Lifelike Computing Systems

Anthony Stein[1], Sven Tomforde[2], Jean Botev[3], Peter R. Lewis[4]

[1]University of Hohenheim, Germany
[2]University of Kiel, Germany
[3]University of Luxembourg, Luxembourg
[4]Ontario Tech University, Canada
anthony.stein@uni-hohenheim.de

Abstract

Technology today is becoming increasingly autonomous, already comprising many features typically associated with human, animal, and even plant behaviour: for example decision-making, problem-solving, prediction, adaptation, and self-regulation. Drawing on and embodying techniques from artificial intelligence (AI) and bio-inspired computing, many so-called intelligent, smart, and self-adaptive systems are designed with the explicit intention of replicating such behaviours. At the same time, the study of artificial life has explored many properties of living systems, both as they are found in nature and as they can be built or conceived of by humans. This has exposed a large variety of mechanisms that produce what we call *qualities* typically associated with life. Examples include self-organisation, homeostasis, self-replication, evolution, learning, self-awareness, and many others. As the technological world becomes ever more complex, interconnected, fast, and invisible, there may be substantial value in these qualities also being present in the systems we build. In this paper, we take the first steps to cast light on what constitute *Lifelike Computing Systems*, why such systems are worth striving for in the first place, and what we need in order to pave the road for them. We bring this notion in line with existing research initiatives sharing the explicit systems engineering focus. Important research aspects are derived that serve as the basis for a working agenda towards lifelike computing systems.

I. Why Lifelike Computing Systems?

Technological systems have always been part of modern human life. Historically, while initially taking the form of passive tools, such as axes and spoons, the industrial revolution saw the advent of powered, mechanised technology, operating "under their own steam" without direct human control over every action. By integrating more complex information processing machinery, automation evolved into autonomy, as decision-making and self-regulation became features of modern technology. Now, so-called "intelligent", "smart", and "self-adaptive systems" find, maintain and recover suitable behaviours for changing contexts. These are designed with the explicit intention of carrying out 'rational' behaviours, leading to technology doing the sorts of things often attributed to natural intelligence (Boden, 2016). At the same time, the study of *Artificial Life* (ALife) (Banzhaf and McMullin, 2012) has explored the properties of living systems, both as they are found in nature and as they can be built by humans, thereby pursuing a "life-as-it-might-be" modeling philosophy (Langton, 1992). This has exposed a large variety of mechanisms that produce qualities typically associated with life. Examples include self-organisation, homeostasis, self-replication, evolution, learning, self-awareness, and many others besides. As the technological world becomes ever more complex and interconnected (Weiser, 1999), fast (Tennenhouse, 2000) and invisible (Norman, 1998), there may be substantial value in these qualities also being present in the systems we build.

The "Lifelike Computing Systems" initiative aims to learn from the study of life and living systems to develop new, practical, computing systems that possess 'lifelike' properties; a further goal is to identify when such complex features are of particular value. The initiative's focus lies primarily on engineered technological systems broadly within the domain of computing. However, this term is not intended to separate itself from or replace previous initiatives; in a large number of cases, there are already technologies and research efforts that strongly lean towards what we deem lifelike computing systems in specific aspects. Rather, our focus is on a holistic view of these properties, their development as a suite, and their fruitful combination. In doing so, the initiative aims to emphasise the drawing together of existing approaches, technologies, and systems. On that basis, we aim to shine a light on existing research towards this vision, and to also determine what future research is necessary (i.e., open questions, knowledge gaps, and limitations) in pursuing a holistic view of lifelike computing systems.

The basic motivation to develop more 'lifelike' systems also has its roots in previous initiatives: cybernetics, for example, considered the increasing complexity of system behaviour, adaptive control and the resulting interrelationships and challenges (Wiener, 2019). Later, the notion of complex adaptive systems (Holland, 1992) introduced another research perspective to capture and understand the underlying principles of naturally existent systems such as e.g., economies or ecology. Observing the advances and trends in computing technology, these insights were taken up again, for example, at the beginning of this millennium in the context of the *Proactive* (Tennenhouse, 2000), *Autonomic* (Kephart and Chess, 2003), and *Organic Computing* (Müller-Schloer and Tomforde, 2017) initiatives; all sharing the common understanding that the controllability of future systems with at that time current techniques was no longer achievable. Since then, we can observe a rapid technological development, which on the one hand has resulted in new types of machine intelligence, adaptive control strategies, and accordingly more autonomous behaviour of technical systems. On the other hand, this also created new possibilities, in that new types of computing and communication technologies have greatly increased performance.

The difference compared to 20 years ago is now that (besides the technological capabilities) the world is changing towards a digitalised, data-driven, technology-mediated environment that intertwines everything into an integrated 'superorganism', as e.g., visible in the context of the Internet-of-Things (Ashton et al., 2009), Internet-of-Everything (Snyder and Byrd, 2017), or social cyber-physical systems (Sha et al., 2008), fields that aim at deeply integrating humans and technical systems. These systems are socio-technical, and open-ended. As a result, the need grows for systems that automatically find solutions for needs, challenges, and dynamics that were either simply not there, or at least not within our sphere of awareness, during development (Tomforde and Müller-Schloer, 2014).

Building on a long and highly successful tradition in biologically-inspired computing (cf. Bongard (2009)), the 'lifelike' vision not only seeks *inspiration* in the living world, but also seeks to *replicate its qualities explicitly in technological systems*. The envisaged agenda also goes beyond pure ALife research, often rightly exploratory in nature, since it focuses explicitly on building purposeful and reliable technological systems for people, indeed based on both AI as well as ALife principles. In this context, the construction of lifelike computing systems will build on several decades of previous bio-inspired initiatives. However, the vision of explicit replication of lifelike qualities marks a departure: indeed, we cannot claim that all bio-inspired systems remain lifelike, nor is this in-general even always a desirable outcome for those designing bio-inspired systems. For example, many evolutionary algorithms, such as a simple (1+1)-EA (cf. De Jong (2016)) are clearly biologically inspired in origin. However, they contain very few of the qualities that we would commonly ascribe to something lifelike. Other examples can be found in, for example, neural-inspired machine learning systems.

In particular, we would like to have certain qualities available in the systems themselves – already implemented by design, when talking about lifelike computing systems. In this paper, we provide an initial discussion what lifelike computing systems might be, by focusing on an initial selection of certain qualities possessed by natural living systems (Section II). This initial list will include rather intuitive aspects such as open-ended evolution or different degrees of intelligence, involving simple reactive behavior but also more complex capabilities such as introspection. But also beyond these intuitive qualities, further aspects such as emergence, resilience and social behaviour will be discussed, which together are hypothesized to allow for more adaptive and reliable system behaviour, ensuring socially sensitive and compliant actions, as well as still fostering transparency even when evolving and self-integrating into higher order system constitutions. However, this initial list will necessarily be both incomplete and overly prescriptive since to our knowledge there is yet no all-agreed consent on what exact ingredients are defining life itself. Given the basic motivation outlined before, the fundamental properties remain indeed almost the same as they have been and still are for e.g., the related Autonomic and Organic Computing initiatives: We are on a quest for highly robust, flexible, trustworthy, reliable, and efficient solutions for technical systems designed to act and survive under the challenging conditions the world bears to them when they are embedded within it. Based on our first discussion of, at least from the authors' perspective essential, lifelike qualities, we will look at the current state – what technology is already available from other research initiatives concentrating on the systems engineering aspect and to which of the delineated qualities they provide valuable contributions (Section III). Based on this brief assessment, we will sketch important aspects of research upon which an initial research agenda can be build with the goal of approaching 'truly' lifelike computing systems (Section IV). Finally, the paper closes with a conclusion and outlook (Section V).

II. What Are Lifelike Computing Systems?

The obvious question that arises here is, "Do we really need yet another computing paradigm?". We answer this question with caution and want to clarify that the quest for Lifelike Computing Systems is not to be considered an orthogonal way of engineering complex computing systems, but an advancing one. That is, we build on existing research initiatives such as Organic, Autonomic, and Self-Aware Computing (cf. Sect. III), attempting to integrate their unique and shared perspectives, the already achieved insights, but also still open challenges into a unifying framework. In simpler terms, we look at the advances and ways of thinking that these research fields have given us, and ask: *what's next?*

In this positioning work, we initiate this research endeavour by first identifying five striking properties or capabilities of *living* systems that we deem especially valuable to be brought into technical systems. However, it should be clearly noted that our first attempt is necessarily incomplete and that we by no means strive to propose a definition of life itself here. Therefore, here we touch upon this initial set of ‘qualities of life’, and hope to stimulate the reader in provoking deeper thoughts and creating broader interest on this notion.

1. Open-ended Evolution. The first quality of living organisms and natural systems is their continuing evolution (Darwin, 1859). Evolution is considered a mechanism to change and thus constitutes a building block for continuing system adaptation. Therefore, we not only think of typical evolutionary computation techniques (De Jong, 2016), but explicitly emphasise open-ended evolution Taylor et al. (2016). In real-world settings, the objectives provided to computing systems by humans are usually neither static nor fixed. Objectives vary, i.e., they are subject to gradual or abrupt changes, seasonal impacts, are often multi-modal, and can be highly contradictory in different sub-goals. In open systems, objectives should also be considered to be multi-scale: present over multiple system levels in a hierarchy of abstraction, with each level (for example) having an impact on the lower one. This clearly raises complexity further still (Diaconescu et al., 2016). Accordingly, and as noted by Stanley and Lehman (2015), relying exclusively on fitness functions, i.e., a mathematical model used to internalise the system objectives into our engineered ‘intelligent’ systems, appears to be unnecessarily limiting for the production of the myriad of creative ways in which living systems evolve to behave. Much more complex mechanisms are needed to steer these systems toward continual adaptive, flexible, and creative behavior, such as basing fitness also on stimuli beyond numerical utility values. Detecting novelty and positively reinforcing novel behaviors might be one possible path to reach this goal (cf. e.g., Lehman and Stanley (2008)), though more broadly, fitness may often arise endogenously within the environment and the people and others with which the system interacts.

2. Intelligence. One of the most distinguishing qualities between living and non-living systems is *intelligence*. As is already the case for the notion of life, there is again no universally-accepted consensus on what exactly defines *natural* intelligence. However, capabilities that are typically mentioned include, among others, *rational thinking*, *problem-solving*, *reasoning* on certain (but more fascinatingly uncertain) knowledge, the acquisition of competence, skill, and knowledge by different forms of *learning*, the use of *intuition* and other mental heuristics, *self-awareness* and various forms of reflection, as well as *emotional thinking*, and *creativity*. For *artificial* intelligence, however, numerous attempts to delineate this scientific discipline can be quickly spotted, especially in these days where AI is once again perceived as probably the most promising technology with outstanding disruptive potential. Definitions here range from similarly listing competencies (e.g., Bellman (1978)) to more inclusive statements concerning machines that *do things like minds do* (e.g., Boden (2016)). Following Russell and Norvig (2020), one can approach such a definition from four directions, where an AI is thought of as (1) Thinking Humanly, (2) Acting Humanly, (3) Thinking rationally, and (4) Acting rationally. For the overarching purpose of engineering and understanding complex computing systems, we often follow the fourth angle and define intelligence in a technical sense as being brought into systems through an agent (software) that acts rationally, which is further defined in the sense of a way that maximises a numeric measure of utility shaped by external performance standards. According to this perspective, an ‘intelligent system’ is essentially a system that can maintain its utility under challenging conditions (e.g., time-varying environments, emergent situations, or disturbances) by *autonomously* adapting its behaviour to changing circumstances. In line with this notion, *learning* is often one of the most obvious and intuitive functions of an intelligent system. Next to the explicit goal of improving a system’s performance in well-defined (and therefore, typically narrowly-defined) tasks, learning also enables a system to extend its knowledge and, thus, to expand its concept of known environments (cf. concept drift and generalisation). This is possible by taking advantage of already gained experiences and direct interaction with the environment to experience new situations in an explorative fashion; the latter however usually comes at the cost of trial-and-error situations, which in the online case, can add additional risk or cost. But beyond this specific perspcetive on ‘learning and acting’ in response to perceived stimuli and feedback in order to maintain a user-defined utility level, when considering lifelike computing systems, we subsume further capabilities of naturally intelligent organisms under this quality. For example, *creativity* is a prerequisite to solving many intricate problems, as well as for developing framings with which to express newly discovered problems. Bringing creativity into our systems by, e.g., computer simulations or models that imitate mental processes detached from the purely reactive timescale, we expect to increase the problem-solving capabilities of systems tremendously. Another distinguishing property of intelligent beings such as humans is their ability for *introspection*. Humans are (to varying degrees at various times) self-aware. We can reflect on our behaviour (even if it often happens retrospectively), e.g., to assess our own knowledge and strengths, and we often have an inner clue that serves as a measure for their quality of work (e.g., perfectionism). And this this section, we have only touched on aspects commonly associated with human intelligence; there are many other quite different forms of intelligence found in the natural world, with quite radically different emphases, to learn from and emulate.

3. Emergence. Beyond thinking of intelligence on an individual scale, for lifelike computing systems, we want to emphasize the importance of *collective intelligence*. In natural living systems, often not every organism can be considered particularly intelligent on an individual basis. However, nature also reveals remarkable capabilities when ‘simple’ (non-intelligent) species act collectively, either in smaller groups or in larger swarms. Often the principle of self-organization governs in such decentralized (swarm) systems, what results in *emergent* phenomena sometimes understood as intelligence of a 'superorganism'. Prominent examples from nature are ant colonies or bee hives which exhibit intriguing capabilities by far not possible to be anticipated by looking at only one single member of the superordinate collective. Nevertheless, emergent effects as a product of local interaction between a large number of self-motivated entities (self-organisation) and resulting feedback loops do not always lead to what we might see as desired system behavior. If we strive for evolving computing systems comprised of intelligent system components, we must not only bear in mind the potentially undesired side-effects, but also develop ways to quantify and measure them (cf. e.g., Mnif and Müller-Schloer (2011)). In lifelike computing systems, emergence thus needs to be detected and "controlled" (or perhaps steered), in the sense that the system itself is enabled to enforce positive (e.g., robustness, adaptivity) and simultaneously dampen adverse (e.g., stalling competition for scarce resources) emergent effects.

4. Resilience. Another striking property of living systems is their *resilience* despite disturbing events, such as volatile climate conditions, natural disasters, and also gradually changing conditions, where they can still successfully recover to maintain acceptable performance, or perhaps to simply survive. Yet, it is clear that this occurs at multiple levels, often depending on the severity: the level of the individual organism, at the population level, and further, at the level of whole ecosystems. Indeed, resilience can, and needs to be viewed on different hierarchical and temporal scales.

Quickly recovering from unforeseen or unanticipated disturbances to maintain a viable system operation might be considered short-term fault-tolerance or, more generally *technical robustness*. However, the capability to consider measures on multiple system scales, e.g., as a response to changing goals that have a strong and long-term impact on overall system properties, or acknowleding that reconfiguration and diverse solution perspectives on a problem exist, might be better referred to as *ecosystem flexibility*. Examples might include the necessity to form entirely new constellations of system parts or to self-integrate with other specialised systems (Bellman et al., 2021). Resilience such as this can involve rapid or indeed more longer-term exploratory learning, but equally can rely on adaptive feedback mechanisms either simple or complex.

The essence of this fourth quality is that, for lifelike computing systems, mechanisms will be needed that minimize the brittleness of technical systems in order to tackle the inherent complexity of the world; and that this cannot be done only by them being insulated from it. Indeed, socio-technical settings from infrastructure services, healthcare, and agriculture, to manufacturing, supply chains, and many other besides, generate dynamic, often unforeseen, and compound environmental, legal, or societal conditions. Lifelike computing systems present an opportunity to move beyond the Hobson's choice of 'carry on regardless even though the scope has changed' or 'redesign and redeploy'.

This calls at least for appropriate degrees of redundancy (cf. 3. Emergence), but also mechanisms for system introspection (cf. 2. Intelligence), and suitable decentralized system architectures that can quickly compensate. 'Failures' should instead be framed as *failures in assumptions*, and be adapted to, reconfiguring into a new space of possibilities (cf. 1. Open-Ended Evolution), and drawing on social awareness of supporting counterparts that can help overcome the disturbance (cf. 5. Social Awareness, below).

5. Social Awareness. Finally, a fundamental quality of human and animal societies is their ability to establish social behaviour, to empathise and reason socially about others, and to establish and follow social norms. Such societies face challenges that increasingly occur in technical systems: they have to interact with unknown populations, sometimes without understanding their language and cultural background, and have to find and maintain an inner balance based on, for example, fairness and equality. Transferred to our human society, this ultimately involves the establishment and continuous consideration of ethical, value-based actions. Technically, this implies – especially when humans are seen as a fundamental part of the system and no longer just as "users" – that systems need to be socially sensitive (Lewis, 2017b) and to have a sense of the ethical implications of their actions (Bellman et al., 2017). In view of lifelike computing systems, mechanisms are needed to: (1) Interact with unknown participants in open systems (e.g., technical trust (Reif et al., 2016)), (2) to recognise interdependencies and mutual influences, especially of a hidden, indirect nature, e.g., Rudolph et al. (2019), (3) to develop and comply to norms that govern autonomous individual behaviour in accordance with overarching common goals (e.g., Kantert et al. (2016)), and finally, (5) to explain their inner reasoning to involved human stakeholders but also other computational counterparts, requiring both an inter-operable abstracted language and context-sensitive human-machine interfaces.

We reiterate that this is not intended as an exhaustive list of characteristics. Rather, we hope it serves both to provoke further thought on how these concepts might show up in the behaviour of future socio-technical systems and to illustrate how the compound nature of features often studied in separate subdisciplines is important for a holistic view of lifelike computing systems. Given the ambitious goals of this initiative for these pivotal characteristics, in the next section we focus on assessing where we are now.

III. Where Are We Now?

The vision of establishing lifelike qualities in future technical systems has its roots in and draws upon previous initiatives, which we briefly review in the following in order to provide an overview of the evolution of the field. Due to space restrictions, we however concentrate on those with an explicit engineering perspective. Accordingly, research fields such as artificial life, theoretical biology, self-organization, etc., are necessarily neglected here, even though they provide the relevant insights for capturing and analysing the complexity of living systems upon which such engineering efforts sit. In the future, we believe the field would be well-served by focussed reviews concerning how insights from these disciplines can tangibly impact the design and operation of lifelike computing systems.

Multi-agent Systems (MAS). MAS consist of several interacting, intelligent entities – so-called *agents* (Wooldridge, 2009). In this context, the term 'agent' refers to a software unit that autonomously processes tasks on behalf of a user or administrator – but it can also refer to robots, humans, or even heterogeneous constellations of them (Wooldridge, 2009). Agents are used to model or solve problems that cannot be handled in a standard monolithic way due to high degrees of parallelism and/or complexity. Usually, a MAS forms a sort of heuristic approach for an otherwise intractable or too complex to model problem (Jennings, 2000). In literature, the concept has been successfully applied to several well-known tasks, e.g., modelling social structures (Sun and Naveh, 2004), on-line trading (Rogers et al., 2007) or devising agent-based models for agricultural systems (Berger, 2001). In an MAS, agents take their decisions based on predefined goals and are able to interact with each other. For our envisioned lifelike computing systems, especially the existing technology for interaction schemes and protocols provide valuable starting points for the delineated qualities of 'emergence' and 'social awareness'.

Proactive Computing (PAC). Tennenhouse (2000) stated that – as he called it – "human-in-the-loop computing" has its limits. Embedded computing devices became increasingly popular, which resulted in a dramatic increase in the number of utilised devices running information and communication technology. The sheer number demanded a paradigm shift in administration to further guarantee controllability, not unlike the challenges we now face more than twenty years later. Although PAC mainly presented a vision and had a strong focus on hardware challenges, the motivation still holds for lifelike computing systems and we can draw inspiration upon PACs vision for nearly all qualities we mentioned above.

Autonomic Computing (AC). Motivated by the increasing complexity in large data centres, Kephart and Chess (2003) argued that computing systems need an automated backbone structure similar to the autonomic nervous system of humans. The idea is that this autonomic structure relieves the designer from specifying all possibly occurring situations and configurations within the design process. Instead, the system itself takes over the responsibility to find appropriate reactions to perceived changes in environmental conditions. It also relieves the administrator of configuration and maintenance tasks, especially in finding optimised settings for resources. Although this is still limited to dedicated control problems, use cases, and controlled decision freedom of the autonomic systems, the basis for certain aspects of lifelike behaviour by means of feedback and self-adaptation is already laid. Concerning the five qualities of life, AC's achievements have clearly contributed to equip technical systems with more 'intelligence' and 'resilience'.

Organic Computing (OC). Based on the motivation of mastering the ever-growing complexity in technical systems, the OC initiative took inspiration from and bring basic concepts from natural and biological into technical systems (Müller-Schloer and Tomforde, 2017; Tomforde et al., 2017) with the aim of transferring formalisations of self-x properties to engineered technological systems. These supported the achievement of higher-order system characteristics, such as robustness, flexibility, and viability under challenging real-world conditions. As a result, traditional design-time decisions are shifted to run time, and into the responsibilities of systems themselves. This includes adaptation decisions based on machine learning technology, detecting changes in the underlying processes to be controlled, or maintaining relationships among distributed systems. However, the concrete control problem is still narrowly defined and dealt with within, to the best possible degree predetermined boundaries. Already from the motivation, but also from the obtained achievements regarding self-adaptive and self-organising system technology, OC can be considered a substantial basis for further developing the qualities of 'resilience', 'emergence' and also 'intelligence'.

Self-Aware Computing (SeAC). The idea of self-awareness in computing arose over many years in a variety of areas of computer science, artificial intelligence, and engineering. Over the last ten years, however, drawing on self-awareness theories in psychology, a fundamental understanding of what self-awareness concepts can mean for the design and operation of computing systems has been developed (e.g., Lewis et al. (2011, 2016); Kounev et al. (2017)). This led to a number of contributions in terms of definitions, architectures, algorithms and case studies, targeted at explicitly designing *computational self-awareness* into technical systems. Computational self-awareness capabilities typically reference internal state, history, social or physical environment, goals, and even a system's own way of representing and reasoning about these things.

A number of architectures for self-aware systems exist (e.g. Lewis et al. (2015); Kounev et al. (2017)), and these typically extend the (self-)knowledge representational and acquisition capabilities of intelligent systems (e.g., building on the MAPE-K architecture or any of the other knowledge-based or learning-based agents (Russell and Norvig, 2020)) with fine-grained details concerning the system itself.

Consequently, the notion of computational self-awareness is clearly also key to lifelike computing systems when it comes to establishing 'intelligence' and the capability of introspection. Lewis (2017a) provides a summary.

Interwoven Systems (IwS) In contrast to traditional system design, the rising utilisation of communication technology and the increasing interconnectedness of systems resulted in blurring system boundaries. Instead of following the 'separation of concerns' idea by building modules that are combined during the development process, the IwS initiative (Bellman et al., 2014) focuses on changing system goals that define the setup and configuration of the composition and structures at runtime. A contained component system can in turn consist of autonomous systems itself, modules can aggregate different groups of entities towards a (sub-)system, and even the goal that is followed might change continually (Bellman et al., 2021). Hence, IwS on the one hand already considers aspects of the 'open-ended evolution' quality in respect to evolving system compositions, but also involves 'social awareness', both deemed key also in lifelike computing systems.

Summary

As a result of the research done in the context of these fields, we find a wide variety of techniques, methods, and architectural approaches that can serve as the basis for lifelike computing systems. Several concepts for designing individual component systems have been proposed, such as the *observer/controller* pattern (Tomforde et al., 2011) from OC, the *monitor-analyse-plan-execute* cycle (Kephart and Chess, 2003) from AC, several from self-aware computing (Lewis, 2017a), as well as those from AI and robotics (Russell and Norvig, 2020). Essentially, however, we note that these are simply agent patterns that each includes or emphasises a different set of features or processes, from our list of characteristics, over the other. One key question will be how to find unifying metamodels (at least in the conceptual space, if not for actual implementation) that support the holistic consideration of lifelike systems in general.

IV. Approaching Lifelike Computing Systems

Based on the notion of lifelike qualities from Section II and the brief overview of available technology from Section III, we now discuss first research avenues for approaching lifelike computing systems. It turns out that available technology from the various fields that have emerged over the past few decades can already be combined very well and integrated with each other. As we have sketched in this paper, several important aspects of potential lifelike computing systems are partly missing in some of the existing related research directions. On the other hand, several of these aspects are treated deeply but in an isolated fashion in others. In an attempt to bridge this gap, in the following, we delineate four important aspects upon which we propose a research agenda towards lifelike computing systems be based:

i) Framework: An integrated approach requires a common understanding of existing activities and techniques. Therefore, much as was done in successfully unifying the field of evolutionary computation (De Jong, 2016), we propose to revisit existing research and technology, integrating it into a unifying framework, resulting in a 'toolbox' in the sense of a methodological repertoire and an architectural metamodel for lifelike computing systems. Thus, each of these specific initiatives can then be clearly seen as addressing a part, a perspective perhaps, on the whole.

ii) Testbeds: It is worth noting that initiatives such as OC and AC, despite 20 years of successful research history, cannot provide uniform benchmarks or testbeds – as is the case, for example, in machine learning. This is mainly due to the challenge that, as by definition an open-ended problem space, a use case that also draws on the reader's intuition as to the possibilities is always necessary to demonstrate the technology. This makes transferability of approaches fundamentally difficult, but neither is this complexity necessarily something to be wished away. For example, it could be argued that in constructing common benchmarking sets for machine learning, the scope of what machine learning systems are expected to do is by definition artificially narrowed. Nevertheless, we believe that for lifelike computing systems, it is important to consider how to establish generic testbeds, that provide for reproducability and comparability, while not sacrificing generality and open-endedness.

iii) System quantification: Typically, the success of technical systems is considered in relation to a specific utility function. For lifelike computing systems, this will only make part of the evaluation. Thus, metrics that have been derived in part for OC/AC systems, e.g., for measuring adaptivity, self-organisation, or robustness, need to be extended in order to quantify the complex inner states of the systems, as well as how these relate to broader interactions and ecosystems. The hierarchical nature of goals and resilience, as discussed in Section II will be an essential consideration.

iv) Computational approaches: In order to realize lifelike computing systems, novel computational processes and their implementations will be needed to underpin and instantiate the above. For instance, beyond conventional approaches like evolutionary computation, mainly targeted at optimization problems and automatic programming, novel algorithms, computational models, architectures, and reflective processes are required that rather aim at establishing the more fundamental requirement for adaptive behaviour in open-ended settings on different scales of system design. As another example, the basic capability of the systems must go beyond the mere data-driven building of knowledge through experience. In particular, we note that computational introspection and reflection, creativity, empathy, and social intelligence are still in their infancy. Introspection, for example, involves the systematic assessment of a system's own knowledge but also an intrinsic motivation for continual self-improvement (e.g., artificial curiosity). Further, the multi-level nature of resilience is a largely unexplored area in computing. This requires a micro-macro perspective, which implies cooperation with other entities on microscopic levels, but also monitoring of emergent macroscopic behavior. Finally, the focus of interest should also be that lifelike computing systems have broad compatibility: with legacy systems; with heterogeneous approaches to problem solving; with upcoming solutions whose existence is yet unknown but which lifelike computing systems ought to be 'prepared' for from first principles; and finally, compatible with the 'socio-' side of the socio-technical systems of which they are part – with human society.

As with our list of characteristics, we do not claim this short sketch of areas of required research focus to be exhaustive. Rather, we see this as a starting point in our pursuit of a unifying research agenda.

V. Conclusion and Outlook

This work presents an initial step towards introducing the vision of lifelike computing systems: technological systems, of benefit to people, that are not only inspired by the living world, but are explicitly intended to replicate its qualities.

The capabilities of traditional technical systems no longer suffice to master the increasing complexity within the larger contexts they operate. Therefore, we postulated five *qualities of life* that we deem essential for future computing systems, in order to meet these heightened requirements. This set of qualities is necessarily incomplete, and the agenda of lifelike computing systems is intended not as any fundamentally new paradigm, but instead to ask the question: *'what's next?'* for adaptive, bio-inspired technical systems – and further, to ask if these initiatives can be placed within a broader unified perspective. In this paper, we sketched what such a perspective might look like, at least some of what it ought to draw on, and how we might answer these questions.

References

Ashton, K. et al. (2009). That 'internet of things' thing. *RFID journal*, 22(7):97–114.

Banzhaf, W. and McMullin, B. (2012). Artificial life. In Rozenberg, G., Bäck, T., and Kok, J. N., editors, *Handbook of Natural Computing*, pages 1805–1834. Springer.

Bellman, K., Botev, J., Hildmann, H., Lewis, P. R., Marsh, S., Pitt, J., Scholtes, I., and Tomforde, S. (2017). Socially-sensitive systems design: Exploring social potential. *IEEE Technology and Society Magazine*, 36(3):72–80.

Bellman, K. L., Botev, J., Diaconescu, A., Esterle, L., Gruhl, C., Landauer, C., Lewis, P. R., Nelson, P. R., Pournaras, E., Stein, A., and Tomforde, S. (2021). Self-improving system integration: Mastering continuous change. *Future Gener. Comput. Syst.*, 117:29–46.

Bellman, K. L., Tomforde, S., and Würtz, R. P. (2014). Interwoven systems: Self-improving systems integration. In *IEEE Int. Conf. on Self-Adaptive and Self-Organizing Systems Workshops 2014*, pages 123–127.

Bellman, R. (1978). *Artificial Intelligence: Can Computers Think?* Thomson Course Technology.

Berger, T. (2001). Agent-based spatial models applied to agriculture: a simulation tool for technology diffusion, resource use changes and policy analysis. *Agricultural Economics*, 25(2-3):245–260.

Boden, M. A. (2016). *AI: Its nature and future*. Oxford Univ. Press.

Bongard, J. (2009). Biologically inspired computing. *IEEE computer*, 42(4):95–98.

Darwin, C. (1859). *On the Origin of Species by Means of Natural Selection*. Murray, London. or the Preservation of Favored Races in the Struggle for Life.

De Jong, K. A. (2016). *Evolutionary Computation: A Unified Approach*. MIT Press, Cambridge, MA, USA.

Diaconescu, A., Frey, S., Müller-Schloer, C., Pitt, J., and Tomforde, S. (2016). Goal-oriented holonics for complex system (self-)integration: Concepts and case studies. In *2016 IEEE 10th International Conference on Self-Adaptive and Self-Organizing Systems (SASO)*, pages 100–109.

Holland, J. H. (1992). Complex adaptive systems. *Daedalus*, 121(1):17–30.

Jennings, N. R. (2000). Building Complex Software Systems: The Case for an Agent-Based Approach. *Communications of the ACM*, 44:12–23.

Kantert, J., Edenhofer, S., Tomforde, S., Hähner, J., and Müller-Schloer, C. (2016). Normative control: Controlling open distributed systems with autonomous entities. In *Trustworthy Open Self-Organising Systems*, pages 89–126. Birkhäuser.

Kephart, J. O. and Chess, D. M. (2003). The vision of autonomic computing. *Computer*, 36(1):41–50.

Kounev, S., Lewis, P., Bellman, K. L., Bencomo, N., Camara, J., Diaconescu, A., Esterle, L., Geihs, K., Giese, H., Götz, S., et al. (2017). The notion of self-aware computing. In *Self-Aware Computing Systems*, pages 3–16. Springer.

Langton, C. G. (1992). Preface. In Langton, C. G., Taylor, C., Farmer, J. D., and Rasmussen, S., editors, *Artificial Life II, Volume X of SFI Studies in the Sciences of Complexity*, pages xiii–xviii, Redwood City, CA. Addison-Wesley.

Lehman, J. and Stanley, K. O. (2008). Exploiting open-endedness to solve problems through the search for novelty. In *Proc. of the Eleventh Intl. Conf. on Artificial Life (ALIFE XI)*, Cambridge, MA. MIT Press.

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

Lewis, P. R. (2017b). Self-integrating and self-improving systems must be socially sensitive. In *IEEE Int. Works.s on Foundations and Appl. of Self* Systems*, pages 148–148. IEEE.

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

Lewis, P. R., Chandra, A., Parsons, S., Robinson, E., Glette, K., Bahsoon, R., Torresen, J., and Yao, X. (2011). A survey of self-awareness and its application in computing systems. In *Proceedings of the Fifth IEEE International Conference on Self-Adaptive and Self-Organizing Systems Workshops (SASOW)*, pages 102–107. IEEE Computer Society Press.

Lewis, P. R., Platzner, M., Rinner, B., Torresen, J., and Yao, X., editors (2016). *Self-Aware Computing Systems: An Engineering Approach*. Springer.

Mnif, M. and Müller-Schloer, C. (2011). *Quantitative Emergence*, pages 39–52. Springer Basel, Basel.

Müller-Schloer, C. and Tomforde, S. (2017). *Organic Computing – Techncial Systems for Survival in the Real World*. Birkhäuser.

Norman, D. A. (1998). *The Invisible Computer*. MIT Press.

Reif, W., Anders, G., Seebach, H., Steghöfer, J.-P., André, E., Hähner, J., Müller-Schloer, C., and Ungerer, T. (2016). *Trustworthy Open Self-Organising Systems*. Springer.

Rogers, A., David, E., Jennings, N. R., and Schiff, J. (2007). The effects of proxy bidding and minimum bid increments within eBay auctions. *ACM TWEB*, 1(2):9.

Rudolph, S., Tomforde, S., and Hähner, J. (2019). Mutual influence-aware runtime learning of self-adaptation behavior. *ACM Trans. Auton. Adapt. Syst.*, 14(1):4:1–4:37.

Russell, S. J. and Norvig, P. (2020). *Artificial Intelligence: A Modern Approach (4th Edition)*. Pearson.

Sha, L., Gopalakrishnan, S., Liu, X., and Wang, Q. (2008). Cyber-physical systems: A new frontier. In *2008 IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (sutc 2008)*, pages 1–9. IEEE.

Snyder, T. and Byrd, G. (2017). The internet of everything. *IEEE Computer Architecture Letters*, 50(06):8–9.

Stanley, K. O. and Lehman, J. (2015). *Why Greatness Cannot Be Planned: The Myth of the Objective*. Springer.

Sun, R. and Naveh, I. (2004). Simulating Organizational Decision-Making Using a Cognitively Realistic Agent Model. *Journal of Artificial Societies and Social Simulation*, 7(3).

Taylor, T., Bedau, M., Channon, A., Ackley, D., Banzhaf, W., Beslon, G., Dolson, E., Froese, T., Hickinbotham, S., Ikegami, T., McMullin, B., Packard, N., Rasmussen, S., Virgo, N., Agmon, E., Clark, E., McGregor, S., Ofria, C., Ropella, G., Spector, L., Stanley, K. O., Stanton, A., Timperley, C., Vostinar, A., and Wiser, M. (2016). Open-Ended Evolution: Perspectives from the OEE Workshop in York. *Artificial Life*, 22(3):408–423.

Tennenhouse, D. (2000). Proactive computing. *Communications of the ACM*, 43(5):43–50.

Tomforde, S. and Müller-Schloer, C. (2014). Incremental design of adaptive systems. *Journal of Ambient Intelligence and Smart Environments*, 6(2):179–198.

Tomforde, S., Prothmann, H., Branke, J., Hähner, J., Mnif, M., Müller-Schloer, C., Richter, U., and Schmeck, H. (2011). Observation and Control of Organic Systems. In *Organic Computing - A Paradigm Shift for Complex Systems*, pages 325 – 338. Birkhäuser.

Tomforde, S., Sick, B., and Müller-Schloer, C. (2017). Organic computing in the spotlight. *CoRR*, abs/1701.08125.

Weiser, M. (1999). The computer for the 21st century. *ACM SIGMOBILE Mob. Comput. Commun. Rev.*, 3(3):3–11.

Wiener, N. (2019). *Cybernetics or Control and Communication in the Animal and the Machine*. MIT press.

Wooldridge, M. J. (2009). *An Introduction to MultiAgent Systems*. John Wiley & Sons Publishers, 2nd edition.