Report on the AMADEOS Workshop on Emergence in Cyber-Physical Systems-of-Systems (CPSoS)

March 10 – 11, 2016

Editor

Oliver Höftberger, Technical University of Vienna

List of Participants

Andrea Bondavalli	University of Florence
Bernhard Frömel	TU Wien
Erwin Heberle-Bors	University of Vienna
Francesco Brancati	ResilTech
Hermann Kopetz	TU Wien
John Rushby	SRI International
Mariken Everdij	Netherlands Aerospace Centre
Mark Bedau	Reed College
Oliver Höftberger	TU Wien
Radu Grosu	TU Wien
Roozbeh Sangi	RWTH Aachen, L4G Project
Somayeh Malakuti	TU Dresden
Sorin Iacob	Thales NL
Uwe Aßmann	TU Dresden

List of Presentations

Hermann Kopetz	Examples of Emergence in Systems of Systems
Erwin Heberle-Bors	Emergence in Biology
Mark Bedau	A defense of Pluralism about Emergence
Hermann Kopetz	Emergence in Cyber-Physical Systems of Systems
Mariken Everdij	Emergence in Air Transport Operations

Uwe Aßmann	Role-Based Emergence: Modeling Emergence with Roles and Contexts
Mark Bedau	How emergence drives the scientific challenges and opportunities, and the philosophical implications, of CPSoS and BioSoS
Andrea Bondavalli	A view on emergence in SoS and what we may do..... about
John Rushby	On Emergent Misbehavior
Sorin Iacob	Local detection of global effects: Principles and Approaches for Anticipating, Detecting, and Measuring Emergence in CPSoS
Somayeh Malakuti	Component-based Representation of Emergent Behavior in Software
Oliver Höftberger	Detection of Causal Dependencies: Anticipation of possible Emergent Behavior
Roozbeh Sangi	Introduction to Local4Global Project
Hermann Kopetz, All	Conclusions

I. Introduction

The objective of the AMADEOS Workshop on Emergence in Cyber-Physical Systems-of-Systems (CPSoS) was to establish an agreed definition of *emergence in CPSoSs*, to clarify the issues around the occurrence of emergent phenomena in CPSoSs and to arrive at design guidelines for the control of emergent phenomena in CPSoSs.

The ultimate purpose of building System-of-Systems (SoS) is to realize novel services and/or functions that go beyond what can be provided by any of the constituent systems (CSs) in isolation. These novel services are emergent services. The concept of emergence is thus at the core of SoS engineering and needs to be fully investigated (understood and deployed).

This report subsumes the presentations and discussions during the workshop. Most of the content in the document is taken from the slides or transcriptions of the talks of the presenters. However, in many cases statements of several participants have been combined. Therefore, knowledge gained from this report is considered to be an intellectual property of all participants listed above. The remaining document starts with different examples of emergence in computer systems and in biology (Chapter II). Then a definition of emergence is provided in Chapter III, followed by a discussion of guidelines, modeling and detection techniques in Chapter IV. In Chapter V future challenges and questions to be investigated are provided. Finally, a short conclusion is given in Chapter VI.

II. Examples of Emergence

Aristotle already concluded that the gliwhole is greater than the sum of the parts. If things are put together, then something new appears, where the new cannot be traced down to the properties or behavior of the parts. Emergent phenomena are ontologically novel. It does not mean that they are novel to the knowledge of the user, but conceptually novel to the properties and the behavior of the parts. This is in contrast to the opinion of some philosophers and other people researching in the area of emergence, who say that emergence only exists if it is new to the user. These people claim that if

an emergent phenomenon can be explained, it is not emergence anymore. However, this kind of definition is subjective, as for some users the phenomenon is emergent, while for others it is not.

Emergence in Computer Systems

In computer science there are emergent phenomena and effects, which can indeed be explained. In contrast, other emergent phenomena, like the *stock market crash in May 6, 2010*, are still unexplained even after many years of investigation. The example of the stock market shows, that systems are built that are not fully understood, and these systems are even at the center of market economy.

Examples of explained phenomena that have been called emergent in the computing literature are:

- A **deadlock between badly synchronized processes**: This is a simple example of emergence that is fully explained. When this phenomenon (i.e., the computer system stopped) appeared the first time, nobody knew why this was happening. After investigation it was found that two processes are not compatible and block each other. This phenomenon was considered emergent by different scientists. The novel phenomenon is a complete stop of the system that holds forever. The notion of a permanent halt does not exist at the micro-level (i.e., the level of the individual processes). A causal dependency between the totality of the processes that acts on each individual process can be observed. This causality is referred to as *downward causation:* the totality of actions/interactions causes/influences/inflicts a change of behavior of the individual processes. On one side the individual processes are part of the totality of processes, and on the other side the totality of processes constraints/influences the behavior of the individual process itself. This leads to a *causal loop* from the lower level to the level above and from above to below. It manifests in a file-based information transfer from one process to the other process, which goes from the upper level back down to the individual processes.

  A deadlock is not predictable with certainty, but probabilistically it is. In other words, in theory deadlocks are predictable (e.g., by simulations), but in practice it is infeasible. Here predictability means that we are sure whether a deadlock occurs in the next round, or not .In order to predict the deadlock it would be necessary to go to the level of the atoms of the transistors in a computer (considering the temperature, air pressure, etc). One possibility to predict a deadlock is, if a symbolic representation (e.g., an interaction graph) of the system (i.e., the interacting processes) exists, where the feedback loop (e.g., circle in a graph) can be found automatically. However, the model (i.e., the symbolic representation) abstracts from reality and it omits certain relevant conditions. The model can only be used to detect the potential of a deadlock, but not to predict a concrete deadlock. Due to the indeterminism caused by true simultaneity (i.e., at the atom level of the transistors) a deadlock is not predictable in principle, when a truly simultaneous system is assumed, even if everything (i.e., the complete state and behavior) at the macro-level is known. While for good emergence the prediction of potential emergent phenomena is sufficient, for bad emergence this is not the case.

- **Fault tolerant clock (FTC) synchronization**: The new phenomenon is the tolerance of a failure in a clock of the system. This phenomenon is explainable. Downward causation results from the algorithm that is used to build an agreed global notion of time among N clocks. This global notion of time inflicts a change of the state of the local clocks. Each clock is determined by the physics of the oscillator and it influences the global notion of time, which leads to upward causation. The phenomenon is predictable. The FTC is based on the assumption that only one clock is faulty. If this assumption is violated, the emergent property might not exist anymore.

- **Thrashing in alarm monitoring**: An event in a plant (e.g., a broken pipe) causes a stigmergic information flow (i.e., in the physical part of a CPSoS) to several sensors, which leads to

correlated messages at those sensors. This incident is followed by an accumulation of messages at a common communication link to the control center, where the sending of messages is retried and even more messages are produced until the real-time properties of the system are violated. The novel phenomenon is the breakdown of communication, which is explainable. The delay caused by the ensemble of concurrent messages in a link of finite capacity causes the real-time communication to break down, which can be interpreted as downward causation. The phenomenon of thrashing is also predictable.

- **Conway's Game of Life (the glider):** The glider in Conway's Game of Life is called emergent by John Holland [Hol00] (and others). Its rules impose a strict cyclic behavior (i.e., evaluation of the next state of each cell). Downward causation appears, as each cycle determines what the next cycle will look like. However, emergence is just a consequence of observation. When observing at a higher level of abstraction something (i.e., a pattern) can be seen that cannot be observed at the lower level of abstraction. Due to the cyclic evaluation of the cells there is no notion of simultaneity in the Game of Life, and therefore, each step is predictable. If the evaluation of the rules of each cell does not happen in strict cycles, it is not predictable anymore.
- **Stock Market Flash Crash on May 6, 2010:** The complexity of the situation (i.e., huge number of traders, trading algorithms, economic circumstances, news and information exchange, etc.) made it impossible to provide an explanation for this phenomenon, even after years of investigation. As in such complex situations *correlation* is easily confused with *causation*, it might also be impossible to determine the actual root cause of this happening.

  While the concept of causation is highly controversial in modern physics (e.g., quantum mechanics), unidirectional temporal cause-effect relations play a prominent role in our subjective models of the world. Often the word cause is only used for events that are controllable. When looking for a cause, the subsidiary conditions that were necessary are usually neglected.

In almost all examples, which are classified as emergent in the literature, downward causation plays a dominant role.

Emergence in Biological Systems

Emergent phenomena are also observed in biological systems, where the key concepts of emergence are:

- Self-organization, stigmergy
- Feedback
- Interactions, interconnectivity
- Patterns, noise
- Unintended consequences

Four levels of evolution in cosmos are distinguished:

- Physical evolution (emergence of the elements in stars, galactic evolution)
- Chemical evolution (emergence of molecules, 2nd generation star systems)
- Biological evolution (emergence of living organisms)
- Cultural evolution (emergence of cultures and civilizations)

As an example, emergence in chemical evolution is considered. In the world of RNA (Ribonucleic acid), ribozymes are capable of self-replication, mutation and catalysis of other nucleic acids, including DNA and peptides. These emergent capabilities are encoded in the nucleotide sequence. The simplest definition of life requires these three capabilities to exist (self-replication, mutation, catalysis). The RNA world shows properties of life and it existed already before the actual life was

formed on our planet. Chemicals form simple molecules that form structures like RNA, which in turn 'simulate' processes of life. Finally, the ribozyme surrounds itself with a membrane in order to become a cell.

In a more precise definition of life, a cell is the unit of life with

- metabolism,
- signal recognition and signal transduction,
- inheritance including change (mutations), and it is
- capable of growth, development and self-replication.

Biologists look at the whole organisms, not just the cells, DNA or RNA. In an organism, there are structures with different levels of complexity, which form the hierarchy of life:

- A **tissue** is a group of similar cells.
- An **organ** consists of different tissues.
- An **organ system** consists of different organs.
- Organs and organ systems fulfill specific **functions** in the organism.
- An **individual** is an organism with distinctive properties.
- A **population** is a group of individuals, part of a species, living in the same area and exchanging genes.
- Individuals of populations of different species, cohabiting in a certain area, represent an ecological community, a specific **ecosystem**.
- Ecosystems can be divided, according to their vegetation, into **biomes**.
- All the biomes of the Earth together represent the **biosphere**.

Emergence also appears in the world of genetics. Genes are associated with traits and interact during trait formation in specific ways with other genes and the environment. Due to the influence of the environment, genes are not 'producing' traits. The individual phenotype (sum of the observable characteristics or traits) emerges in the interaction between the individual's genotype (sum of the variations of individual genes) and the environment. Additional random variations in the genes enable the evolution of organisms. Genes only store information, but do not act or produce anything. Proteins are encoded by the genes and are the actors in a cell. They interact with other proteins and with the environment. Signals are sent between cells by, for instance, hormones or neuro transmitters. These signals are often chemicals conveying a specific message, or other factors like humidity, temperature or the amount of light in the environment. The receiver of a signal requires an appropriate receptor, as otherwise signal could not convey its message and it would not be a signal anymore. This is similar to sensors in CPSoSs, which are needed to read information of the physical environment. Furthermore, in biology, signals are unidirectional.

The cooperation of cells in an organism leads to the formation of tissues, organs and organ systems. This requires a multitude of signals to be exchanged. Obviously, the more cells an organism has, the more signals are required for coordination.

There is a deterministic relationship between genes and proteins: DNA provides the code that is transcribed by an enzyme into RNA, which is further transcribed into proteins. The enzymes are like processors reading the code and producing output. Regulatory genes are those genes that are directing processes of higher complexity. The hierarchy within an organism (tissue, organ, organ system) is reflected in a hierarchy of transcription factor genes. Homeotic genes define the form of organs, while cataster genes define the number of organs. The type of each individual cell is given by the concentration of morphogens (i.e., chemical signals). Within an organ a specific pattern is formed by the gradient of these morphogens and thresholds that define where one tissue ends and another one starts (cf. French Flag Model).

The term *anagenesis* in evolution describes the transformation of one species into another, while *cladogenesis* refers to branching speciation. There are two essential views on biological evolution:

phyletic gradualism – new species gradually emerge – and *punctuated equilibrium* – organisms do not change for a long time, but then they change very rapidly and strongly. Organisms with new body plans might emerge from the accumulation of silent mutations or from macro-mutations that are mutations of homeotic genes. Evolution led to an increasing complexity of body parts, to body homeostasis in order to increase the independence from the environment (e.g., control the body temperature), to incremental shaping of the biosphere by the organisms. Changes on Earth are not arbitrary, but also not circular, nor gradual, but directional, linear, and unpredictably random. A theory exists that evolution has an inherent upwards tendency. Actually, evolution is a staged process where no feedback occurs (i.e., the actions of an organism do not affect the genes of the organism itself, but it might influence the phenotype of its descendants.

Emergence in Air Traffic Management (ATM)

In Air Traffic Management (ATM) different components (agents) are involved: aircrafts, pilots, monitoring via radar, satellites, air traffic controllers on the ground, environment (weather). Interactions among these agents are stochastic (random), discrete (occurring at certain times), continuous interactions or timed interactions.

Example of Emergence in ATM: ATM regulation. Upward causation by the behavior of different participants in the early days of air traffic. The regulation itself creates downward causation on the agents acting in the field.

The term emergence encompasses many different interesting phenomena that are worth studying. A definition of emergence needs to be sufficiently broad in order to capture all different viewpoints. The goal should be:

- Understanding the process of emergence in complex systems: this is necessary to create new forms of complex and robust systems, while being prepared for disturbances
- Understanding the different types of emergence: necessary if we want to understand and master complex systems in science and engineering

III. Definition of Emergence

Philosophical View on Emergence

Many competing opinions on the definition of emergence led to the '*emergence wars*', where people try to find and defend their one true view of emergence. Different people disagree what the term emergence means. In order to defend their position they want to show that other points of view are wrong. General to the discussion of emergence is that a whole emerges from its parts. Therefore, the key general idea of emergence is:

- The whole depends on its parts, and
- the whole is autonomous/independent from its parts.

The concepts of dependence and autonomy seem to contradict each other, but if one kind of dependence is used with a different kind of autonomy, then these are not necessarily inconsistent. One can mean many different things by dependence and there can be many different meanings for autonomy.

In order to avoid the emergence wars, a *pragmatic pluralism* should be adopted. Pragmatic pluralism says that depending on the dependence and autonomy selected, different types ('colors') of emergence are meant, which are perfectly consistent. The question for a pluralist is, *whether the dependence and autonomy are coherent* (i.e., logically consistent). As long as it is coherent, it is a

valid kind of emergence. But even if a certain kind of emergence is coherent, it might not be useful and no examples exist. Therefore the pragmatist asks *if that kind of emergence is theoretically useful*, if it fits reality and if there are examples.

A *bottom-up whole* means:

- The material of a particular token whole at time t is nothing but the organized combination (i.e., the design) of the materials of its parts at t.
- The state of a whole is nothing but the combination of the states of its parts.
- The cause and effect of the state of a whole is nothing but the combination of the cause and effects of its parts.
- Supervenience is implied by a bottom-up whole.

A *robust pattern* refers to different patterns that are produced by the same rule, where different initial conditions are given. However, all these patterns have the same type, i.e., growth rate, density, maze-pattern, etc. Examples can be found in Conway's Game of Life.

An exemplary distinction of emergence could be:

- **Strong emergence:**
  - Dependence is based on:
    - Matter: the matter of the whole is nothing the matter of the parts.
    - Supervene: different micro states (i.e., states of the parts) might lead to the same macro state (i.e., state of the whole), but different macro states cannot be produced by the same micro state.
  - Autonomy:
    - Undefinable: The properties of the whole cannot be defined in terms of the properties of the parts.
    - Brute (downward) cause: it cannot be explained in principle.
  - Example: conscious mind
  - Because conscious mind might be explained in the future by explaining the interactions of neurons, this concept/model might be incoherent (no brute cause is given, no other valid examples available). There is no evidence that this model fits reality. Therefore, pragmatism would reject this concept (for now).
- **Nominal emergence:**
  - Dependence:
    - Bottom-up whole
  - Autonomy:
    - Inapplicable to parts: there is a property of the whole that is just not defined for the parts.
  - Examples: Glider in Conway's Game of Life (new property: motion, speed, direction), vesicles of lipids (new properties: inside and outside, permeable, self-assemble), traffic jams, cellular automata, fault-tolerant clocks, …
- **Weak emergence:**
  - Dependence:
    - Button-up whole
  - Autonomy:
    - Robust complex cause: the properties of the whole have a complex cause from the properties of the parts.

      A whole's robust patterns of behavior is caused by its parts being in an incompressible causal web (rule b1s1 in Convey's game of life). In order to understand what happens in the future, the causal web has to be crawled ("brute force search"). In contrast, with an arbitrary initial state of a compressible causal web it is possible to predict arbitrarily far in the future what will happen,

what pattern will show up (e.g., rule b8s8 in the game of life). The amount of work is always the same. This is like calculating a fixed-point.

- Examples: fault-tolerant clocks, traffic jams, vesicles, origin of life, cellular automata, …

It is an hypothesis that a causal web is incompressible ('complex') iff it is

- highly parallel (large population),
- highly recurrent (causal loops),
- highly context-dependent (synergy), and
- highly non-linear (summing insufficient).

The epistemic consequences are, that it is impossible to derive the ultimate global state in an incompressible causal web, even if the initial and boundary conditions are completely known, except by brute force search in the causal web. Robust global patterns can be derived from extensive empirical observations. However, in an incompressible causal web some global properties might still be derived without crawling the web.

From the example categories above, nominal emergence is the easiest kind of emergence and applicable to most cases of emergence. An example for nominal emergence, which cannot be categorized as weak emergence, is a circle, where the individual dots are the parts of the circle. The dots do not have a size, but the circle has. The circle also has an inside and an outside.

When looking for a definition of emergence, the most appropriate definition for a specific purpose should be chosen, rather than looking for the one true kind of emergence. For the purpose of CPSoSs, the definition of nominal emergence fits best.

Definition for CPSoSs

An SoS is an integration of a finite number of autonomous constituent systems (CS), e.g., embedded systems, which are independent and operable, and which are networked together for a period of time to achieve a certain SoS goal.

Generally, the SoS goal is not achievable by any of the constituent systems in isolation. However, in some cases there might not be a specific goal of the SoS, but the CSs are put together and interact by chance.

A *CPSoS* consists of CSs that communicate in cyber space via messages on a *cyber channel* (i.e., the *Relied Upon Message Interface* – RUMI) or in the physical space via *stigmergic channels* (i.e., the *Relied Upon Physical Interface* – RUPI). The physical systems interact via actions in the environment. *Information* is a preposition about a state or a process in the world. An information item – referred to as *Itom* – is transferred between the CSs. It consists of the *timed data* (e.g., the bit pattern in a cyber system) and the *explanation* of that data. While the data is carried explicitly in a message, the explanation and the time are often implied by context. In an SoS, the context and the time of the sender can be different from the context and the time of the receiver. If this is the case, then a message that carries data without an explanation can be interpreted differently by the sender and the receiver (e.g., 30°F versus 30°C). A stigmergic information channel is present if one cyber-physical system (CPS) acts on the environment common to many CPSs, changing the state of this physical environment and another CPS observes relevant properties of the changed state at some later point in time with an appropriate sensor. Since stigmergic Itoms are derived from the state of the physical environment (not in cyber space) they are exposed to the full spectrum of *environmental dynamics*.

As an example, ants use pheromones, which they distribute on their routes to coordinate their search for food and nest building activities. The ants react based on the intensity of the pheromones. If a source of food becomes unattractive or because of environmental influences (e.g., rain, wind) the

amount of pheromones on a route is decreased and the ants stop to use that route. Stigmergic channels often (unintentionally) close feedback loops, and thus, are the reason for downward causation. For instance, the information flow among drivers on a busy road is mainly of the stigmergic type.

A *multi-level hierarchy* is a modelling structure that limits the overall complexity of a single model and supports the step-wise integration of a multitude of different models. Each level of a hierarchy possesses its unique set of regularities, either natural laws or imposed rules (in the design of artifacts). The phenomenon of emergence is always associated with levels in a hierarchy. However, in some systems the hierarchic levels cannot be clearly distinguished.

The term *holon*, which was introduced by Koestler, refers to the two-faced character of an entity that is considered a whole at the macro level and an ensemble of parts at the micro level. In a holon and in hierarchies of holons – also referred to as *holarchies* – there are two interfaces. One interface where the parts of the holon interact with each other, and the other interface at which the whole interacts with other wholes. This results in a clean multi-level hierarchy. Viewed from the outside (i.e., the macro level) a holon is a stable whole that can be accessed by an interface across its surface. Viewed from below (i.e., the micro-level) a holon is characterized by a set of confined interacting parts. The concept of a holon is almost similar to button-up wholes (see section above), except that buttom-up wholes do not require the interacting parts to be confined. However, if there is no confinement at the lower level, no reasonable abstraction at the higher level can be achieved. The explicit interface at the higher level is required in order to allow the interaction of the lower levels of different holons. For instance, at the cell level the receptors are needed in order to enable molecules at the lower level to interact with the molecules of other cells.

A multi-level hierarchy is a recursive structure where a system – the whole (i.e., the holon) – at the level of interest (i.e., the macro-level), can be taken apart at the level below (i.e., the micro-level), into a set of sub-systems (i.e., the parts) and a design that controls the interactions of the parts. Each one of these sub-systems can be viewed as a system of its own when the focus of observation is shifted from the level above to the level below. This recursive decomposition ends when the internal structure of a sub-system is of no further interest. In the model of a multi-level hierarchy there are two essential relations:

- *Level relations*: How is the upper level related to the lower level? This corresponds to the dependence of the whole from its parts.
- *Interaction relation*: How do the entities at a particular level interact? This corresponds to the autonomy of the whole.

Several different non-exclusive level relations exist. For instance:

- Containment: The whole contains or consists of the parts and the design of the interactions, forming a nested hierarchy. Example: Hierarchy of atoms, molecules, cells, etc.
- Relatedness: The parts of the micro-level are related closer to each other than the wholes at the macro-level. Example: different representations of an Itom and the Itom itself.
- Control: The whole constrains or (partially) controls the behavior of the parts. Example: Blinking of fireflies.
- Description: The parts and the design can be described at different levels of abstraction. Example: Conway's Game of Life.

For SoS design the control relation is the most relevant one. In a control hierarchy, in order to support the simplification at the macro-level and establish an hierarchical control level, a control hierarchy must

- on the one side constrain some degrees of freedom of the behavior of the parts, but
- on the other side it must abstract from, i.e. allow some degrees of freedom of behavior to the parts at the micro-level.

Control originates from two different sources:

- Authority from the outside of the holon: For instance, the authority of a general over the soldiers in a military hierarchy.
- Authority form the inside of the holon: The ensemble of parts at the macro level exercises control over the individual parts at the micro level. This implies that the higher level is equipped with causal powers of its own so that it can inflict effects on the lower level that is causing it.

From the point of view of emergence, authority from the inside is most relevant.

The following two interaction relations are distinguished:

- *Physical Interactions*: come about by force fields, (e.g., electromagnetic or gravitational fields). They are synchronic. Physical structures (e.g., a molecule) are formed by force fields according to physical laws. These interactions are characterized by the distance among the parts, force fields among the parts, relaxation time or frequency of interactions among the parts. When we move up the levels of a material hierarchy the distances increase, the force decreases and the frequency of interactions decreases.
- *Informational Interactions*: come about by the designed exchange of Itoms, either across message channels or stigmergic channels. They are *diachronic*. These interactions happen either direct via state or event messages, or indirect via file-based (in computer systems) or stigmergic (in the physical world) communication.

Emergent behavior in SoSs is caused by informational interactions according to an algorithm. The algorithm that controls the informational interactions is part of the system design. Therefore, the essence for the occurrence of emergent phenomena at the macro-level lies in the organization of the parts, i.e., in the static or dynamic relation among the parts caused by physical or informational interactions among the parts at the micro-level.

Depending on the type of interactions, the following two kinds of emergence can be distinguished:

- *Synchronic emergence:* These are characterized by static interactions, like in crystals leading to the property of hardness or brilliance of a diamond, caused by the interactions of the atoms.
- *Diachronic* emergence: means, that emergence evolves over time – it does not happen at an instance. For example, a traffic jam is a diachronic emergent phenomenon.

***Definition of CPSoS emergence:* A phenomenon of a whole at the macro-level is emergent if and only if it is of a new kind with respect to the non-relational phenomena of any of its proper parts at the micro-level.**

Conceptual/ontological novelty at the macro-level relative to the world of concepts at the micro-level is the landmark of this definition of emergence. The autonomy in this definition is given by the properties of the whole that the part do not have. The novel phenomena can be structures, behavior or properties. In SoSs we are primarily interested in emergent behavior and emergent properties that are defined based on the behavior of the system (e.g., safety is an invariant of the behavior of the system – i.e., the behavior of the system must be such that nothing bad happens), which are associated predominantly with control hierarchies.

The proper conceptualization of emergent phenomena can lead to an abrupt simplification at the next higher level. Therefore, novel concepts must be formed and new laws may have to be introduced at the macro-level to be able to describe the emerging phenomena at the macro-level appropriately. For instance, the concepts of liquidity or the hydrodynamic laws. Since the concepts at the macro-level are new with respect to existing concepts that describe the properties of the parts, the established laws that determine the behavior of the parts at the micro-level will probably not embrace the new concepts

of the macro-level. However, it may be possible to formulate inter-ordinal laws (also called *bridge laws*) to relate the new concepts of the macro-level to the established concepts at the micro-level. In contrast to the view of a number of philosophers, a novel phenomenon at the macro-level is still considered emergent, even if it can be explained by the state of knowledge about the properties and laws that govern the parts at the micro-level. Therefore, the state of knowledge of one person does not impact the classification of a phenomenon as emergent. In a weak emergent system one can even predict what will happen if a simulation exists that is faster than reality.

The Stanford Encyclopaedia on Philosophy states about 'Scientific Reduction': *The term 'reduction' as used in philosophy expresses the idea that if an entity x reduces to an entity y then y is in a sense prior to x, is more basic than x, is such that x fully depends upon it or is constituted by it. Saying that x reduces to y typically implies that x is nothing more than y or nothing over and above y*.

In an artifact, such as an SoS, emergent properties appear at the macro-level if the parts at the micro-level interact according to a design provided by a human designer – this is more than the parts considered in isolation.

Hempel and Oppenheim outlined the following schema for a scientific explanation of a phenomenon: *Given statements of the antecedent conditions and general laws then a logical deduction of the description of the empirical phenomenon to be explained is entailed.* A weaker form of explanation is provided if the general laws in the above schema are replaced by established rules. There are fundamental differences between *general laws* and *established rules*:

- General laws are eternal, inexorable and universally valid, while established rules are context dependent and local.
- Rules about the behavior of things are based on more or less meticulous experimental observations in a limited context.

A special case is the introduction of imposed rules, e.g., the rules of an artificial game, such as chess.

In the field of modern physics, such as quantum mechanics, the meaning of the concept of *causation* is highly controversial. However unidirectional temporal cause-effect relations play a prominent role in our subjective models of the world, and particularly in the design of CPSoSs. *Downward causation* means, that the interaction of the parts at the micro-level cause the whole at the macro-level, while the whole at the macro-level can constrain the behavior of the parts at the micro-level. This results in a *causal loop*.

In a multi-level hierarchy, emergent phenomena often originate from causal loops, which are formed between the micro-level that forms the whole at the macro-level and this whole (i.e., the ensemble of parts) that constrains the behavior of the parts at the micro-level. However, also with linear cause and effect relations (with no feedback), like cascading effects (e.g., nuclear chain reactions, epidemic distribution of viruses), phenomena can be observed that are called emergent.

In a holon there is upward causation by natural laws or from imposed laws and downward causation by the ensemble of parts or from an outside authority (the canon/the algorithm). Within the limits of upward and downward causation, the parts might exhibit free behavior.

Supervenience is a relation between the emergent phenomena of adjacent levels in a hierarchy:

- Sup_1 (Autonomy): A given emerging phenomenon at the macro-level can emerge out of many different arrangements or interactions of the parts at the micro-level.
- Sup_2 (Dependence): A difference in the emerging phenomena at the macro-level requires a difference in the arrangements or the interactions of the parts at the micro-level.

Because of Sup_1 one can abstract from many different arrangements or interactions of the parts at the micro-level that lead to the same emerging phenomena at the macro level. The proper conceptualization of the new phenomena at the macro-level is at the core of the simplifying power of

a multi-level hierarchy with emergent phenomena. Whenever the observed emergent behavior at the macro level deviates from the intended behavior, there must be determinants at the micro-level (e.g., the cause of the observed failure – refer to Sup_2), which enables fault diagnosis.

However, even if a difference in the macro-level requires a difference in the micro-level, there need not be a causal relation. For instance, in case of synchronic interactions, where no time is involved, no cause-effect relation exists. As an example, a circle can be considered, where a segment of the circle is removed. Then this circle does not have the emergent properties inside/outside or diameter. The reason why these properties do not emerge at the macro-level is, that a group of dots (a segment of the circle) at the micro-level is missing. However, this incident is not causal. Therefore, supervenience is not always limited to causal contexts. In CPSoSs we want to limit supervenience to causal contexts.

In a CPSoS different types of causal interactions are observed that might lead to emergent behavior:

- *Causal chains*: Finite sequence of dependencies between states of different constituent systems (CS) of different type. The interaction with other CSs does not influence the CS itself.
- *Causal pseudo-loops*: Finite sequence of dependencies between states of different CSs, where the sequence of CS types forms a periodic pattern. The interaction with other CSs does not influence the CS itself, but CSs of the same type. For instance, a car in a traffic jam does not prevent itself from moving, but it prevents other cars to move forward.
- *Causal loops*: Infinite sequence of circular dependencies between states of different CSs. A CS's interaction with other CSs also impacts later inputs for the CS itself.

Balancing between the upward causation and downward causation (control of properties) is called adaptation.

Jochen Fromm introduced in [Fro05] the following taxonomy of emergence based on different types of feedback:

- **Nominal:** Only upward causation; properties at macro-level are different from properties at micro-level.
    - *Simple Intentional*: by design. Example: Function of simple machines (e.g., steam engine, mechanical clocks). Emergence: The function of the whole is not equal to the function of the components.
    - *Simple Unintentional*: just happening. Consist mostly of homogenous (equal) components, like grains or snowflakes. Example: avalanches, pressure of gases.
- **Weak:** simple form of downward causation
    - *Weak stable:* negative feedback (stabilizing feedback). Examples: flocks of birds or fish, termites' nests, Wikipedia.
    - *Weak unstable:* positive feedback (amplifying feedback). Example: people copy the behavior of other people without a common goal. For instance, ghettos are formed if richer people move away from a district and poorer people move in. This causes other rich people to leave that district as well. Also certain forms of economic crashes are caused by this type of emergence.
- **Multiple:** multiple feedback loops, learning, adaptation
    - *Multiple feedback:* combination of short term unstable feedback (mutations) and longer term stable feedback. Example: Patterns in stripes of zebras or tigers, the game of life (different structures move around).
    - *Adaptive multiple:* by sudden changes in existing complex structures certain barriers are overcome that enable a new kind of emergence. Example: the extinction of the dinosaurs enabled other types of life to occur.
- **Strong:** Strong and supervenience, Multi-level emergence with huge amount of variety. Many layers with a combinatorial explosion of states, such that macro properties cannot be predicted

or explained from the micro states. (No fundamental limitation in understanding, but too many involved layers in the hierarchy). Example: life, culture.

A possible classification of emergence in CPSoSs is based on explanation. Classification by explanation is relative to the state of knowledge (or the state of ignorance) – it may change as knowledge is increased:

- No explanation as of today – mind over neurons
- Explanation in principle – stock market crash
- Explanation by simulation – air traffic example
- Explanation by logic reasoning – clock synchronization

Another classification of emergence could be by prediction. Prediction is in some sense orthogonal to explanation. Classification by prediction is relative to the state of knowledge (or the state of ignorance) – it may change as knowledge is increased:

- Not predictable in principle
- Stochastic predictability (e.g., by simulation)
- Logic predictability

Therefore, the starting point and goal of prediction must be clearly stated.

The term *misbehavior* originates from a paper of J. Mogul [Mog06]. It does not only refer to failures of complex systems that cannot be predicted from their components, interactions, or design, but also to undesired behavior in general.

In CPSoSs there are three new sources of failure:

- Downward Causation: Suppose we have a function and property that depend on clock monotonicity (the clock always goes forward). When this clock is included in a synchronized system, the clocks can go backwards as well. This might lead to the property failing. Therefore, we must recognize that mechanisms of emergence may not preserve prior properties.
- Stigmergic channels: Components may communicate through unanticipated channels and may then get spontaneous and unexpected emergent behavior (e.g., router synchronization, traffic waves). Partitioning is used to eliminate unintended data channels. But stigmergic channels are more difficult. Conjecture: We may need to consciously design emergence on the stigmergic channels to avoid spontaneous unintended ones.
- Ontological novelty: Missed opportunities (not a failure, but less than optimal behavior): For instance, when a pace maker with 60 beats per minute and a lung machine with 20 beats per minute are connected to a patient. Then the patient establishes a relationship between these systems, leading to possible emergent effects. It is known that recovery is better if both systems are harmonic (i.e., synchronized). The systems should already foresee a possible interaction with other systems (e.g., provide an interface), which requires ontological knowledge about the other systems (e.g., the concept of heart beat).

We need to be able to anticipate, detect, and measure not only emergence, but also global properties in general. There is a distinction between *intrinsic properties* and *extrinsic properties*. Intrinsic properties describe properties of a CS and can reliably be measured locally within a short time interval, without the need for external references. Extrinsic properties, in contrast, describe properties of the SoS. They need an external reference/perspective to be measured. It is conjectured that emergent properties are extrinsic, so no combination of intrinsic quantities can describe an emergent

property.
Other global properties are either aggregations or abstractions of intrinsic properties and also important for the proper design and management of SoSs.

Local causes of global effects can be categorized as follows:

- Strongly emergent effects cannot be traced down to specific micro-level causes. Usually this may not happen in CPSoSs.
- Deterministic predictable global effects (e.g., oscillations, linear diffusion) appear in linear non-dissipative systems. They are not really interesting from the perspective of emergence, but useful for SoS management and reaction strategies.
- Deterministic unpredictable global effects (e.g., aperiodic oscillations, bifurcations, phase transitions) are caused by
    - Nonlinear response of CS (far from equilibrium states).
    - Nonlinear composition of CS interactions (e.g. double pendulum, logistic map).
    - Nonlinear or non-isotropic propagation of information.
    - Causal loops (positive feedback), reinforcing causal chains.
    - Dissipative boundary conditions.
- Non-deterministic predictable global effects (e.g., self-organized criticality) are caused by noisy observations, or stochastic state transitions, but they are predictable without exact knowledge of local behavior.
- Non-deterministic unpredictable global effects seem to be close to strong emergence. No examples known.

IV. Guidelines, Modeling and Detection Techniques

Detection Techniques in Air Traffic Management (ATM)

In ATM situation models are used to simulate interactions between agents. These models are stochastic and involve continuous, discrete and timed processes. The simulations are used to model the properties of the whole system after it is changed.

Monte Carlo simulation (with petri nets) are used to evaluate the safety of active runway crossing operations on an airport. Several millions of simulations are required for this safety evaluation. The simulations are conducted to identify event sequences that are relevant for safety and to omit other ones (in further safety assessments) for cost reasons.

Modeling Emergence with Roles and Contexts

As the limits of a language set the limits of the world (that can be modelled or described), dedicated language concepts are needed to describe emergence. One possibility is to use a role-based modeling language. Thereby, *roles* partition *objects* and their types for separation of concerns. Roles allow to compartmentalize the state of an object. Objects themselves consist of core attributes that do not change (e.g., name of a person), and of attributes that might change over time (e.g., affiliation). In role modelling, one distinguishes between core attributes, which are intrinsic and which the object never loses, and roles, which are attributes that can be lost over the lifetime of the object. Roles belong to a context. They change if context changes. For instance, the role that a person is employed by a certain company is only valid as long as the context of employment (consisting of the person as employee and the company as employer) exists.

Roles and contexts should be first class language concepts. This allows to program with objects, roles and contexts. There is a difference in the memory allocation of these concepts because of the differences in their life-time. Furthermore, roles can improve knowledge about the independence of states.

A context encapsulates a set of roles. Contexts can be composed (parallel, refined, nested) to obtain more complex contexts with more roles. Contexts may also have contracts. A parallel composition means, that an object plays a role in more than one context (e.g., a person can be employee and customer of a certain company at the same time). New context can be formed by composing existing contexts (e.g., a context 'business' and a context 'employment' can be composed to a new context 'professional life' that contains the previous two). Refinement of context means to add more information to the refined context (e.g., the context 'business' can be refined by a context 'history' that provides information about the business behavior of a customer over time). By nesting contexts, a context is augmented with new roles (e.g., adding supply chain information to the context of a business).When two systems are composed, the context of one system can be superimposed by the context of another system, such that the original context is removed and replaced by the context of the second system. This allows for change which has not been foreseen when the system was designed.

Role- and context-based emergence can be modeled by superimposition, when two systems are put together to obtain an SoS, the context of one system can be superimposed by the context of the other system. The compound of an SoS forms an outer context for all constituents and their inner contexts. Therefore, an SoS forms a hierarchy of contexts. An SoS may superimpose new contexts to all inner contexts of the constituents, i.e., replace inner contexts of the constituents. A constituent system of an SoS may superimpose new contexts to other constituents of the SoS, i.e., replace inner contexts of the other constituents.

Engineering of Emergent Properties in Synthetic Biology

The goal of synthetic biology is to engineer complex systems with desired weak emergent properties. Actually synthetic biologists learn how to control these properties rather than getting rid of them. While exact emergent behavior of complex mechanisms is unpredictable, typical emergent behavior can be approximately predicted given enough empirical observation ("experience") – trial and error.

The process of *directed evolution* is an artificial selection in a test tube. Starting with an initial population with random variations (e.g., of genetic molecules), a test is performed that measures the success of each variant. In the next step, those variants which have been selected are varied (i.e., making copies with errors built in). Again the best ones of the new population are selected. This process repeats until the desired emergent properties are obtained. This only works for certain kinds of things that can be expressed with genetic molecules. But this technique does not fit for CPSoSs.

How to engineer complex systems with desired emergent properties or behavior with the predictive design technology (PDT):

1) Sample the experimental space (start randomly)
2) Create a model of the experimental space (e.g., by using methods from statistics)
3) Predict optimal next experiments with which the most information about the experimental space can be obtained.

An example use case for this technique is drug formulation, where the effectiveness of drugs is increased by experiments with different combinations.

Component-based Representation of Emergent Behavior

In some systems it might be important to represent emergent behavior as an entity, because the system has to work with that emergent behavior (e.g., it contains some state information) or perform operations on that. Therefore the question arises, how can emergent behavior (e.g., traffic congestion) be represented in software?

Design and implementation of software imply design and implementation of emergent behavior. Modularization of software behavior implies modularizing simple intentional/nominal emergent behavior.

The following characteristics of emergent behavior have been identified:

- Emergent behavior may have a transient nature: It appears when certain conditions hold, and disappears when the conditions no longer hold.
- Emergent behavior may have a crosscutting nature: It arises from the interactions among multiple entities.
- Emergent behavior may have an elastic nature: The multiplicity of its constituents may change over the time.

Event-based modularization can be employed to model emergent behavior in software by using:

- First-class abstraction: event module / gummy module
- Sub-abstraction: event producer / consumer
- Interaction: Required / published events
- Composition: Inheritance / Aggregation

Event-based algorithms are adopted in the multi-agents community to detect emergent behavior. Emergent behavior is represented as a complex event that may appear as a result of causal relations of simpler events. In this sense an event denotes a change in a state. A state change that leads to an emergent phenomenon is called an emergent event. There is a special kind of module that observes the behavior of the constituents in the system and concludes whether an emergent behavior in the system has appeared or not. In case an emergent behavior appeared, the module is activated and the system is able to react on the emergent phenomenon, otherwise the module is deactivated.

An existing system can be extended with event-interfaces generating emergent events in order to indicate an onset of emergent phenomena. This is like a wrapper around the existing system with required and provided interfaces. This representation is independent from the actual implementation of the system. Finally, only event-based interfaces are used to communicate between the constituents of the system.

This technique only allows the online detection of anticipated emergent behavior.

Guidelines for CPSoS Design

With proper observation and documentation of interactions among the CSs the occurrence of the emergent phenomena in SoSs can be explained. Still as SoS designers and/or users we would like to gain a complete awareness of emergent phenomena and be able to control (or mitigate the effects of) the detrimental ones.

In SoSs countless causes of unexpected emergence exist, but in many cases unexpected emergence is driven by *ignorance* about:

- The complete set of requirements that need to be addressed in the SoS.
- The complete set of behaviors of each CPS.

- The complete set of interactions among the CPSs.
- The impact of the environment.

System designers/engineers are ignorant about the full set of system behaviors even at the micro-level (CS level). Moreover, things exacerbate when scaling to SoS.

There is also ignorance about the behavior of components:

- Software/hardware systems operate in an unexpected way. For instance, caused by wrong design or implementation or inappropriate use and inclusion in a system (Software wrong reuse).
- Problem of COTS and legacy CPSs used in CPSoS design, as they may come with unknown development faults, they may contain unknown faults (bugs, vulnerabilities, etc.), or their specifications may be incomplete or even incorrect.

In order to contrast detrimental emergence (and boost the good one), we need to reduce the ignorance, eliminate unexpected behavior and interactions. In the past people worked on tracing ignorance as part of requirements engineering by qualitatively quantifying it (e.g., low, medium, high), and tried to understand how this ignorance propagates through simple designs (AND, OR). As a consequence, for the critical parts possibly hiding unexpected emergent behaviors, specific corrective actions could be triggered:

- Replacement of the components identified as potential sources of unwanted emergence.
- Adoption of different architectures, with different correlations between components.
- Cleaning of the environment context, removing possible sources of unknowns.

The ignorance analysis can be profitably used to guide the selection of the system components among several alternatives, e.g., for COTS selection.

When a legacy system or another CS is integrated into an SoS, different assumptions about the behavior of that integrated system are made. A monitor system can be added to the SoS that continuously observes the real system and checks if these assumptions are violated. In case of a violation an alarm is triggered to indicate that an anomaly could occur. Wrapping is another method to constrain some of the behavior of a legacy system and to allow only intended behavior. However, when the system diverges from the expected behavior, a reaction strategy has to exist.

Black and Koopman [Bla(09] observe that safety goals are often emergent to the system components, e.g., the concept (no) ‘collision’ might feature in the top-level safety goal for an autonomous automobile. But ‘collision’ has no meaning for the brake, steering, and acceleration components. They suggest identifying local goals for each component whose conjunction is equivalent to the system safety goal, recognizing that some unknown additional element X may be needed (because of emergence) to complete the equivalence. An objective is then to minimize X.

Reductionist approaches to system design and understanding may no longer be appropriate as systems are built from incompletely understood components or other systems. System goals are far removed from component functions. Widespread emergent misbehavior seems inevitable. In some cases, one can attempt to reduce emergence and restore validity of reductionism. In other cases, one should embrace emergence and aim for adaptation and resilience.

Eliminate unanticipated behaviors and interactions: Behaviors and interactions due to superfluous functionality, e.g., use of a COTS component where only a subset of its capabilities is required, or functions with many options where only some are required. These can be eliminated by wrapping or partial evaluation. Interactions that use unintended pathways (e.g., A writes into B's memory, or interferes with its bus transmissions, or monopolizes the CPU). Strong partitioning of resources can eliminate these hazards. But we remain vulnerable to pathways through the plant.

Control unanticipated behaviors and interactions: Unanticipated behaviors on intended interaction pathways, e.g., unclean failures or local malfunctions. These can be controlled by strong monitoring. For instance, monitor component behavior against system requirements and shutdown on failure, monitor assumptions and treat source component (or self?) as failed when assumptions are violated, use interface automata to monitor interactions, use inline reference monitors (IRMs) to monitor security.

Engineer for Resilience: Diagnosis here is similar to Perrow's Normal Accidents [Per11]. In his terms, we aim to reduce interactive complexity and tight coupling. One way to do both may be to increase the autonomy of components, i.e., they function as goal-directed agents, e.g., substitute runtime synthesis for design-time analysis (both use formal methods, but in different ways). But then it may be more difficult to design the overall system. Furthermore, this opens the door again to emergent misbehavior. Actions of intelligent components frustrate system goals.

Linking intrinsic to extrinsic quantities in CPSoSs can be done:

- Hierarchical
  - Pro: increasing abstraction and semantic levels
  - Con: abstraction removes potentially relevant details, requiring domain models
  - Approaches: Bayesian networks, nonlinear (non-Gaussian) scale-spaces
- Networked
  - Pro: no model required
  - Con: training required (how to cope with unexpected behaviors?)
  - Approaches: unsupervised machine learning

Causal loops/chains are often enabled by stigmergic channels in the environment. The transformation of information in the environment or the mutual interdependency between states in the environment should not be ignored during the development of a CPSoS. Stigmergic channels are not only relevant where CSs comprise dedicated sensors and actuators to interact with the environment, but the environment might also influence CSs via 'implicit sensors'. E.g., the temperature in the environment might influence the temporal behavior of communication channels in cyber space or it might accelerate/decelerate crystal oscillators.

A network of dependencies can model the interactions between CSs, as well as with the environment at an abstract level. It then enables the automatic detection of causal (pseudo-) loops by performing a graph search. Such loops indicate the possibility of emergent behavior. Then the system engineers can further investigate those loops in order to decide whether any emergent phenomenon caused by this loop should be prevented.

In order to prevent such unintended emergent effects, system engineers can break up unintended causal loops (e.g., prevent transport of energy or information between CSs), constrain or control the environmental states (e.g., keep the temperature constant), change the design or control algorithm (e.g., add conditions to control algorithm).

V. Challenges in the Context of Emergence

Questions about the environment (interactions with environment):

- What is the impact of the environment on CSs?
- Should we have to consider the environment itself as a particular kind of CS interacting with the other CS through stigmergic channels?

- How to define and understand, constrain and predict the environmental behavior?
- Can emergent behavior be triggered by the environment and what we can do about it?

CPSoS behavior cannot be fully predicted, because of

- Unknown scope of the CPSoS
- Incomplete specification of CSs (e.g., the number of specified states is smaller than the actual possible states).
- No top-down design of the SoS. The SoS results from interacting CS.
- In many SoSs there is no global coordination. Only local collaborations and competitions exist. Often there are regional hierarchies and networks.

Investigation of the following questions about local and global system properties might help for developing techniques for detection and prediction of emergent phenomena:

- Is it possible to predict global extrinsic properties from a local perspectives?
- Which global effects can be identified based on locally measured quantities?
- Which local quantities are related to global properties?
- Which local behaviors generate global effects?
- What can be measured? Are there quantifiable relationships between intrinsic and extrinsic quantities?
- How local interactions generate global behaviors? Are there stable relationships between intrinsic and extrinsic patterns?
- Are there fundamental limitations of linking intrinsic properties to extrinsic ones?
- Which intrinsic quantities are useful for anticipating emergent properties? (functional, non-functional)
- Which local system behaviors are likely to lead to unanticipated global effects (including emergence)? (non-linearity, unpredictability, indeterminism)
- What global properties result from ‘local’ (CS-level) cyber-physical interactions?
- How to integrate local measurements? Hierarchical vs. networked
- Is it generally possible to discover stable relationships between intrinsic and extrinsic quantities? Even if unexplained.

VI. Conclusion

People in diverse fields of science consider the concept of emergence differently, which makes it almost impossible to come up with a single definition of emergence that would fit the requirements and desires of all domains and that is accepted by all parties. Furthermore, there is an analytic viewpoint on emergence from the natural science and the prescriptive viewpoint from system engineering. As a consequence, it is more constructive to define emergence such that it best fits the domain of interest. However, it is common to all definitions that the whole is more than its parts.

In the field of CPSoSs engineers need to understand emergent behavior to properly design the system and to avoid unintended emergent effects. It is mostly accepted that strong emergence cannot appear in CPSoSs. Nevertheless, many emergent phenomena in those systems still cannot be explained and/or predicted. Sometimes it is possible to explain the behavior, while it is impossible to predict it. In other cases, it is the other way round. The impossibility of explanation and prediction is often based on the ignorance of different aspects during the design of the system. Especially modeling of stigmergic channels and the environment – including environmental dynamics – need much more consideration, as these have been neglected in the past.

References

[Bla09] Jennifer Black and Philip Koopman, *System safety as an emergent property in composite systems*. In *Proceedings of the IEEE/IFIP International Conference on Dependable Systems & Networks 2009*, pages 369 – 378, Lisbon, 2009.

[Fro05] Jochen Fromm, *Types and Forms of Emergence*, http://arxiv.org/abs/nlin.AO/0506028, 2005.

[Hol00] John H. Holland, *Emergence: From Chaos to Order*. Oxford University Press, 2000, ISBN: 0-19-286211-1.

[Mog06] Jeffrey C. Mogul, *Emergent (mis)behavior vs. complex software systems*. In *Proceedings of the 1st ACM SIGOPS/EuroSys European Conference on Computer Systems 2006*, ACM, pages 293 – 304, Leuven, Belgium, 2006.

[Per11] Charles Perrow, *Normal Accidents: Living with High Risk Technologies*. Princeton University Press. 2011, ISBN: 0-691-00412-9.