at once, their reasoning is always grounded in what their current state, capabilities and operational environment are. If the situation changes, biological systems readily drop goals, take advantage of new opportunities, and (important for integration) change what they

are doing and how they are doing it. That includes recruiting additional components and often different partnering systems.

However, principles are only marginally helpful unless they help developers select or invent approaches to apply them to design problems, and methods to implement them for constructing complex computing systems. In a companion paper [34] in this issue, we describe here some of our results in these directions.

Fundamentally, computing system integration is about making different computational resources work together to provide some capability that none of them has by itself, or that is better in some measurable way than any of them can provide alone. It requires the system designers to have sufficient knowledge about each computational resources' capabilities, strengths and weaknesses, performance, and modes of use to make proper use of them in the appropriate context. No matter how you choose to do integration, you have to satisfy some version of these requirements, even if you discard the relevant data and reasoning processes (the "integration scaffolding") once the system is ready for use.

In [31], we describe in detail our notion of "Integration Science", including integration theory, mechanism, and implementation style that retains all of that scaffolding so it is accessible to the run-time system, which means that integration can occur continually throughout the lifetime of the system.

To show how diverse integration can be, and how wide the applicability of these methods can be (not just comprising the well-known smart city, cloud and grid management applications), we finish this section with some small but interesting tasks for which a SISSY approach seems appropriate, more details can be found in [35]. The setup for all of them is that there is a team of cooperating systems already in place (we address initial team formation elsewhere), and we are writing from the point of view of a new system that wants to join the team (perhaps they have been directed, perhaps they have been asked, perhaps they are volunteering).

The five applications are

- Joining a Game;
- Joining a Caravan of Vehicles;
- Joining a Search;
- Joining a Research Project;
- Joining a Surveillance System.

We describe each in turn and then show that they have many commonalities for any system that is trying to join.

Each **game** has several *roles* to be played by participants (e.g., goalie, striker, shortstop, tight end, bowler, referee, umpire, official scorer, manager / coach), and a set of rules applicable to each role. There may be different competing teams (this use of the term *team* is more specific than our general use of the term to refer to all participants), and there may be different positions / roles on each competing team. To play a game competently requires a set of strategies (beyond the simple rules of the game) that govern interaction behaviour with other team members. There must also be a set of *failouts*,which are strategies to cope with the failure of the game playing strategies (or failure to follow the applicable rules).

A **caravan** is a much more dangerous team because it is presumed to be moving heavy loads fairly quickly. However, there are still a few different roles, since the behaviour of the leading and trailing vehicles (which set the pace and clear the road) are not likely to be the same as the internal vehicles. There may also be a supervisor (local or remote) that can manage the entry and exit of vehicles. The primary focus has to be on the positions and trajectories of the vehicles, and their velocity and acceleration capabilities (which may all be different). The main activity is to open a place for a new vehicle, make sure it can accelerate to the right place in the caravan, and issue weather and road condition warnings to the entire caravan. Here also, failouts are extremely important (flat tyres, unexpected obstacles, sudden failures of vehicles, hail storms, etc.).

A **search team** has different roles for different capabilities: available sensors, region knowledge, equipment knowledge, carrying capacity, mobility, medical expertise, etc.. This team is complicated by the likelihood of a hard but unknown deadline, after which a successful rescue is unlikely or impossible. That tempers all performance considerations with their time requirements.

A **research project** also has roles, depending more on equipment knowledge, background knowledge, domain expertise, etc., and this one has more explicit but less stringent temporal constraints. We include this one because, even though the participants are humans, they have to go through some of the same steps as our computing systems.

A **surveillance system** (reconnaissance of a limited area) has roles for the different sensors and knowledge of their sensitivity and mobility. There may also be *ignored characteristics*, which are indicators of an uninteresting phenomenon, and *target characteristics*, which are indicators of an interesting phenomenon. There also may be a mandate to distinguish usual from unusual phenomena in the field of interest.

Next, and finally, we turn to a summary of aspects common to all of these applications (with sufficient abstraction).

There are different roles for different participants, usually specified in advance (how many, what kind, etc.). The expected capabilities for each role are also specified in advance, involving physical position, velocity and acceleration, arrangement, flexibility, mobility, and carrying capacity (both size and weight).

The expected knowledge is also specified in advance, such as regional, equipment, background domain, or medical. There are often stringent time constraints that affect how much computation can be done for any of these considerations.

There are behavioural constraints (game rules, laws, regulations, etc.), usually different for different roles, in the form of what a participant must, may, may not, or must not do in certain situations.

There may be cooperative cliques (teams in the smaller sense), with different roles within teams. There may be competitive cliques, or actively hostile cliques that need to be monitored.

There are behavioural goals (strategies, plans, etc.), possibly different for different team roles (and for different cliques), involving positions and trajectories, adjustment of movement, environmental characteristics that matter (or do not matter). There may be a mandate to identify usual and unusual phenomena.

There are deficiency mitigation processes for discovering behavioural problems, avoiding them, recovering when they occur, and for predicting them. These often involve control roles to identify problems (e.g., referees) for failure to follow the applicable rules.

There are also other failouts, which are not so much about participant system behaviour as they are about the inevitable breakdown of physical devices and the resultant invalidation of assumptions about the environment, the equipment, and even the system. Some of these can be predicted, some will just be accidents; all of them must be accounted for in advance or detected and deflected or resolved in some way.

## 4. A Taxonomy of SISSY approaches

The previous section introduced the term "Integration Science" and discussed the intention of the field, the principles, and the general applicability. The SISSY initiative emerged as a joint effort combining the insights and the current state of "Integration Science" as a field with the principles of SASO systems, including the similar biological sources of inspiration (especially found in the Organic Computing initiative).

In this section, we investigate the current state of the art in SISSY and determine open challenges to be tackled towards fully operational SISSY systems. To do so, we analyse the different contributions in the broader field of self-adaptive and self-organising (SASO) systems that are especially dedicated to integration. We intentionally do not cover all possible efforts towards integration science, since we want to put an emphasis on those efforts that focus on the autonomic or "self" aspect. In particular, this implies that we leave efforts in other domains such as hardware-based systems, (most of the field of) software engineering, multi-agent systems, or machine learning out of consideration.

The remainder of this section initially describes our research method, including the research questions to be answered by the method. Afterwards, we categorise the different contributions that we identified with the method and summarise the current state of the art. This is accompanied by an assessment of what needs to be done in the specific (sub-)category towards establishing and developing systems with the desired SISSY capabilities as outlined in Section 2. This is later consolidated into a research roadmap (see Section 5) outlining basic directions of necessary research efforts. Finally, this section investigates the correlations of categories of current research efforts to provide an interpretation where a tighter binding is necessary.

### *4.1. Method*

The adopted research method used in this section follows the standard practice in systematic literature reviews (see, e.g., [36]). Since our study focuses on integration aspects of SASO systems, the scope of the review is restricted to SASO contributions where integration decisions play a major role.

To steer our review, we defined a set of research questions to consider in the analysis of each of the selected contributions:

1. Is integration dedicated to system organisation and/or local behaviour decisions of subsystems?
2. What is the purpose of integration within the SASO system?
3. Which integration techniques are employed?
4. What are the triggers to integration behaviour?

We focus on reviewing advanced and high-quality studies published in the main conferences and journals in the areas of self-adaptive, self-organising, and organic/autonomic computing systems. We used the following search query to select contributions automatically:

| (integration) AND (software OR application OR system) |
|---|

We intentionally did not extend the scope to phrases that can be considered as synonyms of self-integration such as self-assembly, dynamic composition, self-/dynamic coordination, self-growing, self-construction, and so on. The basic motivation is that we want to stay with the wording to keep the SISSY focus. Furthermore, the research aims at analysing efforts directly related to self-integration tasks rather than summarising efforts of the self-adaptive and self-organising, autonomic computing or multi-agent communities that may be beneficial for developing SISSY systems.

Table 1 lists the considered venues which were searched using the respective databases (ACM DL, IEEE Xplore). Full names of the conferences can be found in the appendix. To ensure that the selected studies capture recent scientific advances, we considered studies published in or after the year 2000. The last column lists the number of studies that match the query and (in brackets) the number out of the query result that is included in our review of the state-of-the-art after checking the criteria listed in Table 2. While the SASO conference is the main venue for SASO approaches, the authors were not able to identify explicit papers relevant to self-integrating systems using the outlined methodology – which may be explained by the fact that the corresponding workshops (such as the SISSY workshop) have been mainly part of the SASO conferences within the last decade and consequently provided a dedicated forum for these topics.

To select studies suitable for analysis, we considered three inclusion criteria (IC) and three exclusion criteria (EC); see Table 2. A study is included for review if it satisfies all IC and no EC.

During the review process of the queried papers and articles, we analysed the major contributions regarding SISSY technology. 82 out of 187 papers and articles have been selected for further consideration. Here, we classified the paper as belonging to one of the five major categories 1) theory, 2) system, 3) integration, 4) improvement, and 5) consciousness. These

Table 1: List of included venues

| Name | Venue | Publisher | # Studies |
|---|---|---|---|
| ICAC | Conference | IEEE | 19 (15) |
| DASC | Conference | IEEE | 36 (7) |
| SASO | Conference | IEEE | 0 (0) |
| ATC | Conference | IEEE | 25 (11) |
| ICAS | Conference | IEEE | 4 (2) |
| EASE | Conference/Workshop | IEEE | 9 (3) |
| SISSY | Workshop | IEEE/ACM | 53 (31) |
| ACW | Workshop | IEEE | 3 (0) |
| TAAS | Journal | ACM | 37 (12) |
| TCPS | Journal | ACM | 1 (1) |

Table 2: List of inclusion and exclusion criteria for the analysis

| ID | Criterion | # Studies |
|---|---|---|
| **Inclusion criteria:** | | |
| IC1 | The paper is concerned with (self-)integration behaviour | 103 |
| IC2 | The system involves multiple agents | 129 |
| IC3 | At least one agent adapts its integration status at run-time | 100 |
| **Exclusion criteria:** | | |
| EC1 | The integration behaviour is not described in the paper | 76 |
| EC2 | The paper is a glossary, extended abstract, tutorial, demo, etc. | 11 |
| EC3 | A more complete version of the paper is selected for review | 65 |

major categories have been further refined in terms of 16 sub-categories as discussed in the following paragraphs. Furthermore, we kept track of the application scenarios used in the papers. The next sub-section summarises the findings and the corresponding state of the art according to these categories. The following list introduces the sub-categories:

- Theory: This category comprises research efforts that are dedicated to a definition and terminology for SISSY systems as well as examples for specific problem classes. It contains the following two sub-categories:
  - Definition and terminology
  - Problem classes and examples
- System: This category comprises research efforts that are dedicated to designing and developing the overall SISSY infrastructure. The corresponding papers are classified according to the following six sub-categories:
  - Hardware aspects and solutions
  - Design concepts and architectures for SISSY systems
  - Metrics to assess the integration status
  - Membership and participation mechanisms
  - Testing, validating and provisioning of testbeds
  - Communication among the SISSY subsystems
- Integration: This category comprises research efforts that are dedicated to performing the integration decisions of the autonomous SISSY subsystems. The corresponding papers are classified according to the following three sub-categories:
  - Abstraction
  - Resilience
  - Adaptation
- Improvement: This category comprises research efforts that are dedicated to improving the integration decisions over time. The corresponding papers are classified according to the time horizon using the following two sub-categories:
  - Behaviour learning
  - Optimisation / planning
- Consciousness: This category comprises research efforts that are dedicated to learning mechanisms that are not directly related to integration decisions. The corresponding papers are classified according the following three sub-categories:
  - Self-awareness (current conditions and states)
  - Retrospection (forensics of past behaviours and developments)
  - Awareness of the environment and others

### *4.2. Review of the current state of the art*

In this section, we briefly summarise the current state of the art in the categories defined in the previous section. This then serves as a basis to identify gaps of research to establish fully-fledged SISSY systems.

#### *4.2.1. Theory of SISSY systems*

A considerable part of the ongoing research in the field aims at defining and formalising the fundamental concepts and terminology behind SISSY systems. In that connection, a series of problem classes can be derived from the set of example scenarios discussed in the literature.

*a) Definitions:* Self-improving System Integration (SISSY) itself is defined in [37] as the "attempt for the intelligent integration of complex systems that have both adaptive and non-adaptive components or subsystems". More specifically, the paper delineates self-integration as an "ongoing autonomous process for linking a potentially large set of heterogeneous computing systems, devices, and software applications; to meet system goals", and self-improvement as "inventing solutions where none are available and optimising the selection of solutions over time" (which we took as basis for Section 2.
Conceptually related are Computational Self-Awareness [38, 39] and Self-Reflection [40]. This is the ability of a system to learn about itself on an ongoing basis, in order to reason

about and improve its own behaviour in a complex environment. Complexity factors typically considered include scale, uncertainty, environmental dynamics, heterogeneity, decentralisation, and the possibility of situations arising that may mot have been anticipated at design-time. As such, it substantiates a more proactive notion as opposed to purely reactive approaches as, for instance, in regular self-adaptation. [41] further distinguishes different levels of networked self-awareness: Networked Stimulus-awareness (i.e., the perception of intrinsic and extrinsic stimuli impacting the system), Networked Interaction-awareness (i.e., the reasoning about a system's own interactions with the environment inclusive of other systems), Networked Time-awareness (i.e., a system is aware of how its environment is susceptible to change over time), Networked Goal-awareness (i.e., the reasoning about and identification of the goals of other ), and Networked Meta-self-awareness (i.e., the system's ability to determine its own level of self-awareness).

**Assessment and research gap:** These and other contributions constitute substantial efforts in theorising about the necessary concepts underlying SISSY systems, working towards a unified terminology and theoretical "superstructure" which is within reach. However, there are still many hard problems which remain open or are vague [42, 43, 44].

*b) Problem classes:* The application scenarios underlying the various theoretical considerations range from autonomous vehicles [42] over vehicular traffic control [45] and route guidance [46], robotics and simulation [40], emergency response [47], to intelligent surveillance systems [44] and energy systems [48]. From these contributions, together with those that take a more generalist system-of-systems stance [6, 43, 37] or that involve specific interaction models, e.g., [41], we can extract the following core problem classes:

- *Temporality*: the continual integration versus temporary materialisation of a system-of-systems, as well as the timeliness of the reflection and observation mechanisms within the individual systems and their constituents.
- *Predictability*: handling both expected and unexpected integration.
- *Intentionality*: cooperation/collaboration and goal-driven behaviour, alongside derivation of intent.
- *Dependability*: employing computational trust models, handling noise and ensuring sufficient situatedness/locality.
- *Uncertainty*: identifying knowledge gaps to handle unknowns and unknown events, plus reliable skill mapping.

To some extent, different problem classes can overlap. For example, goal orientation is directly related to dependability issues. Most of the time, however, these problem classes directly influence each other, necessitating a careful balance so as to not jeopardise the overall system integration.

**Assessment and research gap:** While these problem classes are indicative, many of their details remain to be examined for their formal understanding and full applicability across domains. The layered system model proposed in [40] constitutes a step in that direction, differentiating between the abstract layers Reaction, Adaptation, Reflection and Collaboration in a heterogeneous set of scenarios. Similarly, [42] identifies the following central challenges that need to be tackled before successful self-integration becomes truly viable: resolution for concepts, temporal planning, trackable events, and merging.
Many of the proposals for tackling the different problem classes need to be corroborated, e.g, with their applicability remaining to be validated and qualitatively assessed [44]. Therefore, a common test environment to allow for different approaches and systems to be deployed, assessed and compared on a shared platform for the development of self-integrating systems, as discussed in [49], is central to SISSY systems research.

#### *4.2.2. System aspects*

*a) Architecture:* Architectures are usually introduced to manage distributed (service-oriented [50]) software maintenance [51], wireless sensor networks [52], embedded systems [53], clouds [54] as well as infrastructures, for instance, Smart Grids [55, 56]. Architectural aspects on emerging coordination [57] and self-improving interweaving [58] in Autonomous Adaptive System are also studied.

**Assessment and research gap:** Architectures that facilitate run-time self-integration of multi-domain decentralised systems are still subject of research. However, the basic adaptation principles and corresponding architectures seem to be mostly reusable for altering and optimising the integration states.

*b) Hardware:* Due to the scope of the query (i.e., the venues listed in Table 1), hardware aspects are less prominently covered by this study. However, there is a lot of work on and innovation in hardware integration, especially with GPUs and FPGAs, and the chosen venues do not obviously capture this work – we just consider those that have been intentionally presented in the context of SASO-based self-integration. Here, contributions range from re-configurable embedded system architecture [59] to integrated neural networks for cooperative multi-robot swarms [60]. Design implications based on the viable systems model for software maintenance are also discussed earlier [51].

**Assessment and research gap:** There is still a significant gap between integration solutions for hardware and software reflected by different fragmented communities. For instance, the pervasiveness of the Internet of Things promises further joint efforts on self-integrating both software and hardware systems.

*c) Metrics:* The systematic review of the literature reveals that studies concerning self-integration are mainly assessed with qualitative methods. However, metrics have been proposed for assessing the integration of data quality in data warehouse development. This can be combined with metrics quantifying a degree of self-adaptation [61] or self-organisation [21] for the basic SASO capabilities. The recent effort focuses on determining measurable qualities for system integration status [19].

**Assessment and research gap:** Benchmarks for assessing the functionality of self-integration are still a subject of ongoing

research. Baselines and use cases are required to validate the effectiveness of proposed metrics.

*d) Membership:* The heterogeneity of pervasive and ubiquitous systems poses several challenges for the management of membership and participation in self-integrating systems [62, 63, 64, 65]. For instance, access control mechanisms and cloud computing architectures for social Internet of Things [66, 67, 64], communication protocols for wearable devices [68] and adaptive protocols [69, 52] as well as semantic representation and ontologies resource access and management [70, 71, 72].

**Assessment and research gap:** Decentralised approaches for membership management and participation increase the complexity of establishing self-integration. Such design approaches usually rely on agents with a partial or incomplete view of the system to preserve privacy, autonomy and locality. Making distributed optimisation and learning mechanisms work effectively under such socio-technical constraints [73] is a subject of ongoing work [74, 75, 76].

*e) Test:* System self-integration poses particular testing and assessment challenges. Testbeds for cyber-physical systems [77], UAVs [78] and Smart Grids [79] have been introduced that are mainly domain-dependent. More general-purpose approaches are limited to bio-inspired high performance computing simulation [80] for self-building embedded systems as well as conceptual architectures [49, 81].

**Assessment and research gap:** Domain-independent IoT testbed solutions that allow the self-integration of different distributed services and IoT applications are missing. In particular, testbeds that could facilitate the seamless prototyping from simulation to live deployments and operation at high technology readiness level (TRL).

*f) Communication:* It is used as means of self-integration of different interacting [82] or migrating [83] data/system components, for instance Smart Grids [84], or as a domain of application of self-integration solution, for instance wireless [85, 86] and mobile 5G networks [87, 88].

**Assessment and research gap:** Software abstractions are not always aligned with communication and protocol abstractions, especially in large-scale distributed systems. Making existing solutions [89] applicable in heterogeneous IoT system is an open challenge.

#### *4.2.3. Integration aspects*

During the analysis of the queried papers, we identified three categories of contributions regarding the actual integration mechanisms: 1) Concepts and techniques for the abstraction of the particular resources and subsystems including legacy systems, 2) mechanisms for establishing resilient behaviour and integration states, and 3) techniques for performing adaptations of the integration status.

*a) Abstraction:* Self-integration requires a certain encapsulation of the autonomous subsystems, including those not running SISSY technology. Most of these abstraction efforts have been described in the context of establishing a so-called "Wrappings" mechanism. The basis of this approach is to equip systems with capabilities that allow for a self-integration and unsupervised cooperation of system components and their services. This includes adding a so-called "Brain Patch" [47] to legacy systems, allowing their efficient integration in large-scale SISSY systems.

The "Wrappings" concept is based on processing explicit qualitative information about all system components and their interconnections. These systems are complex and interacting collections of components [90] and are composed of heterogeneous processes, they are subject to hard and possibly unknown requirements and they must function in complex environments (such as space missions). Consequently, the design and development process for such systems require explicit models of their behaviour, a specific architecture, and models of the environment in which these systems are expected to operate [91]. Furthermore, suitably flexible computer-based design support is needed.

**Assessment and research gap:** The technology has been theoretically discussed and tested in several small-scale testbeds. However, a transfer to large-scale SISSY systems and corresponding proof of scalability and robustness are still open.

*b) Resilience:* Self-integration is typically done in open system constellations where autonomous systems are free to join and leave the overall system at any time. thus subsystems are typically unknown and their behaviour can not necessarily be anticipated. In particular, malicious elements may become part of the system and attempt to exploit it. As a countermeasure, two basic concepts have been discussed in detail: computational trust and security.

In the context of security, compliance with security requirements in multi-tenant applications [92] has been investigated. Furthermore, the need for mutually testing the correct functioning of subsystems has been highlighted in [93] – extending security mechanisms towards a continuous testing solution. Furthermore, security challenges specific for SISSY systems have been identified in [94].

In the context of computational trust, a concept for establishing explicit communities of mutually trusted subsystems and isolating malicious or faulty subsystems has been presented [95]. Based on this initial concept, several extensions have been investigated: Mechanisms for measuring trust and reputation [96], advanced attacks on such trust communities and corresponding countermeasures [97], accusation-based strategies to identify misbehaving subsystems and to establish forgiveness solutions [98], and robust self-monitoring mechanisms at run-time [99]. Furthermore, [100] demonstrated that computational trust and forgiveness techniques can result in improved reliability and reduced overhead.

**Assessment and research gap:** Most of the security issues of self-integrating systems can be addressed by standard technology. However, some challenges remain unsolved, such as how attacks against mechanisms such as mutual influence detection or collective awareness technology can be circumvented. In addition, computational trust is currently just considered as a single value assessment of each particular entity. A more sophisticated trust and reputation system will cover more aspects including reliability and collusion with others, possibly combined with proactive alarms using predictions of trends.

*c) Adaptation:* The actual change of the integration status

is done by adaptation mechanisms that typically follow the Observer/Controller patterns [101] from Organic Computing [2] or the Monitor-Analyse-Plan-Execute(-Knowledge) patterns from Autonomic Computing [9] (see, e.g., [102] or [103]). Alternatives include the system in [104], which describes a simple reactive adaptation logic without learning, or the approach from [105], which compares different decision mechanism, mostly including learning techniques such as artificial neural networks or reinforcement learning [106].

**Assessment and research gap:** Most of the adaptation issues of self-integrating systems seem to be addressable by means of standard mechanism for self-adaptation. However, as outlined in [19], the assessment of the integration status is usually a multi-criteria problem. Open questions include how to incorporate multiple, potentially conflicting goals in the decisions logic, how to prioritise dynamically between goals, and how to integrate user guidance efficiently in this process.

#### *4.2.4. Improvement aspects*

*Self-Improvement* constitutes the second key ingredient for SISSY systems as defined above. The term "improvement" here refers to the ability of the individual subsystems or else the collective interwoven system-of-systems to continually optimise for a better integration behaviour.

This work is concerned with utilising learning mechanisms to improve the integration status at run-time. To our surprise, as of yet, it has seldom been considered in the literature. Nonetheless, a few works could be identified. For instance, Rudolph et al. in [3] propose a generic approach to automatically detect *mutual influences* among multiple agents which share a common environment. The influences are derived based upon the calculation of statistical measures which correlate the individual components of an agent's configuration vector with the performance measure of another agent under consideration. If the positive or negative correlation is detected, mutual influence is assumed between those agents. In consequence it is proposed that those configuration vector components of the agent for which influence could be detected are integrated into the state vector of the actual impacted agent, resulting in the necessity of a state-space adaptation at run-time. Based on that approach, Stein and Tomforde [44] mapped the resulting challenges regarding the underlying machine learning mechanisms to the concept of *transfer learning* and identified the capability of SISSY systems to cope with these issues as crucial. As another subbranch of machine learning, the explicit use of ensemble methods have been suggested to enhance the self-improvement capability of SISSY systems [107, 108]. Krupitzer et al. present a self-learning analyser for time-series forecasting with their SATISFy architecture [107]. Their approach aims at enhancing the proactive adaptation capabilities of autonomic computing systems – a capability which is indisputably necessary for SISSY systems as well. A second ensemble approach is presented by Deist et al. in [108]. Their *Coopetitive Soft Gating Ensemble* promises to self-improve its learning capabilities at the system's run-time using previously seen data based on a human-comprehensible combination scheme that incorporates global, local, and time-dependent weighting factors.

Learning in systems is often concerned with the automatic acquisition of actionable knowledge. SISSY systems are expected to act in dynamically changing and uncertain environments where not all system states can be anticipated *a-priori*. There will therefore be knowledge gaps that challenge SISSY systems [37]. A formal definition of knowledge gaps, as well as a concept for proactive construction of knowledge elements, is presented in [109]. A proactive identification of uncertain regions or incomplete knowledge within a self-improving system's knowledge base is identified as essential for building viable self-learning systems employing run-time learning mechanisms.

A more technical work that uses similarity rank list comparison to integrate different cues in the context of object recognition is provided by Grieben and Würtz in [110]. Clearly, in the era of deep learning-based computer vision, methodologies for robust object recognition despite different viewing angles and varying illuminations are important to allow for a robust usage of vision sensors in autonomous systems employed to carry out safety-critical tasks.

Last but not least, the work of Wang et al. [111] has been identified to fulfil the above-mentioned criteria. In their work, reinforcement learning techniques are combined with multi-agent technology in the context of self-adaptive service composition. The motivation of evolving services in the course of systems lifetime and the resulting highly dynamic learning environment perfectly matches the challenges with which SISSY systems are expected to be confronted. Multi-agent reinforcement learning methodologies, as proposed in their work, should be in the repertoire of learning techniques used in SISSY systems.

**Assessment and research gap:** An important direction for research on self-integrating systems is how to enable them to share knowledge, even in the presence of heterogeneity of the underlying learning mechanisms. The negotiation of a common knowledge model to consolidate knowledge artifacts into exchangeable information seems to be a viable approach to be explored. Also, approaches using the collective of integrated systems for closing identified knowledge gaps [37, 109] seem particularly important for building resilient systems.

To use mutual influence detection as proposed by Rudolph et al. [3], we require much faster evaluation. An approach to achieving this relies on direct information exchange between neighbouring systems. Such exchange would avoid the communication overhead and the magnitude of feature space for learning models and, in turn, limit the required computational capacity.

A third direction in which further research is required should explicitly emphasise the relation to multi-criteria optimisation. However, SISSY systems are required to deal with continual change in the underlying solution (or fitness) landscapes. Each time the integration status of the SISSY system changes – with either global or local scope – the (sub-)systems which are most immediately impacted might also face an abrupt shift of their solution space. Even worse, each reconfiguration of a (sub-)system itself might entail further changes in the landscape, a phenomenon which is referred to as *self-referential fitness land-*

*scapes* in the Organic Computing [112] context. SISSY systems need to detect or even anticipate such situations. In doing so, they will need to use creative components to proactively generate new solutions autonomously in order to attenuate utility degradation as much as possible. However, systems not only integrate in order to collaborate towards a common goal but also to cooperate when operating towards the individual, potentially conflicting goals. Optimisation methods able to deal with both the changing fitness landscape as well as the potentially conflicting goals are undoubtedly key for the success of future SISSY systems.

#### *4.2.5. Consciousness aspects*

To self-improve and self-integrate, systems need to have an understanding of themselves, other systems in the environment, and the environment in which they are embedded. We therefore separated the reviewed literature for references to consciousness in self-improving systems integration into three different areas, 1) awareness of oneself, 2) retrospection of past behaviours and developments, and 3) awareness of the environment and others.

*a) Awareness of the current* self*:* For a system to be able to improve its own performance, it needs to be aware of its current state and behaviour. In most primitive forms, this awareness is hard-coded in the autonomous software agent [22]. The agent is not aware of its performance and state *per se* but has its goals ingrained and operates using a utility function. This utility function allows the system to make decisions based on its most recent actions and current state: They identify and assess their current state, their behaviour, and their most recent actions [40, 47]. The "Wrappings" approach [113] which was discussed previously has specifically been designed for achieving this task.

Self-aware computing systems have also been proposed [114]. While they are not specifically designed to self-integrate or self-improve, they are designed to learn about themselves and their environment on an ongoing basis and use and reason about these models operate towards high-level goals [10]. While there is a plethora of algorithms implementing learning techniques for autonomous agents [115, 116], systems also require an understanding of what knowledge they are missing [117]. This awareness can enable systems to probe their environment and actions purposefully, allowing them to expediently close their knowledge gaps.

To achieve successful integration, the actions of the individual systems need to be planned. Frank [118] proposed a reflective planning approach in which the systems abstract knowledge about most recent actions and their results. If new actions did not achieve the desired outcome, the abstracted information is compared in order to find discrepancies.

**Assessment and research gap:** While there have been great advances in computational self-awareness, the challenge remains of how to satisfy user requirements and provide corresponding guarantees while the system is able to autonomously change its behaviour, goals, and interactions. Furthermore, making systems aware of their abilities, resources, interactions, and, even more so, their awareness, requires additional resources to handle this information. A trade-off arises for the available resources to be used to establish awareness about the *self* and using those resources to operate towards fulfilling user requirements.

*b) Awareness of the environment and others:* Instead of considering only awareness of the *self* and previous behaviour, there have been efforts at explicitly considering others and environments. Esterle and Brown [119, 41] take inspiration from the development of infants and how they explore and experience the world as described by Vygotsky and Piaget. They propose different levels of networked self-awareness: awareness about interactions, about goals, and about actions of others in the environment which have an effect on themselves. Gruhl et al. [120, 121] proposed a modular novelty detection framework that applies different detectors on-demand distinguishing between regions of the input space with a high density of expected observations and those with low density. This framework can be used to detect changes in the behaviour of the environment and others to trigger adaptations. To detect the interference with others, Rudolph et al. [122] propose an approach based on local measurement. As discussed by Barnes et al. [123], assigning agency to these interferences might be key in order to operate and integrate accordingly. The question remains of how to deal with phenomena to which a system cannot clearly assign a causing entity, and whether this inability would lead to superstition and superstitious behaviour in the system.

Autonomous, self-integrating machines also must be aware of the trust of users and operators [124]. Furthermore, they also need to trust the other devices with which the system is integrating. When individual agents trust each other they can group together in trust communities, allowing them to speed up their different processes and interactions among each other [97]. Here, each participant of a trust community is aware of its membership and able to make decisions relating to other members.

**Assessment and research gap:** The question remains of how to enable systems to model behaviours of others through observations. Even when activity traces are available, mapping the actions to specific behaviour and specific goals has not yet been studied. Verifying this behaviour modelling during runtime, even when actions and corresponding goals of a system are known to the observer, creates an additional challenge for managing available resources.

*c) Retrospection of past behaviours and developments:* In contrast to only looking at the current state of affairs, systems continuously improving their integration must reflect on their previous performance. Formal methods can be used to trace the interactions between individual systems and software components [125]. Collecting this information centrally allows validation of system integration. However, in many cases, this information is not shared among the systems and a central solution is not viable. One possible approach to addressing this challenge is to create federations over integrated systems, collecting information over all partners of the corresponding domain [126]. Since systems usually have no awareness of other domains, integrating them across different domains then becomes very challenging. We therefore require an awareness of others.

Utilising these traces of previous actions allows not only for validation of previous actions but can also aid and guide decision processes [127]. Here, historical traces are used to enable the systems to explain their previous decisions, interact with users and support decision making processes, and rely on the previous experiences to make future decisions whether they are recommended to a user or made autonomously by the system.

For autonomously taken decisions, operators will require (security) guarantees that the system performs as expected and will behave according to its specifications. While fault detection, isolation and recovery (FDIR) allow the system to deal with faults during run-time and integration, both self-protection mechanisms and an awareness of its own minimal acceptable performance are required in order to successfully integrate with other systems [33]. An alternative approach to providing individual guarantees with corresponding self-evaluation and fallback mechanisms considers collectives to adhere to security requirements as a whole. Instead of defining a rigid set of security requirements, individual (software) systems can be enabled to adapt internal security requirements while keeping their offered services and interactions [92]. This requires the interacting and integrating systems to have an awareness of each other and their inherent procedures and their previous behaviour.

**Assessment and research gap:** Retrospection of past behaviour, actions, and states of the environment and the *self* requires an awareness of the current states of these aspects. In addition, analysing the states and actions in an appropriate amount of time with reasonable accuracy remains a challenge.

### *4.3. Taxonomy of SISSY contributions*

In our review process, we also annotated the different contributions with tags dependent on their category. If necessary, we assigned multiple such tags, i.e., we assigned them a primary category and potentially additional, secondary categories. Of those research papers eligible for our study, some are classified with more than one out of the 16 available labels. We analyse the relations between various research fields within SISSY utilising the frequency of the resulting classification pairs.

The graph depicted in Fig. 1 shows these relations. The width of the edges is proportional to the observed frequencies, i.e., how often two (sub-)categories are considered by the same contribution. On the other hand, the size of the vertices representing each sub-category indicates the total number of contributions that have been classified to describe research in this category. This graph provides an overview of how contributions are interrelated and connected to each other. Correspondingly, the graph represents a taxonomy generated from our classification of the considered research papers.

The graph reveals that most of the research efforts fall into the category of "integration" (i.e., adaptation, abstraction, and resilience), followed by the three aspects architectures, membership, and communication from the "system" category. Furthermore, there are strong connections between abstraction, architectures, membership, and adaptation, meaning that they are usually considered together. This also takes into consideration various resilience mechanisms. Especially, already considering adaptation, abstract, and resilience mechanisms in the design and architecture supports the observation that the basic technology for establishing SISSY behaviour is already in place.

In turn, hardware aspects and system metrics are rarely covered and only loosely connected with the other efforts. This demonstrates that a more in-depth study of these aspects within the other major research directions is probably needed. Similar observations can be made for awareness, reflection and retrospection from the consciousness category. Obviously, these efforts are not well merged into the main research strands.

This analysis and the corresponding graph representation just summarises the already existing directions of research. Following the discussions of what SISSY systems are (see Section 2 and what we want to achieve with "Integration Science" (see Section 3), there are more fundamental insights that are required to completely realise self-improving self-integrating technology, and to enable this to become an integral part of practical applications. The next section presents a roadmap, utilising this current state of SISSY development as a starting point, which outlines the fundamental gaps which need to be addressed by research in the near future.

## 5. Research Roadmap

As with all technical roadmaps, we start with what we currently can do and see whether we can develop applications that start to approach the SISSY vision. This is clearly seen in the research overview presented above. The lines of current research demonstrate that SISSY has incorporated learning, trust among agents, peer-to-peer negotiation or mediation through a third party as in the Brain Patch, and so forth. What has been shown is encouraging — there are a lot of modest applications waiting for SISSY approaches. But our gains have also shown how far we must go. As an example of what remains before we reach the vision of SISSY, we identify three special challenges: true developmental processes, online trade-offs and negotiation among competing goals, and the development of semantically rich languages for SISSY modelling and communication. For each one, we sketch out our current starting point and some landmarks that will help us assess progress.

### *5.1. True Developmental Processes*

A bit provocatively, we have used the term "true developmental" processes in order to place the emphasis on not just the enlargement or growth of a system in terms of the number or size of its components but to emphasise innovation, novelty, and new uses for new processes and new behaviours. The most successful adaptations right now have been the adaptation of known parameters and behaviours to known goals within fairly well-known operational environments.

As important as these successes are, in SISSY we confront the problem of systems integrating with possibly unknown systems and components within dynamically changing environments. This means that many aspects of the system integration may change, including usage within a known operating environment, the parameters used for or by components at multiple levels of the systems; the goals and system objectives;

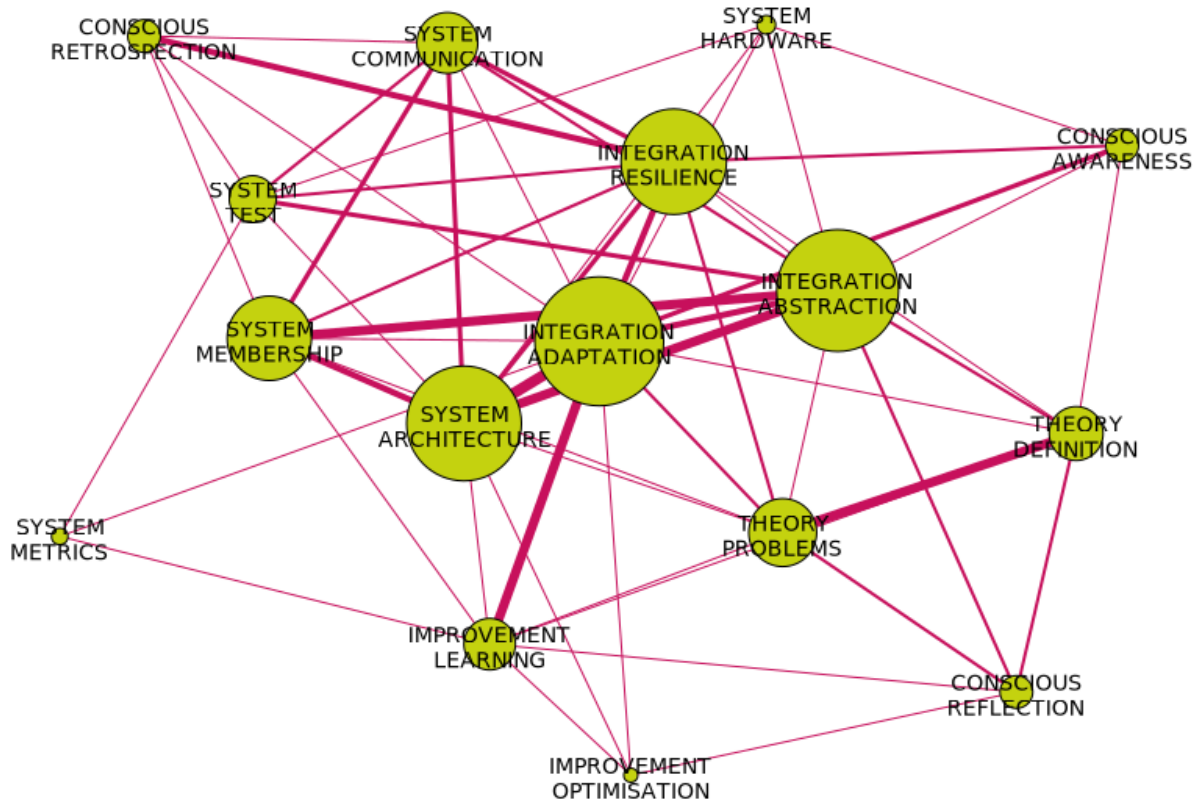


Figure 1: SISSY taxonomy representing relations between the various research fields.

the behaviours and processes. Some of these changes could be adding or making use of previously unused parameters and components in a given configuration or usage within an operational environment. Some of these changes could be subtle timing and ordering changes in configurations which lead to novel behaviours (for example, some neural locomotion pattern generators show how a single pattern generator can lead to strikingly different gaits in diverse animals). Some of these changes could be the creation of new manipulable parameters, new behaviours, new configurations, and even new components through merging and other combinatorial processes.

Our current starting point is the fruitful research and development of methods and systems that can enlarge and expand themselves (e.g., intelligent databases or peer-to-peer sensor networks). Building growing systems that can comfortably scale is an important area of research. Much of the self-adaptation in such systems is geared towards adjusting needed organisation and management in such systems as they grow [128]. There is even the modest ability to add new categories and new features in the organisation of such systems. Systems currently can add new features to, for example, a learning algorithm.

In ongoing new research, SISSY groups have started to collaborate on how they could demonstrate the ability to add and then improve new joint actions of different systems (for example, how to use helicopters and cars together in a common search mission.) As we approach such work, it is important to keep one of D'Arcy Thompson's great insights — that in development one has several developmental processes (or systems) that are developing in parallel and that they come together or in his terms "are moulded one with another; they come into being together and act and react together. They are parts of a whole which, when it loses its composite integrity, ceases to exist." [129]. This is strikingly different from the independence we purposely engineer into components and systems in order to make them reusable in different ways. Is there something that we can use from development processes that does not compromise our important ways of engineering manageable systems? We think the following steps are doable.

Landmark 1 provides methods for generating and examining new combinations of the components from different systems. Merging is an example within animal systems and has been suggested for engineered systems [130]. Allowable combinations can provide a foundation for SISSY negotiations.

Landmark 2 sets up a more sophisticated directed search for new configurations that would resolve needs or gaps in current capabilities. Doing so requires more sophisticated modelling and reasoning capabilities than currently exist.

Landmark 3 sets up generative processes as a first level of

developmental processes to churn out new combinations and configurations as in morphogenesis or the random exploratory behaviour seen in certain animals in a new environment. These processes are of no use to a system without another filtering, evaluative, and reasoning processes that have some criteria for becoming interested in these patterns or a fitness landscape that favours certain combinations. We suggest that these generative processes would need to be constrained or held in check by maintenance processes that delete deleterious new growths and promote new combinations which yield promising results.

Landmark 4 would be the development of balancing processes that stabilise a system's growth and development by having a judicious combination of developmental processes and maintenance processes that serve to preserve the already working solutions of the system. *Balancing processes* are checks on exploratory development and also are stabilising processes. They are just one example of how biological systems constantly make use of competing forces (e.g., extension and flexion, the balance of neurotransmitters in the brain, feedback among excitatory and inhibitory pathways, etc.) to create control processes that can respond quickly by adjusting the balance of such oppositions. This leads directly into a next major challenge, trade-offs and compromises.

### *5.2. Trade-offs*

SISSY involves heterogeneous systems pulled together by circumstances, interests (looking for specific new capabilities) or intent (a city orders all its major systems to be integrated.) In classical design, trade-offs among partially overlapping and partially competing systems, components or behaviours are very common. In SISSY the resolution of such potential conflicts and resource contentions needs to be done by the systems themselves with perhaps some oversight by human operators and developers. Integrating the behaviours and goals of several systems will, of course, involve compromises. Just as a single system cannot do everything it could at any one time, so a new conjoining of systems that will hopefully lead to new capabilities cannot do everything at once. How these compromises are done so as to ensure continued viability [131] — that is effective functioning — of the newly integrated system and yet progress towards some new goals and performance is the challenge [132]. What is the knowledge that is needed to make these trade-offs? Again, in classical design, these trade-offs are often lengthy sets of experiments and analyses done by human developers. How much can be done during the design process to set up the SISSY systems and how much can be done at runtime by the SISSY systems? The starting point for trade-offs is that we give the SISSY systems self-models that include explicit information on what parameters, aspects, and behaviours can be compromised and which ones cannot. It thus will have minimal criteria for all parameters that must be protected to ensure essential system behaviours.

Landmark 1: Reasoning processes take into account the goals of multiple agents.

Landmark 2: Self-optimisation methods can operate over several agents.

Landmark 3: One can demonstrate in negotiations the ability to relax requirements and reach a common plan.

### *5.3. Language: Overall goal is shareable communications*

Holding all of this together is a growth in language capabilities – not only communicating to other human and computational agents but also semantically representing what is understood and done in a shareable manipulable computational form. Let us look at communication and transaction first.

Not only must a system be able to reason about novel opportunities and trade-offs, but it must be able to communicate that to others who may have different language and communication capabilities. In biological systems, animals intentionally and unintentionally alert and inform other animals by their shrieks, emotional utterances, and behaviours. When we talk to each other we point, show, demonstrate, and indicate in many manners, information to others, as well as verbally talking. In computer science, we have, through protocols of many types, shown that rule-based transactions with others can occur through a variety of means. Also, by recognising and organising the environment, we bypass a verbal common language with situated but meaningful behavioural indicators. Hence when I am in Budapest and don't speak the language, I can enter a bakery and point to what I want and open my handful of coins, letting the baker take whatever is necessary. So, part of our engineering in SISSY systems needs to consider how we design the operational environment so that participating systems could possibly make use of situated transactions not dependent upon messages passed. We have certainly done that already for some Internet applications. We have also suggested special safe spaces "integration playgrounds" [133, 13] where systems can experiment with new configurations and study the capabilities of other systems. Here of course trust enters heavily as a key research area as players and partners come and go. These safe places become a key part of our active experimentation and where specialised agents such as the Brain Patch [47] can learn about new participating systems.

But communication, as critical as it is, is not the only reason for increasing language capabilities. Having meaningful reasoning with semantics rich enough to capture the behaviours of a complex system is critical for modelling, summarising what is relevant to different reasoning processes, and conveying those understandings and information to other systems to coordinate joint behaviour.

One of the significant challenges for language capabilities is to invent descriptive terms and information about newly discovered objects, patterns, others and so forth. In the Cal Poly Pomona testbed of robotic cars [77], algorithms used to track a ball were unpredictably doing poorly or well. Eventually, the human developers figured out that it was the wind blowing the light ball in unexpected ways. None of the numerous sensors had been geared to look for that feature. What would allow the systems to first of all show what the problem was (how their results were inconsistent) and then as they experimented over days, name the novel factor and describe it?

Landmark 1: The starting point is rich domain-specific languages, pre-determined at design-time, with some new capa-

bilities for assigning arbitrary terms to identified gaps or novel patterns.

Landmark 2: Language capable of accompanying active experimentation is key, as well as realising that such language goes hand in hand with modelling and reasoning processes. That is, one needs to understand what one is trying to do with the new language abilities. Hence the second landmark is to first analyse what information is needed for new behaviours, goals, players (and hence what must be represented) and what new transactions that information is supposed to support. Then these new placeholder terms must be integrated into the domain-specific languages.

Landmark 3: Lastly one of the biggest challenges for SISSY (as too in science) is the reconciling of different words, terms, and subcultures that have overlapping meanings. In other words, we must integrate the language used. Landmark 3 is a demonstration of being able to combine and link terms among multiple agents.

In Section 3 we introduced five short scenarios that partly differ from the traditional scenarios used in literature: joining a game, joining a caravan of vehicles, joining a search, joining a research project, and joining a surveillance system. Considering the landmarks outlined above, we can see that they are particularly important in the context of these scenarios.

Landmark 1: Rich domain-specific languages are required to allow for on-demand adaptive platooning scenarios (i.e., joining a caravan of vehicles) including semantic context for sensor readings and manoeuvres. In contrast, the characteristics to be covered by joining a surveillance system will differ for the surveillance-oriented part, for instance. However, some operations remain standard self-integration behaviour (e.g., general relations to and interactions with others). The novel patterns to be identified will differ depending on being a game, a research project, or a surveillance system.

Landmark 2: A similar observation can be done for active experimentation - the specific behaviour is application-dependent, but the overall approach follows the same path of integration. For instance, a set of neighbouring smart cameras in a surveillance system may decide to focus on an object highlighted by an artificially generated light-flash to better calibrate their topology models. In contrast, such experimentation in the vehicle caravan example would be to test if waiting for a caravan with approximately the same direction brings benefits – and to use this experience in optimising the decisions in the future. Similar strategies can be found in the other scenarios as well – where a direct link to the exploration-dilemma in autonomous learning systems is visible.

Landmark 3: An example of such different meanings of similar terms is 'shooting' - in a game of robot soccer players, this is the main task of playing with the ball. Consequently, it comes with a positive connotation and does not necessarily need a response if detected. However, a rescue team observing a shot will immediately activate emergency measures – meaning that this comes with a strongly negative connotation.

In summary we have given a quick overview of three daunting challenges in SISSY with first steps and what some of the landmarks would look like for assessing progress.

## 6. Conclusion

The research initiative "self-improving system integration" (SISSY) was established with the goal of mastering the ever-changing demands of system organisation in the presence of autonomous subsystems, evolving architectures, and highly-dynamic open environments. It aims to move integration-related decisions from design-time to run-time – implying a further shift of expertise and responsibility from human engineers to autonomic systems. This introduces a qualitative shift from existing SASO systems, moving from self-adaptation to predefined variation types towards more open contexts involving novel autonomous subsystems, collaborative behaviours and emerging goals.

In order to understand current progress towards the SISSY vision, we revisit existing SISSY research efforts and establish a corresponding terminology focusing on how SISSY relates to the overall field of integration sciences. We then investigate SISSY-related research efforts and derive a taxonomy of SISSY technology. During the review process of the queried papers and articles, we analysed the major contributions regarding SISSY technology. 82 out of 187 papers and articles have been selected for further consideration. Here, we classified the paper as belonging to one of the five major categories 1) theory, 2) system, 3) integration, 4) improvement, and 5) consciousness. These major categories have been further refined in terms of 16 sub-categories as discussed in the paper. Furthermore, we kept track of the application scenarios used in the papers. We summarised the findings and the corresponding state of the art according to these categories.

As an example of what remains before we reach the vision of SISSY, we pulled out three special challenges: creating true developmental processes, producing methods for online trade-offs and negotiation among competing goals, and the development of semantically rich languages for SISSY modelling and communication. For each one, we sketch out our current starting point, which includes much of the research discussed above, and some landmarks that will help us assess progress. This is a promising area of research and development, with some substantial advancements and yet a long way to go. We look forward to collaborating with researchers from many diverse fields of science and engineering.

## References


[1] S. Tomforde, C. Müller-Schloer, Incremental Design of Adaptive Systems, Journal of Ambient Intelligence and Smart Environments 6 (2014) 179–198.

[2] C. Müller-Schloer, S. Tomforde, Organic Computing – Technical Systems for Survival in the Real World, Autonomic Systems, Birkhäuser Verlag, 2017, iSBN: 978-3-319-68476-5.

[3] S. Rudolph, S. Tomforde, J. Hähner, Mutual Influence-Aware Runtime Learning of Self-Adaptation Behavior, ACM Trans. Auton. Adapt. Syst. 14 (1) (Sep. 2019). doi:10.1145/3345319.
URL https://doi.org/10.1145/3345319

[4] K. L. Bellman, C. Gruhl, C. Landauer, S. Tomforde, Self-improving system integration - on a definition and characteristics of the challenge, in: IEEE 4th International Workshops on Foundations and Applications of Self* Systems, FAS*W@SASO/ICCAC 2019, Umea, Sweden, June 16-20, 2019, 2019, pp. 1–3.

[5] K. Bellman, J. Botev, A. Diaconescu, L. Esterle, C. Gruhl, C. Landauer, P. R. Lewis, A. Stein, S. Tomforde, R. P. Würtz, Self-improving system integration-status and challenges after five years of sissy, in: 2018 IEEE 3rd International Workshops on Foundations and Applications of Self* Systems (FAS* W), IEEE, 2018, pp. 160–167.

[6] K. L. Bellman, S. Tomforde, R. P. Würtz, Interwoven Systems: Self-Improving Systems Integration, in: Eighth IEEE International Conference on Self-Adaptive and Self-Organizing Systems Workshops, SASOW 2014, London, UK, Sept. 8-12, 2014, 2014, pp. 123–127.

[7] S. Tomforde, J. Hähner, B. Sick, Interwoven Systems, Informatik-Spektrum 37 (5) (2014) 483–487, Aktuelles Schlagwort.

[8] J. Hähner, S. von Mammen, S. Tomforde, Autonomous Self-Integration in Interwoven Systems, in: International Conference on Architecture of Computing Systems – ARCS Poster Session, 2015, pp. 9 – 10.

[9] J. Kephart, D. Chess, The Vision of Autonomic Computing, IEEE Computer 36 (1) (2003) 41–50.

[10] S. Kounev, P. Lewis, K. L. Bellman, N. Bencomo, J. Camara, A. Diaconescu, L. Esterle, K. Geihs, H. Giese, S. Götz, et al., The notion of self-aware computing, in: Self-Aware Computing Systems, Springer, 2017, pp. 3–16.

[11] M. P. Papazoglou, W.-J. Van Den Heuvel, Service oriented architectures: approaches, technologies and research issues, The VLDB journal 16 (3) (2007) 389–415.

[12] Q. Zhang, L. Cheng, R. Boutaba, Cloud computing: state-of-the-art and research challenges, Journal of internet services and applications 1 (1) (2010) 7–18.

[13] K. L. Bellman, P. R. Nelson, C. Landauer, Active experimentation and computational reflection for design and testing of cyber-physical systems., in: CSDM (Posters), 2014, pp. 251–262.

[14] S. Tomforde, J. Hähner, H. Seebach, W. Reif, B. Sick, A. Wacker, I. Scholtes, Engineering and Mastering Interwoven Systems, in: ARCS 2014 - 27th International Conference on Architecture of Computing Systems, Workshop Proceedings, February 25-28, 2014, Luebeck, Germany, University of Luebeck, Institute of Computer Engineering, 2014, pp. 1–8.

[15] L. Atzori, A. Iera, G. Morabito, The internet of things: A survey, Computer networks 54 (15) (2010) 2787–2805.

[16] N. Komninos, E. Philippou, A. Pitsillides, Survey in smart grid and smart home security: Issues, challenges and countermeasures, IEEE Communications Surveys & Tutorials 16 (4) (2014) 1933–1954.

[17] H. Prothmann, S. Tomforde, J. Branke, J. Hähner, C. Müller-Schloer, H. Schmeck, Organic Traffic Control, in: Organic Computing – A Paradigm Shift for Complex Systems, Autonomic Systems, Birkhäuser Verlag, Basel, CH, 2011, pp. 431–446.

[18] M. Sommer, S. Tomforde, J. Hähner, Resilient Traffic Management with Organic Computing Techniques, in: Proceedings of the 1st International Systems Competition on Autonomic Features and Technologies for Road Traffic Modelling and Control Systems, held together with The 16th International IEEE Conference on Intelligent Transport Systems (IEEE-ITS13) at Den Hague, Netherlands, 2013, pp. 1–12, (Winner of the ARTS Competition at IEEE-ITS13).

[19] C. Gruhl, S. Tomforde, B. Sick, Aspects of measuring and evaluating the integration status of a (sub-) system at runtime, in: 2018 IEEE 3rd International Workshops on Foundations and Applications of Self* Systems (FAS* W), IEEE, 2018, pp. 198–203.

[20] R. Kazman, K. Schmid, C. B. Nielsen, J. Klein, Understanding Patterns for System of Systems Integration, in: Proc. 8th Intl. Conf. on System of Systems Engineering (SoSE), 2013, pp. 141–146.

[21] S. Tomforde, J. Kantert, B. Sick, Measuring self-organisation at runtime - A quantification method based on divergence measures, in: International Conference on Agents and Artificial Intelligence (ICAART), 2017, pp. 96–106.

[22] M. Wooldridge, An introduction to multiagent systems, John Wiley & Sons, 2009.

[23] P. Stone, M. Veloso, Multiagent systems: A survey from a machine learning perspective, Autonomous Robots 8 (3) (2000) 345–383.

[24] N. Howard, E. Cambria, Intention awareness: improving upon situation awareness in human-centric environments, Human-centric Computing and Information Sciences 3 (1) (2013) 1–17.

[25] S. Tomforde, B. Sick, C. Müller-Schloer, Spotlight on organic computing, Herausgegeben von Prof. Dr. Bernhard Sick, Universität Kassel (2017) 1.

[26] S. Wandelt, X. Sun, D. Feng, M. Zanin, S. Havlin, A comparative analysis of approaches to network-dismantling, Scientific reports 8 (1) (2018) 1–15.

[27] T. J. Holmes, Localization of industry and vertical disintegration, Review of Economics and Statistics 81 (2) (1999) 314–325.

[28] C. Landauer, K. L. Bellman, Integration systems and interaction spaces, in: Frontiers of Combining Systems, Springer, 1996, pp. 249–266.

[29] K. Bellman, The Conflict Behavior of the Lizard, Sceloporus Occidentalis: And Its Implication for the Organization of Motor Behavior, University of California, San Diego, 1979.

[30] K. L. Bellman, F. B. Krasne, Adaptive complexity of interactions between feeding and escape in crayfish, Science 221 (4612) (1983) 779–781.

[31] K. L. Bellman, C. Landauer, Towards an integration science: The influence of Richard Bellman on our research, Journal of Mathematical Analysis and Applications 249 (1) (2000) 3–31.

[32] K. V. Moffaert, T. Brys, A. Chandra, L. Esterle, P. R. Lewis, A. Nowé, A novel adaptive weight selection algorithm for multi-objective multi-agent reinforcement learning, in: Proceedings of the 2014 International Joint Conference on Neural Networks (IJCNN), IEEE Press, 2014, pp. 2306–2314.

[33] K. Bellman, What reasonable guarantees can we make for a SISSY system, in: 2018 IEEE 3rd International Workshops on Foundations and Applications of Self* Systems (FAS*W), 2018, pp. 231–233. doi:10.1109/FAS-W.2018.00052.

[34] K. Bellman, C. Landauer, How does self-integration change integration science?, Future Generation Computing Systems (in review) 1 (1) (2020) 1–22.

[35] C. Landauer, Degrees of Intimacy in SiSSy Systems "How to Join a Team", in: International Workshops on Foundations and Applications of Self* Systems (FAS* W), IEEE, 2019, pp. 10–17.

[36] B. A. Kitchenham, S. Charters, Guidelines for performing systematic literature reviews in software engineering, Technical Report EBSE-2007-01, Keele University (2007).

[37] K. Bellman, Strategies for Helping SISSY Systems Deal with Knowledge Gaps and Unknowns, in: 2019 IEEE 4th International Workshops on Foundations and Applications of Self* Systems (FAS*W), 2019, pp. 24–27. doi:10.1109/FAS-W.2019.00019.

[38] P. R. Lewis, A. Chandra, F. Faniyi, K. Glette, T. Chen, R. Bahsoon, J. Torresen, X. Yao, Architectural aspects of self-aware and self-expressive computing systems, IEEE Computer 48 (2015) 62–70.

[39] P. R. Lewis, Self-aware computing systems: From psychology to engineering, in: Proceedings of the 2017 Design, Automation & Test in Europe Conference & Exhibition (DATE), 2017, pp. 1044–1049.

[40] S. Tomforde, J. Hähner, S. von Mammen, C. Gruhl, B. Sick, K. Geihs, "Know Thyself" – Computational Self-Reflection in Intelligent Technical Systems, in: Eighth IEEE International Conference on Self-Adaptive and Self-Organizing Systems Workshops, SASOW 2014, London, United Kingdom, September 8-12, 2014, 2014, pp. 150–159. doi:10.1109/SASOW.2014.25.

[41] L. Esterle, J. N. A. Brown, I Think Therefore You Are: Models for Interaction in Collectives of Self-Aware Cyber-physical Systems, ACM Trans. Cyber-Phys. Syst. (in press) (2020) 1–24.

[42] C. Landauer, K. Bellman, Process planning and self-improvement in cyber-physical systems, in: Self-Adaptive and Self-Organizing Systems Workshops (SASOW), 2014 IEEE Eighth International Conference on, IEEE, 2014, pp. 144–149.

[43] K. L. Bellman, The SISSY Challenge: Expected and Unexpected Integration, in: 2017 IEEE 2nd International Workshops on Foundations and Applications of Self* Systems (FAS*W), 2017, pp. 146–147. doi:10.1109/FAS-W.2017.137.

[44] A. Stein, S. Tomforde, Transfer Learning is a Crucial Capability of Intelligent Systems Self-Integrating at Runtime, in: 2019 IEEE 4th International Workshops on Foundations and Applications of Self* Systems (FAS*W), 2019, pp. 32–35. doi:10.1109/FAS-W.2019.00021.

[45] H. Prothmann, J. Branke, H. Schmeck, S. Tomforde, F. Rochner, J. Hähner, C. Müller-Schloer, Organic Traffic Light Control for Urban Road Networks, International Journal of Autonomous and Adaptive Communications Systems 2 (3) (2009) 203 – 225. doi:http://dx.doi.org/10.1504/IJAACS.2009.026783.

[46] H. Prothmann, S. Tomforde, J. Lyda, J. Branke, J. Hähner, C. Müller-Schloer, H. Schmeck, Self-organised Routing for Road Networks , in: Proc. of the International Workshop on Self-Organising Systems (IWSOS'12), held in Delft, The Netherlands, March 15 - 16, 2012, no. 7166 in LNCS, Springer Verlag, 2012, pp. 48 – 59.

[47] K. L. Bellman, C. A. Landauer, Early work on the brain patch, a reflective service for system of systems integration, in: Autonomic Computing (ICAC), 2015 IEEE International Conference on, IEEE, 2015, pp. 249–254.

[48] A. Diaconescu, S. Frey, C. Müller-Schloer, J. Pitt, S. Tomforde, Goal-oriented Holonics for Complex System (Self-)Integration: Concepts and Case Studies, in: Proceedings of the 10th IEEE International Conference on Self-Adaptive and Self-Organising Systems, held September 12 – 16, 2016 in Augsburg, Germany, IEEE, 2016, pp. 100–109.

[49] C. M. Barnes, K. Bellman, J. Botev, A. Diaconescu, L. Esterle, C. Gruhl, C. Landauer, P. R. Lewis, P. R. Nelson, A. Stein, C. Stewart, S. Tomforde, CHARIOT - Towards a Continuous High-Level Adaptive Runtime Integration Testbed, in: 2019 IEEE 4th International Workshops on Foundations and Applications of Self* Systems (FAS*W), 2019, pp. 52–55. doi:10.1109/FAS-W.2019.00026.

[50] Y. Maurel, A. Diaconescu, P. Lalanda, Ceylon: A service-oriented framework for building autonomic managers, in: 2010 Seventh IEEE International Conference and Workshops on Engineering of Autonomic and Autonomous Systems, IEEE, 2010, pp. 3–11.

[51] E. A. Stoyanov, A. MacWilliams, M. A. Wischy, D. Roller, Distributed Software Maintenance Using an Autonomic System Management Approach based on the Viable System Model, in: International Conference on Autonomic and Autonomous Systems (ICAS'06), IEEE, 2006, pp. 58–58.

[52] S. Tomforde, I. Zgeras, J. Hähner, C. Müller-Schloer, Adaptive Control of Wireless Sensor Networks, in: Proceedings of the 7th International Conference on Autonomic and Trusted Computing (ATC'10), held in Xi'an, China (October 26-29, 2010), 2010, pp. 77 – 91.

[53] S. Wildermann, J. Teich, Self-integration for virtualization of embedded many-core systems, in: Self-Adaptive and Self-Organizing Systems Workshops (SASOW), 2014 IEEE Eighth International Conference on, IEEE, 2014, pp. 170–177.

[54] A. Di Stefano, G. Morana, D. Zito, C2@ home-a novel user-side cloud-of-clouds management architecture, in: 2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing, IEEE, 2015, pp. 1473–1480.

[55] P. Lalanda, A. Diaconescu, Integration of pervasive platforms with icasa, in: 2019 IEEE 4th International Workshops on Foundations and Applications of Self* Systems (FAS* W), IEEE, 2019, pp. 49–51.

[56] S. Frey, A. Diaconescu, D. Menga, I. Demeure, A generic holonic control architecture for heterogeneous multiscale and multiobjective smart microgrids, ACM Transactions on Autonomous and Adaptive Systems (TAAS) 10 (2) (2015) 1–21.

[57] V. Lesch, C. Krupitzer, S. Tomforde, Emerging self-integration through coordination of autonomous adaptive systems, in: 2019 IEEE 4th International Workshops on Foundations and Applications of Self* Systems (FAS* W), IEEE, 2019, pp. 6–9.

[58] S. Tomforde, S. Rudolph, K. Bellman, R. Würtz, An Organic Computing Perspective on Self-Improving System Interweaving at Runtime, in: Proceedings of the IEEE International Conference on Autonomic Computing, IEEE, 2016, pp. 276 – 284.

[59] Y. Cai, Y. Zhao, L. Lan, Design and implementation of a peripheral bus based on a new kind of reconfigurable system, in: 2011 IEEE Ninth International Conference on Dependable, Autonomic and Secure Computing, IEEE, 2011, pp. 286–291.

[60] E. Tuci, R. Groundefined, V. Trianni, F. Mondada, M. Bonani, M. Dorigo, Cooperation through self-assembly in multi-robot systems, ACM Transactions on Autonomous and Adaptptive Systems 1 (2) (2006) 115–150. doi:10.1145/1186778.1186779.

[61] S. Tomforde, M. Goller, To adapt or not to adapt: A quantification technique for measuring an expected degree of self-adaptation, Comput. 9 (1) (2020) 21.

[62] M. Handte, G. Schiele, V. Matjuntke, C. Becker, P. J. Marrón, 3pc: System support for adaptive peer-to-peer pervasive computing, ACM Transactions on Autonomous and Adaptive Systems (TAAS) 7 (1) (2012) 1–19.

[63] Q. Duan, Network service description and discovery for high-performance ubiquitous and pervasive grids, ACM Transactions on Autonomous and Adaptive Systems (TAAS) 6 (1) (2011) 1–17.

[64] H. Koshutanski, F. Massacci, Interactive access control for autonomic systems: from theory to implementation, ACM Transactions on Autonomous and Adaptive Systems (TAAS) 3 (3) (2008) 1–31.

[65] C. A. Lee, Managing a posteriori federations, in: Foundations and Applications of Self* Systems (FAS* W), 2017 IEEE 2nd International Workshops on, IEEE, 2017, pp. 130–131.

[66] T. Renner, A. Kliem, O. Kao, The device cloud-applying cloud computing concepts to the internet of things, in: 2014 IEEE 11th Intl Conf on Ubiquitous Intelligence and Computing and 2014 IEEE 11th Intl Conf on Autonomic and Trusted Computing and 2014 IEEE 14th Intl Conf on Scalable Computing and Communications and Its Associated Workshops, IEEE, 2014, pp. 396–401.

[67] J. Wu, M. Dong, K. Ota, J. Li, B. Pei, A fine-grained cross-domain access control mechanism for social internet of things, in: 2014 IEEE 11th Intl Conf on Ubiquitous Intelligence and Computing and 2014 IEEE 11th Intl Conf on Autonomic and Trusted Computing and 2014 IEEE 14th Intl Conf on Scalable Computing and Communications and Its Associated Workshops, IEEE, 2014, pp. 666–671.

[68] I. Kevin, K. Wang, A. Rajamohan, S. Dubey, S. A. Catapang, Z. Salcic, A wearable internet of things mote with bare metal 6lowpan protocol for pervasive healthcare, in: 2014 IEEE 11th Intl Conf on Ubiquitous Intelligence and Computing and 2014 IEEE 11th Intl Conf on Autonomic and Trusted Computing and 2014 IEEE 14th Intl Conf on Scalable Computing and Communications and Its Associated Workshops, IEEE, 2014, pp. 750–756.

[69] S. Tomforde, E. Cakar, J. Hähner, Dynamic Control of Network Protocols - A new vision for future self-organised networks, in: J. Filipe, J. A. Cetto, J.-L. Ferrier (Eds.), Proceedings of the 6th International Conference on Informatics in Control, Automation, and Robotics (ICINCO'09), held in Milan, Italy (2 - 5 July, 2009), INSTICC, Milan, 2009, pp. 285 – 290.

[70] K. Breitman, M. Perazolo, Using formal ontology representation and alignment strategies to enhance resource integration in multi vendor autonomic environments, in: Fourth IEEE International Workshop on Engineering of Autonomic and Autonomous Systems (EASe'07), IEEE, 2007, pp. 117–126.

[71] Y. Cai, W. Lu, X. Che, Y. Lu, Knowledge-enhanced multi-semantic fusion for concept similarity measurement in continuous vector space, in: 2015 IEEE 12th Intl Conf on Ubiquitous Intelligence and Computing and 2015 IEEE 12th Intl Conf on Autonomic and Trusted Computing and 2015 IEEE 15th Intl Conf on Scalable Computing and Communications and Its Associated Workshops (UIC-ATC-ScalCom), IEEE, 2015, pp. 415–423.

[72] D. Bonino, A. Bosca, F. Corno, An agent based autonomic semantic platform, in: International Conference on Autonomic Computing, 2004. Proceedings., IEEE, 2004, pp. 189–196.

[73] K. Bellman, J. Botev, H. Hildmann, P. R. Lewis, S. Marsh, J. Pitt, I. Scholtes, S. Tomforde, Socially-sensitive systems design, IEEE Technology & Society Magazine, Special Issue on Social Concepts in Self-Organising Systems 36 (3) (2017) 72–80.

[74] E. Pournaras, P. Pilgerstorfer, T. Asikis, Decentralized collective learning for self-managed sharing economies, ACM Transactions on Autonomous and Adaptive Systems (TAAS) 13 (2) (2018) 1–33.

[75] E. Pournaras, J. Nikolic, A. Omerzel, D. Helbing, Engineering democratization in internet of things data analytics, in: 2017 IEEE 31st International Conference on Advanced Information Networking and Applications (AINA), IEEE, 2017, pp. 994–1003.

[76] V. Smith, C.-K. Chiang, M. Sanjabi, A. S. Talwalkar, Federated multi-task learning, in: Advances in Neural Information Processing Systems, 2017, pp. 4424–4434.

[77] P. Nelson, Cars: A wrappings-based test bed for self* cyber-physical systems and their integration, in: 2019 IEEE 4th International Workshops on Foundations and Applications of Self* Systems (FAS* W), IEEE, 2019, pp. 44–48.

[78] V. Mandadapu, C. Stewart, Using game theory to manage self-aware unmanned aerial systems, in: 2018 IEEE 3rd International Workshops on Foundations and Applications of Self* Systems (FAS*W), Trento,

Italy, September 3-7, 2018, 2018, pp. 222–225.
[79] B. Becker, S. Kochanneck, H. Schmeck, Test beds for component integration in energy systems, in: 2019 IEEE 4th International Workshops on Foundations and Applications of Self* Systems (FAS* W), IEEE, 2019, pp. 40–43.
[80] U. Brinkschulte, M. Pacher, B. Betting, A simulator to validate the concept of artificial dna for self-building embedded systems, in: 2014 IEEE Eighth International Conference on Self-Adaptive and Self-Organizing Systems Workshops, IEEE, 2014, pp. 160–169.
[81] G. Chen, L. Luo, R. Gong, S. Gui, Dependability analysis for aadl models by pvs, in: 2009 Eighth IEEE International Conference on Dependable, Autonomic and Secure Computing, IEEE, 2009, pp. 19–24.
[82] Y. Celik, A. Pradeep, J. Y. Shi, Ankacom: A development and experiment for extreme scale computing, in: 2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing, IEEE, 2015, pp. 2010–2016.
[83] L. Zhu, Y. Wang, W. Zhang, S. Tan, Cpn based validation on pervasive cloud task migration, in: 2015 IEEE 12th Intl Conf on Ubiquitous Intelligence and Computing and 2015 IEEE 12th Intl Conf on Autonomic and Trusted Computing and 2015 IEEE 15th Intl Conf on Scalable Computing and Communications and Its Associated Workshops (UIC-ATC-ScalCom), IEEE, 2015, pp. 986–990.
[84] H. S. Kim, et al., Power system applications integration for the smart grid, in: 2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing, IEEE, 2015, pp. 965–969.
[85] I. Kevin, K. Wang, D. Somu, T. Parnerkar, Z. Salcic, Intelligent reconfigurable gateway for heterogeneous wireless sensor and actuator networks, in: 2015 IEEE 12th Intl Conf on Ubiquitous Intelligence and Computing and 2015 IEEE 12th Intl Conf on Autonomic and Trusted Computing and 2015 IEEE 15th Intl Conf on Scalable Computing and Communications and Its Associated Workshops (UIC-ATC-ScalCom), IEEE, 2015, pp. 262–269.
[86] J. L. Fernandez-Marquez, G. Di Marzo Serugendo, J. L. Arcos, Infrastructureless spatial storage algorithms, ACM Transactions on Autonomous and Adaptive Systems (TAAS) 6 (2) (2011) 1–26.
[87] P. Goncalves, J. L. Oliveira, R. L. Aguiar, A wbem based solution for a 4g network integrated management, in: Joint International Conference on Autonomic and Autonomous Systems and International Conference on Networking and Services-(icas-isns' 05), IEEE, 2005, pp. 29–29.
[88] J. Hähner, K. Streit, S. Tomforde, Cellular traffic offloading through network-assisted ad-hoc routing in cellular networks, in: IEEE Symposium on Computers and Communication, ISCC 2016, Messina, Italy, June 27-30, 2016, 2016, pp. 469–476.
[89] E. Pournaras, Overlay service computing-modular and reconfigurable collective adaptive systems, Scalable Computing: Practice and Experience 16 (3) (2015) 249–270.
[90] C. Landauer, Infrastructure for studying infrastructure, in: 2013 Workshop on Embedded Self-Organizing Systems, 2013, pp. 1–8.
[91] C. Landauer, K. L. Bellman, Integration by negotiated behavior restrictions, in: Foundations and Applications of Self* Systems (FAS* W), 2017 IEEE 2nd International Workshops on, IEEE, 2017, pp. 117–121.
[92] S. Alqahtani, X. He, R. Gamble, Adapting compliance of security requirements in multi-tenant applications, in: Foundations and Applications of Self* Systems (FAS* W), 2017 IEEE 2nd International Workshops on, IEEE, 2017, pp. 122–129.
[93] H. Heck, S. Rudolph, C. Gruhl, A. Wacker, J. Hähner, B. Sick, S. Tomforde, Towards autonomous self-tests at runtime, in: 2016 IEEE 1st International Workshops on Foundations and Applications of Self* Systems (FAS*W), Augsburg, Germany, September 12-16, 2016, 2016, pp. 98–99.
[94] H. Heck, B. Sick, S. Tomforde, Security issues in self-improving system integration - challenges and solution strategies, in: 2018 IEEE 3rd International Workshops on Foundations and Applications of Self* Systems (FAS*W), Trento, Italy, September 3-7, 2018, 2018, pp. 176–181.
[95] S. Edenhofer, S. Tomforde, J. Kantert, L. Klejnowski, Y. Bernard, J. Hähner, C. Müller-Schloer, Trust communities: An open, self-organised social infrastructure of autonomous agents, in: W. Reif, G. Anders, H. Seebach, J.-P. Steghöfer, E. André, J. Hähner, C. Müller-Schloer, T. Ungerer (Eds.), Trustworthy Open Self-Organising Systems, Springer, 2015, pp. 127–152.
[96] J. Kantert, S. Edenhofer, S. Tomforde, C. Müller-Schloer, Representation of trust and reputation in self-managed computing systems, in: 13th IEEE International Conference on Dependable, Autonomic and Secure Computing, DASC 2015, Liverpool, UK, October 26-28, 2015, 2015, pp. 1827–1834.
[97] S. Edenhofer, J. Kantert, L. Klejnowski, S. Tomforde, J. Hähner, C. Müller-Schloer, Advanced attacks to trusted communities in multi-agent systems, in: Self-Adaptive and Self-Organizing Systems Workshops (SASOW), 2014 IEEE Eighth International Conference on, IEEE, 2014, pp. 186–191.
[98] S. Edenhofer, C. Stifter, U. Jänen, J. Kantert, S. Tomforde, J. Hähner, C. Müller-Schloer, An Accusation-based Strategy to Handle Undesirable Behaviour in Multi-Agent Systems, in: Proceedings of the 12th IEEE International Conference on Autonomic Computing, held in Grenobles, France, July 07 – 10, 2015, IEEE, 2015, pp. 243 – 248.
[99] J. Kantert, S. Edenhofer, S. Tomforde, J. Hähner, C. Müller-Schloer, Robust Self-Monitoring in Trusted Desktop Grid for Self-Configuration at Runtime, in: Proceedings of SASO Workshops 2014, 2014, pp. 178 – 185, dOI 10.1109/SASOW.2014.28.
[100] J. Kantert, S. Tomforde, G. von Zengen, S. Weber, L. Wolf, C. Müller-Schloer, Improving Reliability and Reducing Overhead in Low-Power Sensor Networks using Trust and Foregiveness, in: Proceedings of the 13th IEEE International Conference on Autonomic Computing, held in Würzburg, Germany, 19 – 22 July, IEEE, 2016, pp. 325–333.
[101] S. Tomforde, H. Prothmann, J. Branke, J. Hähner, M. Mnif, C. Müller-Schloer, U. Richter, H. Schmeck, Observation and Control of Organic Systems, in: C. Müller-Schloer, H. Schmeck, T. Ungerer (Eds.), Organic Computing – A Paradigm Shift for Complex Systems, Autonomic Systems, Birkhäuser Verlag, 2011, pp. 325–338.
[102] S. Tomforde, S. Rudolph, K. L. Bellman, R. P. Würtz, An organic computing perspective on self-improving system interweaving at runtime, in: 2016 IEEE International Conference on Autonomic Computing, ICAC 2016, Wuerzburg, Germany, July 17-22, 2016, 2016, pp. 276–284.
[103] C. Krupitzer, F. M. Roth, M. Pfannemüller, C. Becker, Comparison of approaches for self-improvement in self-adaptive systems, in: 2016 IEEE International Conference on Autonomic Computing (ICAC), IEEE, 2016, pp. 308–314.
[104] D. Kaminsky, B. Miller, A. Salahshour, J. Whitmore, Policy-Based Automation in the Autonomic Data Center, in: 2008 International Conference on Autonomic Computing, IEEE, 2008, pp. 209–210.
[105] P. Idziak, S. Clarke, An analysis of decision-making techniques in dynamic, self-adaptive systems, in: Self-Adaptive and Self-Organizing Systems Workshops (SASOW), 2014 IEEE Eighth International Conference on, IEEE, 2014, pp. 137–143.
[106] S. Tomforde, A. Brameshuber, J. Hähner, C. Müller-Schloer, Restricted On-line Learning in Real-world Systems, in: Proc. of the IEEE Congress on Evolutionary Computation (CEC11), held 05 Jun - 08 Jun 2011 in New Orleans, USA, IEEE, 2011, pp. 1628 – 1635.
[107] C. Krupitzer, M. Pfannemüller, J. Kaddour, C. Becker, SATISFy: Towards a Self-Learning Analyzer for Time Series Forecasting in Self-Improving Systems, in: 2018 IEEE 3rd International Workshops on Foundations and Applications of Self* Systems (FAS*W), 2018, pp. 182–189. doi:10.1109/FAS-W.2018.00045.
[108] J. Schreiber, M. Bieshaar, A. Gensler, B. Sick, S. Deist, Coopetitive Soft Gating Ensemble, in: 2018 IEEE 3rd International Workshops on Foundations and Applications of Self* Systems (FAS*W), 2018, pp. 190–197. doi:10.1109/FAS-W.2018.00046.
[109] A. Stein, S. Tomforde, A. Diaconescu, J. Hähner, C. Müller-Schloer, A Concept for Proactive Knowledge Construction in Self-Learning Autonomous Systems, in: 2018 IEEE 3rd International Workshops on Foundations and Applications of Self* Systems (FAS*W), 2018, pp. 204–213. doi:10.1109/FAS-W.2018.00048.
[110] R. Grieben, R. P. Würtz, Cue Integration by Similarity Rank List Coding - Application to Invariant Object Recognition, in: 2017 IEEE 2nd International Workshops on Foundations and Applications of Self* Systems (FAS*W), 2017, pp. 132–137. doi:10.1109/FAS-W.2017.133.
[111] H. Wang, X. Chen, Q. Wu, Q. Yu, X. Hu, Z. Zheng, A. Bouguettaya, Integrating Reinforcement Learning with Multi-Agent Techniques for Adaptive Service Composition, ACM Trans. Auton. Adapt. Syst. 12 (2)

(May 2017). doi:10.1145/3058592.
URL https://doi.org/10.1145/3058592
[112] C. Müller-Schloer, H. Schmeck, T. Ungerer (Eds.), Organic Computing – A Paradigm Shift for Complex Systems, Autonomic Systems, Birkhäuser Verlag, Basel, CH, 2011.
[113] C. Landauer, K. Bellman, Designing cooperating self-improving systems, in: Autonomic Computing (ICAC), 2015 IEEE International Conference on, IEEE, 2015, pp. 273–278.
[114] A. Agarwal, J. Miller, J. Eastep, D. Wentziaff, H. Kasture, Self-aware computing, Tech. rep., MASSACHUSETTS INST OF TECH CAMBRIDGE (2009).
[115] R. S. Sutton, A. G. Barto, Reinforcement learning: An introduction, MIT press, 2018.
[116] S. J. Russell, P. Norvig, Artificial intelligence: a modern approach, Pearson Education Limited,, 2016.
[117] A. Stein, S. Tomforde, A. Diaconescu, J. Hähner, C. Müller-Schloer, A concept for proactive knowledge construction in self-learning autonomous systems, in: 2018 IEEE 3rd International Workshops on Foundations and Applications of Self* Systems (FAS*W), 2018, pp. 204–213. doi:10.1109/FAS-W.2018.00048.
[118] J. Frank, Reflecting on planning models: A challenge for self-modeling systems, in: Autonomic Computing (ICAC), 2015 IEEE International Conference on, IEEE, 2015, pp. 255–260.
[119] L. Esterle, J. N. Brown, Levels of Networked Self-Awareness, in: 2018 IEEE 3rd International Workshops on Foundations and Applications of Self* Systems (FAS*W), 2018, pp. 237–238. doi:10.1109/FAS-W.2018.00054.
[120] C. Gruhl, B. Sick, A. Wacker, S. Tomforde, J. Hähner, A building block for awareness in technical systems: Online novelty detection and reaction with an application in intrusion detection, in: IEEE 7th International Conference on Awareness Science and Technology, iCAST 2015, Qinhuangdao, China, September 22-24, 2015, 2015, pp. 194–200.
[121] C. Gruhl, B. Sick, S. Tomforde, Novelty detection in continuously changing environments, Future Generation Computer Systems 114 (2020) 138–154.
[122] S. Rudolph, S. Tomforde, B. Sick, J. Hähner, A Mutual Influence Detection Algorithm for Systems with Local Performance Measurement, in: Proceedings of the 9th IEEE International Conference on Self-adapting and Self-organising Systems (SASO15), held September 21st to September 25th in Boston, USA, 2015, pp. 144–150.
[123] C. M. Barnes, L. Esterle, J. N. A. Brown, "When you believe in things that you don't understand": the effect of cross-generational habits on self-improving system integration, in: 2019 IEEE 4th International Workshops on Foundations and Applications of Self* Systems (FAS*W), 2019, pp. 28–31. doi:10.1109/FAS-W.2019.00020.
[124] P. Andras, L. Esterle, M. Guckert, T. A. Han, P. R. Lewis, K. Milanovic, T. Payne, C. Perret, J. Pitt, S. T. Powers, N. Urquhart, S. Wells, Trusting intelligent machines: Deepening trust within socio-technical systems, IEEE Technology and Society Magazine 37 (4) (2018) 76–83. doi:10.1109/MTS.2018.2876107.
[125] A. Mishra, A. K. Misra, Formal aspects of specification and validation of dynamic adaptive system by analyzing execution traces, in: 2011 Eighth IEEE International Conference and Workshops on Engineering of Autonomic and Autonomous Systems, 2011, pp. 49–58. doi:10.1109/EASe.2011.14.
[126] C. A. Lee, Managing a posteriori federations, in: 2017 IEEE 2nd International Workshops on Foundations and Applications of Self* Systems (FAS*W), 2017, pp. 130–131. doi:10.1109/FAS-W.2017.132.
[127] A. Garcia Dominguez, N. Bencomo, J. M. Parra Ullauri, L. H. Garcia Paucar, Towards history-aware self-adaptation with explanation capabilities, in: 2019 IEEE 4th International Workshops on Foundations and Applications of Self* Systems (FAS*W), 2019, pp. 18–23. doi:10.1109/FAS-W.2019.00018.
[128] S. Tomforde, From "normal" to "abnormal": A concept for determining expected self-adaptation behaviour, in: 2019 IEEE 4th International Workshops on Foundations and Applications of Self* Systems (FAS*W), Umea, Sweden, June 16-20, 2019, 2019, pp. 1–6.
[129] J. A. Thomson, On growth and form, Nature 100 (2498) (1917) 21–22.
[130] A. Diaconescu, P. Mata, K. Bellman, Self-integrating organic control systems: from crayfish to smart homes, in: ARCS Workshop 2018; 31th International Conference on Architecture of Computing Systems, VDE, 2018, pp. 1–8.
[131] S. Beer, The viable system model: Its provenance, development, methodology and pathology, Journal of the operational research society 35 (1) (1984) 7–25.
[132] C. Landauer, Mitigating the inevitable failure of knowledge representation, in: Proceedings the 2nd International Workshop on Models@run.time for Self-aware Computing Systems, 2017, pp. 1–8.
[133] K. Bellman, Self-reflection and a version of structured "playing" may be critical for the verification and validation of complex system of systems, CSDM 2013 (2013).

## Appendix A. Considered venues

- ICAC: IEEE/ACM International Conference on Autonomic Computing
- DASC: IEEE International Conference on Dependable, Autonomic, and Secure Computing
- SASO: IEEE International Conference on Self-Adaptive and Self-Organising Systems
- ATC: IEEE International Conference on Autonomic and Trusted Computing
- ICAS: IEEE International Conference on Autonomic and Autonomous Systems
- EASE: IEEE International Workshop / IEEE International Conference on Engineering of Autonomic and Autonomous Systems
- SISSY: International Workshop on Self-improving System Integration (IEEE/ACM)
- ACW: Autonomic Computing Workshop
- TAAS: ACM Transactions on Autonomous and Adaptive Systems
- TCPS: ACM Transactions on Cyber-Physical Systems