Autonomous and Collaborating Cyber-Physical Systems

Prof. dr. Ben van Lier – ben.van.lier@centric.eu
Director of Strategy & Innovation at Centric, Gouda
Professor at Steinbeis University, Berlin
Professor at University of Applied Science Rotterdam

Abstract*—The term "autonomy" comes from the Greek roots "autos" and "nomos", which mean "self" and "law" respectively. This survey paper investigates the concept of autonomy in connection with the development of cyber-physical systems. The concept of autonomy refers to a form of self-regulation or self-government of individual or collaborating systems. An autonomously operating system will need to be able to recognise and differentiate itself among other systems to be able to function on different levels. To be able to increase systems' autonomy, they need self-configuring, self-optimising, self-healing, and self-protecting capabilities. By developing such capabilities, systems will be able to adapt to changes that emerge from within them or in their environment. Networking these systems subsequently allows them to communicate and interact with each other, ultimately enabling them to jointly make decisions. This kind of joint decision making potentially amplifies the autonomy of the whole of interconnected systems in the performance of joint activities.

Keywords*—Autonomy, Self-Adaptive, Communication, Collaboration, Cyber-Physical Systems, Systems of Systems.

I. INTRODUCTION

Heidegger [1] claimed that the development of cybernetics will have major repercussions for science in general and for our thinking about technology in particular. Writing about the new science of cybernetics, he stated that: "No prophecy is necessary to recognize that the sciences now establishing themselves will soon be determined and guided by the new fundamental science which is called cybernetics" [1:376]. According to Rid [2], the science of cybernetics as developed by Wiener actually enables an elegant link-up between "electronic engineering and the life sciences, blurring the line between living and nonliving systems" [2:66]. The experiences of Ashby [3, 4] in the realm of cybernetics in particular show how systems, which in Ashby's case were machines, can be interconnected and thus be able to stabilise themselves, both separately and as a whole, in their interconnected functioning. The stable functioning of this new and interconnected whole comes about through intercommunication and interaction between the interconnected systems. Intercommunication, interaction, and feedback loops all help create a functioning, stable, self-adaptive whole. Horn [5] draws attention to the fact that the development of the Internet and the ensuing connections between humans and between humans and technological systems in particular is something that takes us, as human beings, to new heights of complexity. The Internet uses interconnection, intercommunication, interaction, and the associated feedback loops to form a new synthesis based on new combinations of humankind and technology. This new level of complexity is the result of, among other things, the interconnection of computers as objects in networks, so Horn claims, "to connect – some might say entangle – this world of computers and computing systems with telecommunication networks" [5:4]. In Horn's view, what people generally consider a positive development is that the new whole acquires an ability to self-organise. This positive experience leads Horn to the following observation: "That's why we need a systematic approach that follows for coordination and automatic management across entire networks of computing systems – systems built on various platforms and owned (or even shared) by various entities. Autonomic computing is thus a holistic vision that will enable the whole of computing to deliver much more automation than the sum of individually self-managed parts" [5:11]. Successively, this paper will examine how thinking about autonomous systems has evolved over time. Questions will be asked about which properties systems need to be developed in order to become more autonomous. Subsequently, the way in which autonomous systems can develop in new combinations of hard- and software, also called cyber-physical systems, will be discussed. These new combinations have opportunities for communicating, collaboration and making decisions through the applied software. The conclusions will briefly reflect on new questions these new capacities can evoke.

II. AUTONOMOUS

Autonomic computing, as referred to here by Horn, is based on the concept of autonomy, a word that comes from the Greek *autos* and *nomos*, which mean "self" and "law" respectively. The concept of autonomy thus refers to a form of self-government or self-regulation of individually or jointly operating systems. Horn's vision of a development of a form of autonomous computing does, however, directly lead to the question of what autonomy of computer systems can or should look like and what fundamental features autonomous systems in principle need to have to be able to collaborate in networks. In Horn's theory, autonomous systems should have a basic ability to organise and manage the processes that they need for their functioning. Collaborating autonomous computer systems will, in his

978-1-5386-4444-7/18/$31.00 ©2018 IEEE

view, have differentiating elements at more specific and higher levels. In Horn's words: "To be autonomic, a computing system needs to know itself – and comprise components that also possess a system identity" [5:20]. An autonomously operating system will need to be able to recognise and differentiate itself among other systems to be able to function on different levels. Aside from that, the system will have a continuous need for detailed knowledge from its constituent components, as well as knowledge of the status of these components and therefore of the functioning of the autonomous system as a whole. The autonomous system will, based on the information collected, determine the maximum capacity that is available to the system as a whole to perform its tasks. Besides this capacity, the autonomous system needs to be able to collaborate with other systems and jointly make decisions in mutual consultation. And finally, it is important in Horn's theory that an autonomous system has an independent ability "to know the extent of its owned resources, those it can borrow or lend, and those that can be shared or should be isolated" [5:21]. To be able to use all these capabilities simultaneously, an autonomous system needs to have some kind of awareness of the functioning of the whole. This is because, as Horn points out, "a system can't monitor what it doesn't know exists, or control specific points if its domain of control remains undefined" [5:21]. Through a form of overall self-awareness of its own functioning and that of its surroundings, the whole needs to be able to (re)configure itself amidst changing and unpredictable conditions for the performance of a specifically assigned task. For such a whole to be able to deal with disruptions, what is needed according to Horn is that "adaptive algorithms running on such systems could learn the best configurations to achieve mandated performance levels" [5:22]. In Horn's thinking, such a form of self-learning capability enables the autonomous whole to recover or self-configure, or in Horn's words, "to recover from routine and extraordinary events that might cause some of its parts to malfunction" [5:24]. The creation of learning and self-recovering mechanisms for autonomous systems will enable these systems to either find or develop alternative working methods or determine the mode of (re)configuration to be able to guarantee the operability of the system as a whole. The development and application of such self-recovery capabilities requires, according to Horn, that systems be aware of the environment in which they have to perform their tasks to adequately adapt themselves to this environment. For Horn, the potential set of capabilities such as self-awareness, self-organisation, self-healing, and self-(re)configuration is comparable to the capabilities of organisms within an ecosystem, which he describes as follows: "In nature, all sorts of organisms must coexist and depend upon one another for survival (and such biodiversity actually helps stabilize the ecosystem" [5:27]. Ganek and Corbi [6] follow Horn's reasoning. They go along with Horn's claim that a development is needed where systems acquire even greater autonomy. The need to develop increasingly autonomous computer systems is, according to Ganek and Corbi, prompted by a combination of rapid changes in the scale, scope, and requirements for application in mission-critical conditions. Kephart and Chess [7], too, claim that as the diversity of systems increases, "architects are less able to anticipate and design interactions among components, leaving such issues to be dealt with at runtime" [7:41]. The solution would, in their view, be to create more autonomous systems that are able to operate themselves in fulfilling the tasks that have been assigned to the system. In Kephart and Chess' thinking, the term autonomous system symbolises "a vast and somewhat tangled hierarchy of natural self-governing systems, many of which consist of myriad interacting, self-governing components that in turn comprise large numbers of interacting autonomous, self-governing components at the next level down" [7:41]. Like biological systems, autonomous computer systems will, in their view, "maintain and adjust their operations in the face of changing components, workloads, demands and external conditions and in the face of hardware or software failures, both innocent and malicious" [7:42]. For Parashar and Hariri [8], a computer system that has an autonomous ability to adapt its behaviour to changes in its environment is a homeostatic system. They describe such a system as follows: "Such a system reacts to every change in the environment, or to every random disturbance, through a series of modifications that are equal in size and opposite in direction to those that created the disturbance. The goal of these modifications is to maintain internal balances" [8:248]. A form of self-adaptability to changes emerging from a system's environment is something that Parashar and Hariri feel is necessary to keep the system as a whole stable. They refer back to Ashby [3], who stated that "adaptive behaviour is equivalent to the behaviour of a stable system, the region of the stability being the region of the phase-space in which all the essential variables lie within their normal limits" [3:64]. Like Salehie and Tahvildari [9], Parashar and Hariri identified the same four characteristics as were used by Ganek and Corbi, which they abbreviated as the CHOP properties. These four CHOP properties are self-configuring, self-healing, self-optimising, and self-protecting. They claim that all these properties play a role in adapting to changes that enter the system from the system's environment and are likely to affect the system's behaviour. Salehie and Tahvildari define self-configuring as "the capability of adapting automatically and dynamically to environmental changes" [9:5], and the self-healing property as "the capability of discovering, diagnosing and reacting to disruptions" [9:5]. Self-optimising is defined as "efficiently maximizing resource allocation and utilization for satisfying requirements of different users" [9:5]. And finally, their definition of self-protection as a system property is "the capability of reliably establishing trust, and anticipating, detecting and recovering from the effects of the attacks" [9:5]. According to Agarwal and Harrod [10], the developments outlined here, and the above properties of autonomous systems, will propel the development of more organically functioning computer systems. These systems will fundamentally differ from today's more procedurally oriented computer systems. The difference they identify is caused by the fact that it is impossible in autonomous operation to preconfigure all tasks in a system into possible scenarios, which leads them to conclude that "the organic computer also implements learning and decision making engines in judicious combination of hardware and software to determine the appropriate actions based on given observations" [10]. To Huebscher and McCann [11], it is clear that autonomous execution of tasks by a system of systems is possible only when the joint systems collaborate to achieve a shared objective. This notion of collaboration of individual elements to realise a shared objective is, according to Huebscher and McCann, a fundamental focus point of research into possible forms of collaborating multi-agent systems. They conclude that the development of collaboration between different systems requires a process of mutual alignment and decision making between these systems. This decision-making process can be based on mutually agreed consensus rules about decisions that are to be made jointly. These decisions will, in turn, lead to joint performance of a specific action or transaction, or for the realisation of a shared objective.

III SELF-ADAPTATION

Based on a form of awareness of itself as a whole, a system is able to adapt to changes emerging from within itself or from its surroundings. This ability can be considered the system's adaptive capability. The possibility of independent adaptation or behaviour change by an autonomously operating system in response to changes is what we refer to as self-adaptation. The development of an ability to self-adapt to changes arising from a system's environment is the first and a necessary precondition for the development of autonomous operation of any random system. In the words of the United States Department of Defense (DoD) [12]: "Autonomy is a capability (or a set of capabilities) that enables a particular action of a system to be automatic or, within programmed boundaries, self-governing" [12:1]. Back in 2012, the Department of Defense still worked based on the assumption that all autonomous systems are in one way or another under the responsibility of human operators. Besides human operators, there are algorithms and software that regulate the behaviour of an autonomous or semi-autonomous system. According to Mitchell [13], the term algorithm generally refers to "steps by which an input is transformed to an output" [13:129]. According to the DoD in 2012, autonomous systems still use algorithms and software in which humans have specified boundaries. These boundaries determine how independent the autonomous system is and how able it is to autonomously make decisions or perform actions that have been or will be delegated to the system. In the above description, the autonomy of a system is more than an intrinsic property of an isolated and unmanned system. The autonomy of a system should, according to the DoD, in fact be considered an outcome of a process of collaboration between human and system(s), both in the development and in the execution of tasks or actions by the system or systems. The new combination of object (hardware), rules based on which this object operates (algorithms), and the way in which the object performs its tasks (software) is what determines, in conjunction with humans, the boundaries of autonomy within which the system is able and allowed to operate independently. This new combination of hardware, algorithms, software, and humans controls the execution of tasks and actions by the object or objects. The US DoD added in 2012 that the increasing complexity of interconnected humans, algorithms, software, and hardware creates a great variety of challenges, both in the area of interaction between interconnected systems in dynamic environments and in collaboration between human and system. The greatest challenge in this development is, however, the required shift in focus from system hardware to the algorithms and software that a system needs to be able to function autonomously. In 2016, the DoD [14] observed that the development of system autonomy had up to then produced a result that ensued from the transfer of authority from human to system to enable the system to perform actions independently within predefined boundaries. The restriction imposed by these boundaries basically curtails or even eliminates the system's possibilities of operating outside these boundaries, thus also constituting a restriction of the system's autonomy. To be able to operate with a far-reaching level of autonomy, the US DoD argues that "a system must have the capability to independently compose and select among different courses of action to accomplish goals based on its knowledge and understanding of the world itself and the situation" [14:4]. Further development of system autonomy therefore requires such autonomy to be embedded in an increasing number of algorithms and interconnected software entities. To inspire stakeholder confidence in decisions made by systems individually or jointly, the question of how to shape regulation of this decision-making process must be addressed at an early stage in the design process for these procedures. Getting designers and stakeholders together at an early stage to have them come up with possible conditions that these decisions have to meet will create the possibility to apply "adequate indicator capabilities so that inevitable context-based variations in operational trustworthiness can be assessed and dealt with at run-time" [14:14]. According to Scharre et al. [15], an essential dimension of autonomous systems is thus increasingly created by the level of complexity of the system itself and the environment within which the system has to operate. In Scharre's words: "Complexity matters because it affects the human operator's ability to predict the behavior of the system" [15:11]. The complexity created by interconnections, intercommunication, and interaction within the system and between the system and other objects in its environment will reduce the transparency of the system's functioning, making it harder for human stakeholders to fathom the system's operations. The result of such increasing complexity could, according to Scharre, be that: "predicting the system's behavior, particularly when operating in complex and unstructured real-world environments can be more challenging" [15:11]. In this same context, Laddaga [16] states that the assumption that algorithms and software can give an autonomous system the ability to self-adapt requires that the software, in turn, be able to implement any changes on the fly. Laddaga and

Robertson [17] concluded that this basic premise means that: “we design and code an application as a control system. The runtime software is treated like a factory, with inputs and outputs, and a monitoring and control facility that manages the factory” [17:1]. The algorithms and software that autonomous systems need should therefore, according to Laddaga and Robertson, consist of parts that jointly control and monitor the whole. Self-adaptive software will play a major role in the development of all kinds of embedded software for use in areas such as robotics, manufacturing, aerospace, self-driving cars, and sensor systems. Laddaga and Robertson: “As such, self-adaptive software is an ideal framework for building pervasive computing systems” [17:2]. Salehie and Tahvildari [9] point out that when you take a system of software components that need to be able to regulate themselves and the behaviour of other systems as the starting point, you need interoperability of information between these parts. In their view, “interoperability is always a concern in distributed complex systems for maintaining data and behavior integrity across all constituent elements and subsystems” [9:50]. Due to the fact that, as Brun et al. [18] claim, algorithms and software “become the bricks and mortar of many complex systems (i.e. systems composed of interconnected parts that as a whole exhibits one or more properties (behaviors among the possible properties) not obvious from the properties of the individual parts)” [18:16], algorithms and software have basically become de facto essential factors in the development of self-adaptive systems. According to Brun et al., self-adaptive systems that both operate in a distributed manner and work together differentiate themselves through their self-organising capability. These systems use their self-organising capability to jointly perform activities on a local level while adhering to simple rules, as described by Van Lier [19]. Brun et al. claim the following: “The global behaviour of the system emerges from these local interactions. It is difficult to deduce properties of the global system by analyzing only the local properties of its parts. Such systems do not necessarily use internal representations of global properties or goals; they are often inspired by biological or sociological phenomena” [18:50]. To enable interaction between the systems involved, a feedback loop is a minimum requirement. According to Brun et al., this much-needed feedback loop is made up of at least four activities, namely collecting, analysing, deciding, and acting. The feedback cycle starts with the collection of relevant data from the sensors from the system’s environment. Such sensor data is subsequently enriched with data and information from other sources. The next step sees the system analyse the data and information collected. Based on the outcome of this analysis, the system comes up with proposals to go into the decision-making process. The decision that is ultimately made by the system will be focused on adapting the system to a new and targeted status. Brun et al. claim that such feedback loops will be instrumental in controlling the uncertainty that exists between systems and their environment. Feedback loops not only need to be fit for purpose, they also need to be visible. Visibility of feedback loops will, so Brun et al. argue, make it possible to identify which parts of the feedback loops have important impact on the functioning of the system as a whole. Cheng [20] also sees the feedback loop as a central element in control theory, “which provides well-established mathematical models, tools, and techniques to analyze system performance, stability, sensitivity, or correctness” [20:14]. The increasing interconnectedness of algorithm-based and software-based autonomous systems does, however, lead to an increase in complexity as well, according to Cheng et al. This increasing complexity is, in turn, already leading the software engineering community to invest in new ways for the development, implementation, and management of the ever-evolving, increasingly interconnected landscape of software-intensive systems and services. One of these new ways is described by Baudry and Monperrus [21], who are tying in with the concept of biological ecosystems. In their view, the biological ecosystem makes for a good basis for an approach to the growth and development of these complex and dynamic systems.

IV. COMMUNICATION

As should be clear from the previous, the development of entirely autonomous systems and their mutual collaboration requires a lot more research. New steps in this development are currently already being taken with the development of so-called cyber-physical systems. In this context, Lee [22] points out that integration of physical processes and IT is not a new phenomenon. The existing combinations are, in his view, shaped in the concept of embedded systems. Further development of such embedded systems is possible by connecting them in networks. Such networking, however, also means that the available knowledge about the existing combinations of hardware and software needs a radical rethink. In Lee’s words: “However, the applications we envision demand that embedded systems be feature-rich and networked, so bench testing and encasing becomes inadequate” [22:2]. Poovendran [23] argues that “tomorrow’s CPS must be able to adapt rapidly to anomalies in the environment and embrace the evolution of technologies while still providing critical assertions of performance and other constraints” [23:1365]. Ragunathan [24], in turn, claims that the new combination of cyber-physical systems requires a feature to bridge the gap between the cyber world of computing and communication of these cyber-physical systems and the physical world. He states the following on this: “Cyber-physical systems (CPS) are physical and engineered systems whose operations are monitored, coordinated, controlled and integrated by a computing and communication core. This intimate coupling between cyber and physical will be manifested from the nano-world to large scale wide area systems-of-systems.” [24:1]. The US National Institute of Standards and Technology [25] has defined cyber-physical systems as “smart systems that include engineered interacting networks of physical and computational elements” [25:xiii]. According to Geisberger and Broy [26], cyber-physical systems are “the product of the ongoing development and integrated utilization of two main innovation fields: systems containing embedded software and global data networks like the internet, featuring distributed and interactive application systems" [26:23]. The conclusion is then that the development of cyber-physical systems hinges both on embedded software and on a system's ability to connect to networks in its environment and communicate and interact with other systems in the network based on algorithms and software. Communication between cyber-physical systems consists in the exchange and sharing of data and information in the form of messages between networked systems. If a random cyber-physical system is able to receive, store, and process data and information from its environment, this process enables the system to assign its own meaning to the information received, whereby this significance determines what other activities the system must perform. After all, the meaning assigned supports the system in choosing tasks and how to perform them. The continuing cycle of receiving, processing, assigning meaning, and executing between individual cyber-physical systems can be seen as a process of communication, feedback, and interaction between a diverse range of cyber-physical systems. Networking cyber-physical systems will ultimately lead to the development of a new whole in time and space, a cyber-physical system of systems that operates as a whole and develops based on intercommunication and interaction, thus resembling the processes in a biological ecosystem. Communication is therefore the basis for new interactions between systems and processes of joint decision making between collaborating cyber-physical systems.

V. Collaboration

The process of communication and interaction is also the basis for new possibilities for the adaptation, configuration, or reconfiguration of an individual system or group of systems. Systems can also make decisions between them to self-optimise the functioning of one or multiple interconnected systems. They can also jointly make decisions on the repair of one or multiple faults in one or multiple systems that impede the functioning of the individual or the group. And finally, joint decision making can help protect the functioning of one single system or groups of systems against external attacks. For individual and autonomous systems to be able to make mutual decisions, they need a reliable communication system for the performance of information transactions between separately operating and distributed systems. The communication system as a whole will have to be robust or fault-tolerant, i.e. the systems must always be able to keep functioning, also when constituent systems are not working or not working adequately. When distributed systems in the form of cyber-physical systems perform information transactions in direct partnership with each other, they must reach consensus on the meaning to assign to the information transaction to perform. This jointly assigned meaning, in turn, must lead to a reliable transaction or joint acceptance and processing of the information involved. It must be possible for every autonomous and distributed cyber-physical system to record a jointly performed transaction so that the origins of the information transaction can always be traced without the information having to be available in a central location. Finally, there has to be a protocol in place that specifies all conditionalities for consensus on decisions and distributed recording of these decisions. Lamport [27] defines such a distributed system as "a collection of distinct processes which are spatially separated and which communicate which another by exchanging messages" [27:558]. And he goes on to define the communication process between such systems as a system of events with a predefined order when he says that "we assume that sending a message is an event in a process" [27:559]. Lamport assumes that every system is capable of sending these communication elements directly to other processes, and of receiving similar elements directly from other processes. The ability to send and receive mutually reliable messages between different processes requires distributed algorithms that must ensure that each process follows similar rules for the sending and receiving of messages, meaning that there is no longer a need for centralised synchronisation or storage of these messages. Such a direct form of sending and receiving communication elements between random cyber-physical systems is, according to Lamport, conditional on the active participation of all processes involved in the application of the distributed algorithms that are needed for it. Active participation is possible, Lamport explains, when all processes "know all the commands issued by other processes, so that the failure of a single process will make it impossible for any other process to execute State Machine commands, thereby halting the system" [27:562]. Communication processes' interconnections with and dependency on random and distributed systems means that a system of systems must be able to keep functioning without problems in one or multiple separate systems or components of systems leading to the system of systems malfunctioning or not functioning at all. This means, in Lamport's [28] view, that we have to think about fault-tolerant systems. He considers the concept of a disruption of one or multiple processes within a system meaningless without a notion of time, which leads him to state that "we can only tell that a computer system has failed ("crashed") when we have been waiting to long for a response" [28:96]. Another condition that has to be met to make fault-tolerant systems possible is that "each machine must maintain its own copy of the user machine state" [28:109]. In Lamport's view, communication between systems that function as part of a greater whole can be considered secure when it is impossible, or at least difficult, to disrupt the required communication between the systems through, for example, unauthorised activity or by spreading information that has not been approved beforehand. For a combination of distributed systems to ultimately be able to jointly form a fault-tolerant system, Pease, Shostak and Lamport [29] claim that what is needed is an ability to absorb the effects of faulty functioning or non-functioning of distributed systems by using "voting schemes involving more than one round of information exchange; such schemes might force faulty processors to reveal themselves as faulty or at least to behave consistently enough with respect to the non-faulty processors to allow the latter to reach an exact agreement" [29:228]. Lamport assumes that distributed systems will have to be able to reach consensus unaided on transactions that can lead to, for example, self-adaptation, once distributed algorithms can be developed that can regulate the consistency of these voting schemes. In his opinion, the ability to continuously maintain an interactive form of consistency between separate systems is a fundamental precondition for the design and development of distributed systems, where executive control is also distributed. Lamport [30] describes the procedure to obtain this kind of consistency by using the analogy of the functioning of a parliament in an ancient civilisation, the Paxon parliament. The key requirements behind this algorithm are, firstly, fundamental trust between the entities involved and, secondly, consistency where "each Paxon legislator maintained a ledger in which he recorded the numbered sequence of decrees that were passed" [30:2]. Key conditions for the use of these individual ledgers used by individual systems are described in what is known as the Paxos protocol, including the condition that each decision be recorded using indelible ink so that recorded decisions cannot be changed at a later stage. The Paxos protocol is focused primarily on achieving consistency in recording decisions in the respective distributed ledgers to prevent saving of contradictory information. The Paxos protocol also contains, among other things, rules to ensure that decision-making procedures are initiated and ballots are conducted, rules on quorums for these ballots, and how to reach consensus between separate systems on decisions to be made. Furthermore, the protocol provides rules on the manner in which the decision made is to be recorded in the respective ledgers. Once a decision has been recorded by all involved in their own distributed ledger and can no longer be changed, this decision can be considered to be a shared block that appears in all distributed ledgers. In a group of interconnected cyber-physical systems, consensus would then enable decision-making on joint activities or transactions by random systems. The decisions made are securely recorded in distributed ledgers, which creates new opportunities for learning from previous decisions, while also leading to a higher level of security because the whole no longer depends on central storage of decisions by a trusted third party. Lamport [31] claims that such an approach to the process of decision making based on votes and consensus also offers the possibility of having systems learn from previous decisions. To make this kind of learning happen, a learner node needs to be included in the network that serves specifically to facilitate learning from jointly made decisions, where, according to Lamport, "a learner can learn what value has been chosen" [31:3]. Interconnectedness in networks thus facilitates not only communication and interaction, but also a form of joint decision making about the use of capabilities such as self-adaptation, (re)configuration, self-recovery, optimisation, and self-protection by groups of cyber-physical systems. Autonomy and self-awareness of interconnected cyber-physical systems thus automatically grow as their new capabilities for intercommunication, interaction, and decision making develop.

VI. CONCLUSIONS

Due to their interconnected nature, cyber-physical systems form a new and stand-alone whole, i.e. a synthesis of networked interconnected hardware and software, which is also referred to as a cyber-physical system of systems. The new whole of cyber-physical systems will, as Van Lier [32] argued "continue to evolve as more cyber-physical systems are networked and start communicating and interacting based on algorithms, software, and information" [32:708]. Maier [33] is of the opinion that such a new whole of collaborating cyber-physical systems must be considered to be a system of systems when "its components fulfil valid purposes in their own right and continue to operate to fulfil those purposes if disassembled from the overall system, and the components systems are managed (at least in part) for their own purposes rather than the purposes of the whole" [33:268]. By establishing connections between networked cyber-physical systems, new relationships are formed, according to Boardman [34], between and with other autonomous cyber-physical systems. For Boardman, these new relationships mean that each of these systems "will have to be persuaded of the value of all this - to change, to render service, and to collaborate with other systems" [34:119]. Olfati-Saber et al. [35] point out that, in a network of agents in the form of autonomously operating cyber-physical systems, it is important "to reach an agreement regarding a certain quantity of interest that depends on the state of all agents. A consensus algorithm (or protocol) is an interaction rule that specifies the information exchange between an agent and all of its neighbours on the network" [35:215]. For Jamshidi [36], systems of systems are first and foremost "large-scale integrated systems which are heterogeneous and independently operable on their own, but are networked together for a common goal. The goal, as mentioned before, may be cost, performance, robustness etc" [36:ix]. Dahmann [37] claims that a cyber-physical system of systems is characterised by a joint "set or arrangement of systems that results when independent and useful systems are integrated into a larger system that delivers unique capabilities" [37:2]. Samad and Parisini [38] consider the correlation of decentralised and distributed networked compositions of heterogeneous and (semi-)autonomous elements the defining feature of a system of systems. In their view, the aspect of autonomy within this new whole is key, because "autonomy is inherent in SoS – not just in the function of the SoS but also in the function of the component systems" [38:1]. The freedom of autonomous systems and systems of systems as a whole thus also leads to a new and exceptional challenge in terms of governance and control. Jaradat and Polinpapilinho [39] point out that the behaviour of this new whole cannot be understood by "micromanaging individual systems, autonomy at management and operations levels of individual systems" [39:6]. All of this leads Mens and Grosjean [40] to suggest that the development of interconnected, intercommunicating, and interacting systems such as cyber-physical systems and the intrinsic dynamics of these developing and hardware-based, software-based, and connection-based wholes cannot yet be adequately analysed and therefore not be fully grasped in their development. Maybe as van Lier [41] states it could be helpful to look more into this developing new whole from the perspective of cyber-physical ecosystem a whole that is more than the sum of its constituents parts [41:6].

VII. References

[1] Heidegger M. (1964) The Task of Thinking in: Basic Writings ed. Krell D.F. San Francisco, Harper. English translation 1977. ISBN 0060638451

[2] Rid, T. (2016) Rise of the Machines. The Lost History of Cybernetics. Scribe Publications, ISBN 9781925228649

[3] Ashby, R. W. (1952) Design for a Brain. New York, John Wiley & Sons

[4] Ashby, R. W. (1956) An Introduction to Cybernetics. London Chapman & Hall LTD.

[5] Horn P. (2001). Autonomic Computing: IBM's Perspective on the State of Information Technology. Armonk, NY.

[6] Ganek A.G. and Corbi T.A. (2003) The dawning of the autonomic computing era. IBM Systems Journal 42(1), pp. 5-18

[7] Kephart J.O. and Chess D.M. (2001) The Vision of Autonomic Computing. IEEE Computer Society, pp. 41-50, January 2003

[8] Parashar M and Hariri S. (2005). Autonomic Computing: An overview. In: Autonomic Computing Unconventional Programming Paradigms, F. P. Banâtre JP., Giavitto JL., Michel O. (eds.) Lecture Notes in Computer Science 3566. Springer, Berlin, Heidelberg, pp. 257-269

[9] Salehie M. and Tahvildari L. (2009). Self-Adaptive Software: Landscape and Research Challenges. ACM Transactions on Autonomous and Adaptive Systems (TAAS) 4(2).

[10] Agarwal A. and Harrod D. (2006). Organic Computing, MIT CSAIL and DARPA IPTO, p. 4

[11] Huebscher M.C. and McCann J.A. (2008) A survey of autonomic computing—degrees, models, and applications. ACM Computing Surveys (CSUR), Volume 40, Issue 3, Article No. 7, pp. 1-31

[12] Defence Science Board (2012). Task Force Report: the Role of Autonomy in DoD Systems, US Department of Defense, Office of the Under Secretary of Defence for Acquisition, Technology and Logistics.

[13] Mitchell M. (2009). Complexity: A Guided Tour. New York, Oxford University Press. ISBN 9780199798100

[14] Defence Science Board (2016). Summer Study on Autonomy. US Department of Defense. Office of the Under Secretary of Defense for Acquisition, Technology and Logistics.

[15] Scharre P. (2016) Autonomous Weapons and Operational Risk. Ethical Autonomy project, February 2016

[16] Laddaga R. (1999). Creating Robust Software through Self-Adaptation. IEEE Intelligent Systems and their applications 14(3), pp. 26-29

[17] Laddaga R. and Robertson P. (2004). Self-Adaptive Software: A Position Paper. ACM Transactions on Autonomous and Adaptive Systems (TAAS), Volume 4, Issue 2, Article No. 14

[18] Brun Y. Di Marzo Serugendo G. Gacek Chr. et al. (2009) Engineering Self-Adaptive Systems through Feedback Loops. Cheng B.H.C. Heidelberg Springer-Verlag Berlin, pp. 48-70

[19] Lier, B. van (2015) Advanced Manufacturing and Complexity Science. Ultra-Large-Scale Systems, Emergence and Self-Organisation. 19th International Conference on Systems Theory, Control and Computing (ICSTCC 2014), 14-16 October 2015 https://doi.org/10.1109/ICSTCC.2015.7321307

[20] Cheng B.H.C. Lemos de R. Giese H. Inverardi P. and Magee J. (2009). Software Engineering for Self-Adaptive Systems: A Research Roadmap. Self-Adaptive Systems, Cheng B.H.C. et al. Heidelberg Springer Verlag, pp. 1-26

[21] Baudry B. and Monperrus M. (2012). Towards Ecology Inspired Software Engineering, Arxiv:1205.1102v2 [cs.SE] 10 Jul 2012, Research Centre Rennes - Bretagne Atlantique: 9

[22] Lee E.A. (2006) Cyber-Physical Systems - Are Computing Foundations Adequate? NSF Workshop on Cyber-Physical Systems: Research Motivation, Techniques and Roadmap. Austin TX USA

[23] Poovendran R. (2010). Cyber-Physical Systems: Close Encounters Between Two Parallel Worlds. Proceedings of the IEEE 98(8), pp. 1363-1366

[24] Ragunathan R. Lee I. Sha L. Stankovic J. (2010). Cyber-Physical Systems: the Next Computing Revolution. Design Automation Conference 2010, Anaheim, California USA ACM

[25] US National Institute of Standards and Technology (2016) Framework for Cyber-Physical Systems Release 1.0 May 2016 Cyber-Physical Systems Public Working Group

[26] Geisberger E. and Broy M eds.(2015) Living in a Networked World. Integrated research agenda Cyber-Physical Systems. Acatech study, March 2015

[27] Lamport L. (1978a). Time, Clocks, and the Ordering of Events in a Distributed System. Communications of the ACM 21(7), pp. 558-565

[28] Lamport L. (1978b). The Implementation of Reliable Distributed Multiprocess Systems. Computer Networks 2, pp. 95-114

[29] Pease, Shostak and Lamport (1982), Byzantine Fault Tolerance

[30] Lamport L. (1998). The Part-Time Parliament. ACM Transactions on Computer Science 16(2), pp. 133-169

[31] Lamport L. (2002) Lower Bounds for Asynchronous Consensus. MSR-TR-2004-72

[32] Lier van B. (2017) Can Cyber-Physical Systems Reliably Collaborate within a Blockchain? Metaphilosophy, vol. 48, no. 5, October 2017

[33] Maier M.W. (1998) Architecting Principles for Systems-of-Systems. Systems Engineering, Vol. 1 Issue 4, pp. 267-284

[34] Boardman J. and Sauser B. (2006) System of Systems - the meaning of of. International Conference on Systems of Systems Engineering, Los Angeles, CA, USA, IEEE/SMC

[35] Olfati-Saber R. Fax A.J. and Murray R.M. (2007) Consensus and Cooperation in Networked Multi-Agent Systems. Proceedings of the IEEE, vol. no 1. January 2007, pp. 215-233

[36] Jamshidi M. (2008) Introduction to System of Systems - System of Systems Engineering. Principles and Applications. Ed. Jamshidi M. CRC Press Taylor & Francis group Boca Raton, Florida ISBN 9781420065893

[37] Dahmann J. Baldwin K.J. and Rebovich G. (2009). Systems of Systems and Net-Centric Enterprise Systems, 7th Annual Conference on Systems Engineering Research Loughborough, UK

[38] Samad, T. and Parisini, T. Eds. (2011) Systems of Systems: The Impact of Control Technology. In: The Impact of Control Technology, IEEE

[39] Jaradat M. and Polinpapilinho K. (2011). A Synthesis of Definitions for System of Systems Engineering. American Society for Engineering Management, Lubbock, Texas, USA. in: Proceedings of the 32st National ASEM Conference: Winds of Changes: Staking Paths to Explore New Horizons

[40] Mens T. and Grosjean P. (2015) The Ecology of Software Ecosystems. Computer, October 2015, pp 112-114

[41] Lier van B. (2017) The Industrial Internet of Things and Cybersecurity. IEEE Xplore 2017 21st International Confernce on Systems Theory, Control and Computing. Pp. 641-647

Biography

Professor dr. Ben van Lier is Director of Strategy & Innovation at Centric, a Dutch IT company with offices in Belgium, Norway, Romania, Sweden and Germany. In this capacity, he focuses on research and analysis of developments on the interface between organisations and technology, as well as on future technological developments. In 2013, he was appointed Professor at Steinbeis University Berlin. In this role, he focuses on qualitative research on topics such as systems and complexity theory, interoperability of information and the network-centric approach. In 2015, he was also appointed Professor at Rotterdam University of Applied Sciences. He fulfils both roles alongside his work at Centric.