Engineering and Mastering Interwoven Systems

Sven Tomforde, Jörg Hähner, Hella Seebach, Wolfgang Reif
University of Augsburg
{sven.tomforde|joerg.haehner|seebach|reif}
@informatik.uni-augsburg.de

Bernhard Sick, Arno Wacker
University of Kassel
{bsick|arno.wacker}
@uni-kassel.de

Ingo Scholtes
ETH Zürich
ischoltes@ethz.ch

Abstract—**Networked systems are becoming increasingly complex in development and operation. Due to this complexity, it is mostly impossible to follow a simple sequential design-deploy-use cycle. Instead, development and operation will become more evolutionary in nature. Additionally, one can observe that individual complex systems are coupled with each other, even though this has never been intended in the early development of these systems. As a result, we are facing interwoven systems – multiple open time-variant systems are coupled and interact having, e.g., different goals and objectives as well as changing system and communication structure. Based on and extending the idea of composing Systems of Systems, this article identifies challenges that are becoming increasingly apparent as the inevitable integration of systems progresses.**

I. MOTIVATION

Today's Information and Communication Technologies (ICT) have become so ubiquitous that they are integrated in virtually every aspect of our life. In this perspectives article, we argue that this integration not only crucially influences our society – it also feeds back on the modelling, analysis of today's and future ICT infrastructures. Even today, most of our technical infrastructures cannot be understood as a single system anymore: ICT systems are becoming an increasingly integral part of energy networks, thus forming interdependent layers that strongly influence each other. Similarly, the structure and dynamics of social organisations play an important role in a variety of today's ICT systems either explicitly, such as in the case of social software, or implicitly, such as in information or recommendation systems that are influenced by collective human behaviour. As another example, consider transportation infrastructure and supply networks, which – through an integration of transportation modes such as airlines, railways and roads – are multi-layered systems by themselves. In addition, these multi-layered systems are tightly coupled to ICT infrastructures, energy networks, and collective human behaviour, thus forming a complex multiplex of interconnected systems outside the control of a central organisation – we refer to this by the term of **"Interwoven Systems"**.

Traditionally, we have focused on the modelling, design, and analysis of rather isolated systems whose boundaries could clearly be defined based on geography, the set of used technologies, or a sphere of organisation influence. In this article, we argue that we are increasingly confronted with the challenge of mastering complex interwoven systems, which – in addition to being large and consisting of heterogeneous and potentially unreliable components – are tightly coupled with other systems in a way that blurs system boundaries and thus invalidates design principles such as separation of concerns. This development is facilitated by increasing interconnectedness of infrastructures and technologies, virtualisation techniques, and the general trend towards globalisation. Interestingly, the challenges resulting from this coupling are likely to be aggravated by the adoption of systems with self-adaptive and self-organising characteristics that are now becoming increasingly common. The adaptation of a system to changing environmental conditions is already a hard problem that has nevertheless been addressed successfully in a number of research projects. In most real-world scenarios, however, these changing environmental conditions are formed by other systems which have self-adaptive properties themselves. By this, they not only exhibit time-variant features that do not easily allow an outside prediction of their behaviour. The complex feedback resulting from coupling several self-adaptive systems can also give rise to emergent behaviour which cannot be predicted by studying a single system alone. In fact, even today there are numerous examples for interwoven systems that can give rise to such complex phenomena: When changing energy systems and economic incentives in a way that makes them adapt to varying demand, it is crucial to take into account models for the collective response of consumers, who again adapt to these changes. Information and collaboration systems that adapt to the time-varying social dynamics of their users are likely to influence the very dynamics they adapt to, thus forming a complex interwoven system whose behaviours is difficult to model and predict and which is, thus, difficult to design. Finally, even systems which are not directly coupled with collective human behaviour, e.g., large-scale data centres, have been shown to be prone to unexpected phenomena that are due to the interference between different subsystems.

We argue that the increasing deployment of self-adaptive and self-organising ICT systems and their integration with other natural or man-made self-organising systems opens a novel set of research challenges that needs to be addressed by the computer science community. In the remainder of this article, we introduce some of these challenges and provide real-world examples for systems in which they are likely to occur. Summarising existing interdisciplinary theories and conceptual frameworks that can serve as cornerstones for an integrated design of interconnected self-adaptive systems, we further highlight some solution perspectives and call for a concerted research agenda.

II. INTERWOVEN SYSTEMS

As motivated in the previous section, the most important challenge for systems engineering within the next decade will be to deal with ultra-large-scale systems (see e.g. [1]) consisting of e.g. heterogeneous, exchangeable, and time-variant elements. The term "heterogeneous" focuses on the origin: The overall system consists of interconnected elements that can be newly engineered, commercial off-the-shelf (COTS), and existing or legacy (see, e.g., [2]). Furthermore, "exchangeable" refers to the continuous evolution of such systems in the sense of replacing, adding, or removing components, while "time-variant" additionally considers time-varying environments (and, thus, systems). This section describes the major corner stones of interwoven systems, followed by a term definition and characterisation of such systems.

A. *Corner Stones of Interwoven Systems*

In order to be able to describe such a class of systems, the term "System of Systems" (SOS) [3] serves as basis. No commonly agreed definition of the term SOS is available. The most prominent ones are the following: "A system is a collection of entities and their interrelationships gathered together to form a whole greater than the sum of the parts." [4] and "A system of systems is a set of different systems so connected or related as to produce results unachievable by the individual systems alone." [5].

System of Systems:
From these definitions, it is apparent that an SOS is an aggregation of heterogeneous systems, their resources, and their capabilities. The set of individual systems together creates a new, more complex system – the SOS. Thereby, this SOS is not just the sum of the constituting *component systems* (i.e., a self-motivated stand-alone system as part of the SOS), but offers more functionalities and capabilities or results in higher performance in terms of a system-wide objective (compared to the performance of the individual systems acting on their own), see e.g. [6]. This definition of an SOS leads to the observation that current processes, tools, and design methods are inadequate or at least incomplete for design, development, and maintenance purposes in a technical environment [4].

According to Maier, an SOS can be distinguished from more conventional systems by considering fundamental characteristics [7]. Thereby, a first major consideration is that in an SOS "the various components are large-scale systems in their own right". This refers to the autonomy and independence of each SOS as well as to the fact that each SOS is in itself a collection of autonomous systems. Furthermore and according to [2], an SOS is characterised by five fundamental characteristics:

1) **Operational Independence of the Individual System**: Each SOS consists of a set of independent systems. In this context, *independent* means that each contained system acts on its own and fulfils its own goals – if the SOS is decomposed, each system maintains its own performance.
2) **Managerial Independence of the System**: Each component system within the SOS is integrated and maintained individually. This means that control and maintenance of the component system do not depend on the existence of the SOS.
3) **Geographic Distribution**: Typically, the component systems within the SOS are spatially distributed. The aggregation towards an SOS is done using communication methods.
4) **Emergent Behaviour**: Each component system has its own behaviour. The resulting behaviour of the SOS is more than just the aggregation of individual behaviour of the contained component systems. Instead, novel functionality appears that is not directly apparent from the individual components.
5) **Evolutionary Development**: An SOS is not a static entity – its structure, organisation, and functionality change over time. Only limited external control is applied which leads to a self-evolutionary process.

Challenges of upcoming systems go far beyond the SOS definition and characterisation. Especially, communication mechanisms between component systems have to be established. Systems have to be able to automatically learn to exchange understandable problem descriptions for collaborations in the sense of application-oriented information. This might be realised by dynamic ontologies, since component systems and challenges change during runtime.

Federations of Systems:
In various fields of technology, a potentially large set of systems is required to fulfil a certain common objective or perform a complex mission. Thereby, only very limited (if at all) centralised control is available to achieve a synchronised and coordinated behaviour of the involved SOS. In such a case, a *"Federation of Systems"* (FOS – see [2]) is needed. The term FOS has been defined by Krygiel as an SOS "[...] managed without central authority and direction." [5]. This means that the constituting member SOS in a FOS are completely independent of each other (i.e., belong to different authorities) and self-motivated (i.e., pursue their own goal). Due to the lack of centralised control, collaboration and cooperation are required with self-organised processes to collectively come to a decision. Consequently, an FOS is characterised by three key-aspects: (1) a high degree of autonomy, (2) heterogeneity in terms of participating SOS, and (3) distribution of organisation structure and decision processes.

Challenges of upcoming systems go far beyond the scope of FOS. For instance, multiple goals can exist at runtime that are inconsistent or even conflicting. Complex tasks to be solved by federations require a problem-specific composition of these federations and a hierarchical structure of the organisation (i.e., according to specific abilities and a coordinated behaviour at different layers of abstraction). Such hierarchies have to be established by the systems themselves, they also have to be terminated or modified in response to changes in the conditions, requirements and goals.

B. *Perspectives Beyond SOS: Interwoven Systems*

The terms SOS and FOS are used to define a concept for describing current technical systems. In this section, we provide an interpretation how this description can be used as basis for mastering Interwoven Systems (IS) and the resulting complexity. Thereby, IS aims at developing a new paradigm with a special focus on system design, cooperation and interaction between component systems, and the behaviour at runtime. This paradigm is intended to simplify the mastering of such systems and to result in novel techniques for design, operation, observation of mutual influences, or coupling and extendibility of such systems.

Characterisation of Interwoven Systems:
IS builds upon the previous definitions and characterisations of SOS and FOS and consequently consist of a set of more or less autonomous component systems that might again consists of autonomous systems. Therefore, the previous five characteristics of SOS remain valid, but are further augmented with a set of important characteristics.

1) **Operational Independence of the Individual System**: Due to scalability reasons and existence of various administrative authorities, a centralised control for IS is not possible. Thus, IS are characterised by self-organisation of systems and their federation. This challenging characteristic for current and future networked systems can also be observed in initiatives such as Autonomic and Organic Computing [8], [9].
2) **Managerial Independence of the System**: Although individual systems might belong to the same authority, they are handled as independent in order to keep the maintenance problem at a manageable level. IS can be characterised by varying administrative domains.
3) **Geographic Distribution**: Apparently, an IS consists of a set of interconnected systems. This implies a distribution of the contained component systems, although it does not necessarily demand a wide-spread geographical distribution. In contrast, large data centres can also be understood as IS, where the geographical distribution is, e.g., restricted to a certain building. From a data-perspective, a synchronisation of data reflecting the time-variance of IS is needed. Such an IS is characterised by the possibility to separate the component systems in the sense of defining system boundaries as administrative domains on the basis of geographical separation.
4) **Emergent Behaviour**: Operational independence (i.e., self-organisation) and local interaction can result in emergent behaviour [10]. These emergent effects can either be positive or negative. Especially coupling independent IS at a large-scale can result in unanticipated behaviour – which is referred to as *emergence*. Hence, IS are characterised by being able to recognise emergent behaviour and act accordingly.
5) **Evolutionary Development**: IS are seldom deployed in one piece and by one authority – instead, their composition changes over time and at runtime. In combination with the self-organisation aspect, this leads to the need of transferring design-time decisions to runtime and into the responsibility of the individual systems. Thereby, the system is allowed to adapt itself to changing conditions and consequently self-optimise its own behaviour. Thus, an IS is characterised by a continuous evolution that does not rely on user interaction.

On top of this, the term IS highlights the networked nature and distributed self-organised management and control of these systems. This results in further characteristics that go far beyond the SOS description:

6) **Mutual Influences of Networked Systems**: The self-organising and self-optimising behaviour of individual component systems in combination with coupling and interconnection of systems leads to the effect of mutual influences: If one component system adapts its behaviour, the interconnected component systems might also need to modify their behaviour in response to this change. Such a coupled system can easily lead to oscillating and uncontrollable behaviour. Thus, an IS is characterised by the demand of a federative approach and a kind of smartness within the cooperation and coupling between the component systems in order to avoid such undesired and unstable behaviour.
7) **Heterogeneity of Component Systems and Federations**: A major aspect of an IS is the openness of the networked system and, consequently, the heterogeneity of the contained component systems. Since no central authority is responsible for managing the IS, and federations can appear in response to specific problems, no membership process and control is available. Consequently, a designer of a component system has to be aware of the fact that (from an individual perspective) collaboration partners of the system can behave in an unpredictable or even malicious way and (from a system-wide perspective) the IS organisation and structure continuously change. Hence, an IS is characterised by security and trustworthy mechanisms to deal with openness and heterogeneity.
8) **Uncertainty**: The heterogeneity, the self-organised adaptation of component systems, and the continuous evolution of such systems result in a limited predictability of their behaviour. Hence, IS are characterised by uncertainty in the system's behaviour and decisions under uncertainty.

Term Definition: "Interwoven Systems":
We refer to a system as an IS if at least the previously defined eight characteristics are fulfilled. An IS is a system consisting of coupled and interacting component systems that communicate directly or indirectly with each other. Such an IS is an ultra-large-scale system which faces changes at runtime that are not defined or even anticipated at design-time: individual component systems can change in architecture, parametrisation, and goals, while the overall IS can change, e.g., in terms of number of contained component systems, communication infrastructure, logical structure of the collaboration between component systems, or the set of goals to be achieved by the IS. The behaviour of the IS is not just an aggregation of the functions of the component systems – instead, it is a result of the functional repertoire, the resolution of possibly conflicting goals, and the interaction between component systems.

First, the main challenges for IS are a collaborative analysis and modelling of the situation. Second, the management of the component systems is required to achieve a balanced and goal-oriented behaviour of the IS – this means to handle this class of systems and autonomously maintain the desired behaviour and performance. In Section V, we define in detail the research challenges resulting from this problem definition.

III. USE CASES

A. High-performance Computing Landscape

The modern High-Performance Computing (HPC) landscape used for the execution of heterogeneous computing tasks is one example for IS. Typically, such systems consist of multiple (possibly multi-site) compute clusters. Each cluster contains a number of compute nodes, which are connected using various communication technologies, such as Ethernet or Infiniband. Often, the particular communication technology used within a single cluster (or even rack) is different to the one used to connect different clusters with each other. Each compute node itself is a complex sub-system that may contain heterogeneous processors, memory sub-systems, and communication devices. Examples include CPUs, GPUs, and other custom elements for processing, different memory hierarchies, and networking interfaces. For cost reasons, many sub-systems used in such HPC systems are COTS components developed by different vendors independently from each other.

Large HPC systems clearly fulfil characteristics (1) – (3) of IS as, for scalability, fault-tolerance and economic reasons, they are often divided up into different separated data centres. Emergent effects can already be observed in negative examples, e.g., the temporary Google blackout from August 16th, 2013[1]. This is – among other reasons – due to the fact that most HPC systems are not built in one step before being taken into service. Instead, it is favourable to incrementally build these systems as the demand for services increases. This can also lead to systems where multiple generations of hardware and software have to interact with each other. Since not all combinations have been tested in advance, this may lead to diverse interactions and behaviour with, sometimes, sub-optimal performance or system failures.

B. Power Management Systems

Today's power management systems (PMS) are already SOS, but future PMS will change towards an IS. While the whole PMS already consists of a huge amount of power plants, each power plant can still operate independently from the others (1). Managerial independence of the PMS (2) is fulfilled since, to a certain extent, the power plants operate economically independent. Point (3) is given through the natural geographical distribution, especially since the amount of distributed energy resources such as biogas plants, solar plants, and wind farms increases significantly (see Acatech Future Energy Study [11]). The PMS's stability is an emergent behaviour (4) since no single power plant can provide this stability on its own. The state of current, centralised, manually managed PMS is already the result of an evolutionary development (5) and future grid structure and management will also evolve.

Due to the growing number of generators, the increasing dependence on unreliable sources (wind, solar), and the increasing ability to control distributed energy resources, the properties (6) to (8) of IS become a necessity for mastering future PMS. Some techniques from SOS [12] and self-organising hierarchies [13] are already successfully applied to simulations of PMS. These results can be used in other SOS such as gas distribution grids or district heating systems, but the combination and coupling of these resulting SOS still remain very complex and unpredictable. The coupling of several SOS and networks lead to new challenges, e.g., in terms of mutual influences and positive as well as negative feedback loops between these networks (6). This can be seen, e.g., in the coupling of district heating systems and PMS. The output of a combined heat and power plant (CHP) might be increased in case more heat is requested from a distinct heating system than initially expected. As a CHP produces both, heat and power, this might cause a surplus of power within the PMS. These interfering feedback loops must be investigated and controlled. Heterogeneity is a given facet of PMS (7) because of the nature and evolved architecture over time. Uncertainty comes into these systems, e.g., by unreliable sources (wind, solar), and unreliable communication channels (8).

C. Vehicular Traffic

Traffic control and management serve as another example for the existence of IS. Optimal control strategies for traffic lights depend on the observed traffic conditions in terms of vehicles passing the underlying intersection. Thereby, the task of such an intersection controller varies in terms of the controlled intersection's topology and its position in the network (i.e., residential area, arterial road, or highway). Besides the setup of green times, control strategies can reflect coordination in the sense of developing progressive signal systems.

Management of traffic adds further active components to the reactive traffic control system (switching traffic lights is a reaction to observed conditions, *management* actively influences traffic). Guiding drivers through the network, recommending routes and incorporating a detection of shortages in capacity or incidents within the control of traffic lights and the driver guidance leads to mutual influences between geographically distributed elements and emergent behaviour. Thereby, traffic behaviour and estimations of the system's state are characterised by uncertainty and time-variances. The transportation system becomes even more heterogeneous if taking other carriers, such as aircrafts, railways and pedestrians, into account.

Intersections and carriers belong to different authorities, but the performance of the overall system has a broader scope than just one managerial instance. The goal of engineers is to develop solutions that, e.g., minimise pollutions and the number of stops per passenger traversing the network. Measuring such aspects for closed environments (i.e., a certain urban district) is limited as effects from neighbouring network parts are faced. In order to achieve balanced and scalable solutions, current research already focuses on the development of dynamic hierarchical components and distributed control among self-organised intersection controllers [14], [15]. As a result, traffic control and management systems are IS consisting of heterogeneous populations (i.e. vehicles with varying behaviour, topologically different intersections and road elements, and different authorities).

D. Socio-technical systems

Another important area of application of an IS perspective is the design of *socio-technical systems*. *Socio-technical* systems integrate a *social* and *technical* layer in a way that gives rise to ICT systems that cannot be understood by studying a single layer alone. Today, we are surrounded by numerous systems which can be characterised as socio-technical [16]. Obvious representatives of such systems are all kinds of *social software* such as wikis, blogs, online social networks, communication systems such as instant messaging and E-Mail, but also collaboration tools and social platforms increasingly used in an enterprise context. An important aspect of many of these systems is that the two layers influence each other. As an example, one can consider collaboration tools regularly used in distributed software engineering projects. Clearly, interaction mechanisms implemented in such technical systems influence the social layer, i.e., if and how users of the system will collaborate and how the social organisation of software development teams will evolve over time. The reverse influence, i.e., the influence of the social layer on the technical layer has so far been rather limited and operated on rather long time scales. Today, we see an increasing adoption of systems that are *socially aware* in the sense that they monitor the structure and dynamics of social organisations and adapt their behaviour accordingly in real-time.

Such socio-technical systems are increasingly being influenced by the structure and dynamics of societies and the

---

[1] see e.g. http://www.informationweek.com/security/vulnerabilities-and-threats/googles-four-minute-blackout-examined/d/d-id/1111211

resulting processes that link pieces of content to each other. At the same time, the ranking of information, e.g., implemented by search engines, feeds back on society and thus influences how future links are being formed and how information is being filtered. The last decades have seen remarkable advances in the design and management of complex ICT systems. At the same time, by means of agent-based modelling techniques and the statistical analysis of massive scale data sets, we begin to understand the mechanisms behind collective social phenomena and how they are influenced by different interaction mechanisms [17], [18].We argue that the increasingly tight coupling between modern ICT systems and large-scale social organisations calls for an integrated systems design of socio-technical systems [19].

IV. RELATED METHODOLOGY AND BUILDING BLOCKS FOR INTERWOVEN SYSTEMS

Mastering the increasingly coupled complex systems that today's ICT infrastructures rest upon will inevitably need to be rooted in the following existing lines of research.

A. Organic Computing

Organic Computing (OC) [8] is based on the insight that we are increasingly surrounded by large collections of autonomous systems, which are equipped with sensors and actuators, are aware of their environment, communicate freely, and organise themselves in order to perform the actions and services that seem to be required. Consequently, OC designs and develops technical systems which are equipped with sensors (to perceive their environment) and actuators (to manipulate it). Such an *organic* system adapts autonomously and dynamically to the current conditions of the perceived environment. This adaptation process has impact on the system's performance, which is continuously optimised by the organic system itself. OC systems are characterised by self-X properties (similar as, e.g., formulated for the Autonomic Computing initiative [9]). Thereby, the current research activities focus on mostly homogeneous and closed entities: Although OC introduces self-organisation capabilities and makes use of autonomy to solve technical problems, the systems are still designed with a certain perspective regarding the functional goal. This means, the interaction of heterogeneous and continuously changing component systems describes novel challenges for the OC domain as the resulting influences between different independent systems are mostly neglected using abstractions. When considering mutual dependencies between large-scale IS of possibly different domains, novel challenges appear that cannot be covered with the existing OC technology.

B. Multi-Agent Systems

The term *Multi-Agent System* (MAS) [20] describes a system consisting of several homogeneous or heterogeneous (in terms of capabilities) entities, so-called *agents*, that collectively solve a given problem. Research on MAS focusses especially on interaction between these agents, their cooperative behaviour, and the decision processes of individual agents within the overall system. The agent definition has been augmented by and compared to holons and holonic approaches [21] – holons are comparable to SOS consisting of homogeneous entities, while IS are focusing on mostly heterogeneous and time-variant entities. The IS paradigm builds upon the ideas of MAS and introduces novel concepts for, e.g., the effects of coupling several autonomous systems.

C. Extreme-scale Computing

Recently, extreme-scale [22] and ultra-large-scale systems [1] have been identified as the software challenge of the future. Thereby, extreme-scale computing incorporates several important research and development challenges related to scalability. This includes the parallelisation of computing tasks, corresponding computer architectures, load balancing, and synchronisation issues. Hence, concepts for mainly hardware-based scalability in terms of controlling large populations of, e.g., computing nodes exist, but a perspective towards other networked systems and the mutual influences between autonomously interacting systems is not considered. An important insight from the community is that more of the same (again in terms of computing nodes) is not enough to handle complex (calculation) tasks.

Similarly, ultra-large-scale computing [1] (ULSC) is focusing on systems with a high amount of contained elements. ULSC has its background in cyber-infrastructures and is therefore closer connected to IS. The observation made here is that emerging systems are already ultra-large-scale in terms of geographical distribution and the number of contained elements. In addition, heterogeneous components are contained that work to a certain degree on the basis of self-organisation. ULSC postulates to enable runtime adaptation of systems through model adaptation and corresponding code generation. This is based on the existence of models that also reflect the impact of interactions and mutual influences within the whole networked systems, an assumption not considered valid for IS.

D. Control Theory

Control theory is an interdisciplinary field of mathematics and engineering and relevant for the control of various physical processes [23], [24]. A major part of the field covers so-called *closed loop systems*, where actions of the system's controlling element influence the input that is used to decide about these actions. This concept is also described as *feedback* processes. The approach to handle these processes and incorporate the observed feedback mainly depends on the existence of complete and consistent (mathematical) models. In reality, systems seldom consist of isolated control loops; in contrast, coupled loops can be observed,resulting in a complex problem.

Future IS will combine different control principles ranging from centralised/supervisory decision routines over cooperative/collaborative procedures for groups of subsystems down to local and decentralised regulation for single subsystems. The control algorithms on all layers essentially have to account for time-varying goals, constraints, and interactions with the environment or among the subsystems. In contrast to most existing control strategies of centralised and decentralised control, decision procedures for IS will have to be tuned to high degrees of adaptability in online operation. This requirement raises a plethora of challenges in the development of control algorithms, including changing availabilities of sensor data, consideration of time-varying constraints, the heterogeneity of coupled dynamical subsystems, controllability and observability issues, robustness and stability of feedback controllers which are adapted online, and the timeliness of computing decision and control actions within the various decision instances.

E. Complex Systems Theory

During the last decades, biological, social, technical, and economic systems have increasingly been studied from the perspective of *complex systems*. Emphasising the fact that such systems are composed of numerous interacting elements with possibly heterogeneous characteristics, *complex systems theory* focuses on systemic properties that emerge in large aggregates of interacting elements. Methods that originated in the study of many-particle systems in statistical physics as well as agent-based modelling techniques allow to relate collective phenomena occurring at the macroscopic level of a system with the microscopic level [25]. Approaches which are now summarised as complex systems theory can be traced back over several decades and have been used in studies of pattern formation [26], critical phenomena such as percolation [27], scale-invariant behaviour [28], or swarm behaviour [29].

The increasing availability of data has recently facilitated to augment this perspective, taking into account actual *interaction networks* observed in real-world complex systems. During the last two decades, the associated field of *complex networks* has identified a number of topological features that crucially influence collective dynamics such as diffusion or epidemics [30], [31], synchronisation [32], as well as the robustness of systems against cascading failures or targeted attacks [33], [34]. While these findings are certainly interesting, the need to go beyond the commonly employed *single system perspective* has recently been acknowledged in the community studying complex networks. This need is based on the fact that most real-world systems such as, e.g., social networks, power grids, transportation networks, information networks, or various types of technical infrastructures cannot easily be understood as a single system governed by a single interaction topology. They rather consist of multiple interwoven layers of complex networks, with nodes and links in different layers having different characteristics. Ongoing research projects are studying the extension of methods from the study of complex networks to such systems which – depending on the community – are referred to as *multiplex networks*, *multi-layer networks*, *interdependent networks* or *networks-of-networks* [35], [36], [37], [38], [39].

Apart from such multi-layer approaches, a further recent development in the field of complex networks is the focus on *network dynamics*. Clearly, interaction topologies in real complex systems are not static but rather vary over time. Social interactions, economic relations or links in communication systems rather change over time, thus giving rise to an additional level of complexity which – thanks to time-resolved data – is being studied in the area of *temporal* or *dynamic networks* [40]. One of the interesting recent findings is that order-correlations in the temporal dynamics of interaction topologies can significantly change systemic properties compared to what one would expect when studying them from the traditional, time-aggregated perspective [41], [42]. We expect these theoretical works on complex multi-layer and temporal networks to be an important cornerstone for the modelling, analysis and design of future interconnected ICT infrastructures.

F. Cyber-Physical Systems

There are several definitions for Cyber-Physical Systems (CPS) [43], [44], [45]. In [45], we see a common definition that includes the requirement that the system has multiple entities which are interconnected. The authors of [44] state that CPS are the integration of computation with physical processes. Hence, single devices are included in the definition. CPS connect two quite different worlds, the world of embedded systems (with real-time requirements, sensors and actuators, dependability, deterministic behaviour) with the world of digital networks (with globally available services, data clouds, multi-modal man-machine interfaces). CPS have to evolve with the environment they are situated in and with changing user demands. This makes CPS an ideal application domain for methods and the general paradigm of IS.

V. RESEARCH CHALLENGES AND SOLUTION PERSPECTIVES

Solutions towards the manageability of IS have to consist of multiple and partially orthogonal perspectives that also allow for the combination of existing methodologies with new approaches. We suggest to distinguish between three distinct perspectives: the *System Perspective (SP)*, the *Process Perspective (PP)*, and the *Data Perspective (DP)*. Depending on a particular design task, analysts, designers, and developers may look at an IS from one of these perspectives. The System Architect, for example, whose tasks include the design of a viable system architecture for a given set of problems typically has a component-oriented view on the overall system. Questions to be answered here include the choice of core components (hardware and software) and their interconnections. In contrast, the Business Analyst views tasks from the Process Perspective. This includes identifying the organisation's operating and business models and breaking them up into a set of processes that have to be supported by the IS. Finally and orthogonal to the SP and PP, the Data Perspective involves the identification of relevant information, their sources, and when and where to process and store it. This involves matching the demands of both SP and PP.

A. System Perspective

- At runtime (i.e., online) IS must adapt to a time-variant environment. The IS's components observe new processes that emerge, existing ones that become obsolete, or others that change their characteristics. At a more abstract level, these phenomena may be characterised as novelty, obsoleteness, and concept shift or drift. How can IS react to such phenomena? First, in order to detect them, they must be aware of their own capabilities and weaknesses. They must be able to detect and measure time-variance in their environment (which is much more than emergence detection, as various phenomena have to be detected and assessed gradually), and recognise when they have to adapt their facilities. Such an adaptation may include parameters but also the architecture of the IS's components or the IS itself. Generative modelling techniques will play an important role to achieve this, e.g., from the field of probability theory [46].
- IS will evolve and learn in a highly autonomous way, but not without any interference of humans. How can the degree of autonomy be increased? First, it is necessary to investigate how knowledge can be "injected" into the system, e.g., knowledge contained in simulation systems or stored in ontologies. Second, the components have to learn from each other, e.g., by communication or approaches such as imitation. Third, humans have to be part of the

process, but they have to be integrated in a very efficient and effective way, e.g., by new (collaborative) active learning approaches.
- IS will be subject to security attacks. This requires to select, (possibly) adapt, and integrate suitable methods for security and trust management.
- From a systems point of view, resources are added and removed continuously at runtime. This adds challenges with respect to fault tolerance, but also requires that systems can cope with changing input parameters, (e.g. provided by sensors) and changing output modalities (e.g. actuators).
- While the number of coupled systems increases, the resulting overall complexity gets out of hand. However, it is necessary to still be able to derive and guarantee system properties. The basis here will be found in the domain of system self-organisation and Complex Systems by shifting many typical design-time decisions into the runtime of the system and by investigating the correlation between local and global behaviour within the system.

B. *Process Perspective*

- IS are heterogeneous open systems that can change their structure at runtime. A major challenge is to enable subsystems to (semantically) have a common "language". This involves all aspects from communication between systems, e.g., discovery, negotiation, and collaboration. Exemplary first approaches may be found in the domain of Internet-of-things, and include ontologies.
- IS must guarantee to hold certain constraints, e.g., functional or temporal constraints. Constraints may be hard (e.g., to guarantee the safety of humans) or soft (e.g., to reduce energy costs), and typically we may have several. How can we ensure that such diverse constraints are met? For example, approaches from conventional control theory (e.g., for hard constraints) could be combined with data-driven optimisation techniques from the field of machine learning (e.g., for soft constraints) or artificial intelligence. In particular, we have to investigate the stability of networked systems with time-invariant structure and we have to develop methods for a synthesis of distributed control mechanisms.
- IS work in uncertain environments, i.e., observations or measurements are uncertain, the parametrisation of models is uncertain and communication is uncertain. The meaning of the term *uncertainty* may be adopted from [47]. There, "uncertain" is a generic term for other terms such as "likely", "doubtful", "plausible", "reliable", "imprecise", "inconsistent", or "vague". In IS, we have to cope with various kinds of uncertainty. How can we deal with uncertainty, how can we quantify it, and how can this knowledge eventually be used? Solutions can be found in the field of possibilistic or probabilistic techniques. Uncertainty at several levels can, for example, be addressed with second order techniques such as type-II fuzzy systems or second-order probability distributions.
- The uncertainty at design-time leads to the need of self-optimisation at runtime. In contrast to typical optimisation problems, IS have to find best-effort, near-to-optimal solutions. The overall problem is *concurrent optimisation* among the distributed and self-adaptive component systems, i.e. contradicting optimisations, reconciliation, optimisation under constraints, and optimisation in real-time. Therefore, fast and efficient search algorithms have to be developed based on, e.g., nature-inspired search heuristics that take the connections to other related component systems into account.
- Sub-systems are being installed and maintained at different times. This results in situations where different variants of the same type are present. Despite this, the seamless interaction between sub-systems has to be guaranteed. Here, approaches based on the idea of reflective, anticipatory, and self-modelling systems will provide new methodology.
- IS are self-organising and can dynamically form groups, hierarchies, and other structures in order to perform a given task. In addition to approaches from various domains, new methods inspired by social and biological role models may lead to valuable contributions.

C. *Data Perspective*

- It will certainly not be possible to develop IS only in model-driven engineering approaches, i.e., using analytical or physical knowledge. The processes and systems that have to be modelled are not (completely) known. How can a conventional model-driven design approach be extended? We certainly need new approaches from the field of models@runtime [48] and combinations with data-driven engineering approaches based on appropriate data analysis and machine learning techniques (cf. also the field of fusion of hard and soft computing [49]).
- Processes within the IS have to create, process, and store data within the system. The challenge here is to provide methods for process-adaptive management and storage of this data. Solutions will need to address problems related to the self-organisation of the data management. This includes the seamless integration of methods for replica placement, consistency management, as well as storage and retrieval mechanisms which can, e.g., be found in distributed systems.

VI. CONCLUSION

In this article we have introduced the idea of *Interwoven Systems*, which was motivated by observations and challenges that arise from the inevitable evolutionary development and operation of complex coupled systems. The resulting systems are time-varying and open in nature and often show emergent behaviour. This is due to the fact that individual complex systems are coupled with each other, even though this has never been intended in the early development of these systems. The key definition is based on the idea of building systems of systems and is complemented by related approaches from research directions such as Organic Computing, Multi-Agent Systems, and Complex Systems. The paper closes with new research challenges that are aligned along three major perspectives that different types of developers typically take, depending for what they are responsible.

REFERENCES

[1] L. Northrop, *Ultra-Large-Scale Systems: The Software Challenge of the Future*. Software Engineering Institute; Carnegie Mellon, 2006.

[2] A. P. Sage and C. D. Cuppan, "On the Systems Engineering and Management of Systems of Systems and Federations of Systems," *Inf. Know. Sys. Man.*, vol. 2, pp. 325 – 345, 2001.

[3] W. A. Owens, “The Emerging U.S. System of Systems,” in *Dominant Battlespace Knowledge*, S. Johnson and M. Libicki, Eds. Washington DC, US: National Defence Unity Press, 1996.

[4] J. Boardman and B. Sauser, “System of Systems - the meaning of of,” in *Proc. of IEEE/SMC Conf. on System of Systems Engineering*, 2006, pp. 118 – 123.

[5] A. J. Krygiel, *Behind the Wizzard’s Curtain: An Integration Environment for a System of Systems*. Institute for National Strategic Studies, Washington, D.C., 1999.

[6] A. Meilich, “System of Systems (SoS) Engineering and Architecture Challenges in a Net Centric Environment,” in *Proc. of IEEE/SMC Int. Conf. on System of Systems Engineering*, 2006, pp. 1 – 5.

[7] M. W. Maier, “Architecting Principles for Systems-of-Systems,” in *Proc. of Council on System Engineering (INCOSE)*, 1996, pp. 567 – 574.

[8] C. Müller-Schloer, “Organic Computing: On the feasibility of controlled emergence,” in *Proc. of CODES and ISSS’04*. ACM Press., 2004, pp. 2–5.

[9] J. O. Kephart and D. M. Chess, “The Vision of Autonomic Computing,” *IEEE Computer*, vol. 36, no. 1, pp. 41–50, 2003.

[10] D. Fisch, M. Jänicke, B. Sick, and C. Müller-Schloer, “Quantitative Emergence – A Refined Approach Based on Divergence Measures,” in *Proc. of IEEE Conf. on Self-Adaptive and Self-Organizing Systems (SASO 2010)*, Budapest, Hungary, 2010, pp. 94–103.

[11] H.-J. Appelrath, H. Kagermann, and C. Mayer, “Future energy grid,” *Migrationspfade ins Internet der Energie. Acatech Studie Februar*, 2012.

[12] J.-P. Steghöfer, G. Anders, F. Siefert, and W. Reif, “A System of Systems Approach to the Evolutionary Transformation of Power Management Systems,” in *Proc. of INFORMATIK 2013 – Workshop on “Smart Grids”*, ser. LNI. Bonner Köllen Verlag, 2013.

[13] J.-P. Steghöfer, P. Behrmann, G. Anders, F. Siefert, and W. Reif, “HiSPADA: Self-Organising Hierarchies for Large-Scale Multi-Agent Systems,” in *Proc. of Int. Conf. on Autonomic and Autonomous Systems*, 2013, pp. 71 – 76.

[14] S. Tomforde, H. Prothmann, J. Branke, J. Hähner, C. Müller-Schloer, and H. Schmeck, “Possibilities and limitations of decentralised traffic control systems,” in *2010 IEEE World Congress on Computational Intelligence (IEEE WCCI 2010)*. IEEE, 2010, pp. 3298–3306.

[15] H. Prothmann, S. Tomforde, J. Lyda, J. Branke, J. Hähner, C. Müller-Schloer, and H. Schmeck, “Self-organised Routing for Road Networks ,” in *Proc. of IWSOS’12*, ser. LNCS 7166. Springer, 2012, pp. 48 – 59.

[16] J. Kleinberg, “The Convergence of Social and Technological Networks,” *Commun. ACM*, vol. 51, no. 11, pp. 66–72, Nov. 2008.

[17] F. Schweitzer, P. Mavrodiev, and C. J. Tessone, “How can social herding enhance cooperation?” *Advances in Complex Systems*, vol. 16, p. 22, March 2013.

[18] D. Garcia and F. Schweitzer, “Modeling online collective emotions,” in *Proc. of WS on Data-driven user behavioral modelling and mining from social media*. ACM Press, 2012, p. 37.

[19] R. P. Ingo Scholtes and F. Schweitzer, *The Social Dimension of Information Ranking: A Discussion of Research Challenges and Approaches*. Springer, 2014.

[20] M. J. Wooldridge, *An Introduction to MultiAgent Systems*, 2nd ed. Hoboken, NJ, US: John Wiley & Sons Publishers, 2009.

[21] A. Giret and V. Botti, “Holons and agents,” *Journal of Intelligent Manufacturing*, vol. 15, no. 5, pp. 645–659, 2004.

[22] A. Hoisie and V. Getov, “Extreme-scale Computing - Where ’Just More of the Same’ does not Work,” *IEEE Computer*, vol. 42, no. 11, pp. 24 – 26, 2009.

[23] R. A. Brooks, “How to build complete creatures rather than isolated cognitive simulators,” in *Architectures for Intelligence*. Erlbaum Publishers, 1991, pp. 225–239.

[24] K. H. Ang, G. Chong, and Y. Li, “PID Control System Analysis, Design, and Technology,” *IEEE transactions on control system technology*, vol. 13, no. 4, pp. 559–576, 2005.

[25] F. Schweitzer, *Brownian Agents and Active Particles: Collective Dynamics in the Natural and Social Sciences*. Springer, 2003.

[26] A. M. Turing, “The Chemical Basis of Morphogenesis,” *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences*, vol. 237, no. 641, pp. 37–72, 1952.

[27] H. E. Stanley, J. S. AndradeJr., S. Havlin, H. A. Makse, and B. Suki, “Percolation phenomena: a broad-brush introduction with some recent applications to porous media, liquid water, and city growth,” *Physica A: Statistical and Theoretical Physics*, vol. 266, no. 1-4, pp. 5–16, Apr. 1999.

[28] P. Bak, C. Tang, and K. Wiesenfeld, “Self-organized criticality: An explanation of the $1/f$ noise,” *Phys. Rev. Lett.*, vol. 59, pp. 381–384, Jul 1987.

[29] F. Schweitzer, W. Ebeling, and B. Tilch, “Complex Motion of Brownian Particles with Energy Depots,” *Physical Review Letters*, vol. 80, no. 23, pp. 5044–5047, Jun. 1998.

[30] R. Pastor-Satorras and A. Vespignani, “Epidemic Spreading in Scale-Free Networks,” *Physical Review Letters*, vol. 86, no. 14, pp. 3200–3203, 2001.

[31] J. D. Noh and H. Rieger, “Random Walks on Complex Networks,” *Physical Review Letters*, vol. 92, no. 11, 2004.

[32] A. Arenas, A. Diaz-Guilera, J. Kurths, Y. Moreno, and C. Zhou, “Synchronization in complex networks,” *Physics Reports*, vol. 469, no. 3, pp. 93–153, Dec. 2008.

[33] J. Lorenz, S. Battiston, and F. Schweitzer, “Systemic Risk in a Unifying Framework for Cascading Processes on Networks,” *European Physical Journal B*, vol. 71, no. 4, pp. 441–460, 2011.

[34] R. Cohen, K. Erez, D. Ben-Avraham, and S. Havlin, “Breakdown of the Internet under Intentional Attack,” *Physical Review Letters*, vol. 86, no. 16, pp. 3682–3685, Apr. 2001.

[35] S. V. Buldyrev, R. Parshani, G. Paul, H. E. Stanley, and S. Havlin, “Catastrophic cascade of failures in interdependent networks,” *Nature*, vol. 464, no. 7291, pp. 1025–1028, Apr. 2010.

[36] M. Kivelä, A. Arenas, M. Barthelemy, J. P. Gleeson, Y. Moreno, and M. A. Porter, “Multilayer Networks,” *ArXiv e-prints*, Sep. 2013.

[37] C. D. Brummitt, R. M. DâSouza, and E. A. Leicht, “Suppressing cascades of load in interdependent networks,” *Proceedings of the National Academy of Sciences*, vol. 109, no. 12, pp. E680–E689, 2012.

[38] G. J. Baxter, S. N. Dorogovtsev, A. V. Goltsev, and J. F. F. Mendes, “Avalanche Collapse of Interdependent Networks,” *Phys. Rev. Let.*, vol. 109, no. 24, p. 248701, Dec. 2012.

[39] J. Um, P. Minnhagen, and B. J. Kim, “Synchronization in interdependent networks.” *Chaos*, vol. 21, no. 2, p. 025106, Jun. 2011.

[40] P. Holme and J. Saramäki, “Temporal networks,” *Physics Reports*, vol. 519, no. 3, pp. 97–125, Oct. 2012.

[41] R. Pfitzner, I. Scholtes, A. Garas, and C. J. Tessone, “Betweenness Preference: Quantifying Correlations in the Topological Dynamics of Temporal Networks,” *Phys. Rev. Let.*, vol. 110, no. 19, p. 198701, 2013.

[42] I. Scholtes, N. Wider, R. Pfitzner, A. Garas, C. J. Tessone, and F. Schweitzer, “Slow-Down vs. Speed-Up of Information Diffusion in Non-Markovian Temporal Networks,” *Arxiv preprint*, pp. 1–10, 2013.

[43] O. Sander, A. Klimm, and J. Becker, “Hardware Support for Authentication in Cyber Physical Systems,” *it - Information Technology*, vol. 55, no. 1, pp. 19–25, 2013.

[44] E. A. Lee, “Cyber-Physical Systems - Are Computing Foundations Adequate?” in *Proc. of NSF WS on Cyber-Physical Systems*, 2006. [Online]. Available: http://chess.eecs.berkeley.edu/pubs/329.html

[45] E. Geisberger and M. Broy, Eds., *agendaCPS*. Springer Verlag, 2012, vol. 1.

[46] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY: Springer, 2006.

[47] A. Motro and P. Smets, *Uncertainty Management in Information Systems – From Needs to Solutions*. Springer Verlag, 1997.

[48] G. Blair, N. Bencomo, and R. B. France, “Models@ run.time,” *Computer*, vol. 42, no. 10, pp. 22–27, 2009.

[49] S. J. Ovaska, Ed., *Computationally Intelligent Hybrid Systems: The Fusion of Soft Computing and Hard Computing*. John Wiley & Sons, Chichester, New York, 2004.