

# Emergence in Cyber-Physical Systems-of-Systems (CPSoSs)

Hermann Kopetz<sup>1</sup>, Andrea Bondavalli<sup>2</sup>, Francesco Brancati<sup>3</sup>,  
Bernhard Frömel<sup>1</sup>(✉), Oliver Höftberger<sup>1</sup>, and Sorin Iacob<sup>4</sup>

<sup>1</sup> Institute of Computer Engineering,  
Vienna University of Technology, Vienna, Austria  
h.kopetz@gmail.com,  
{froemel,oliver}@vmars.tuwien.ac.at

<sup>2</sup> Department of Mathematics and Informatics, University of Florence,  
Florence, Italy  
andrea.bondavalli@unifi.it

<sup>3</sup> Resiltech SRL, Pisa, Italy  
francesco.brancati@resiltech.com

<sup>4</sup> Thales Netherlands B.V, Delft, The Netherlands  
sorin.iacob@nl.thalesgroup.com

## 1 Introduction

The essence of the concept emergence is aptly communicated by the following quote, attributed to Aristotle, who lived more than 2000 years ago: *The Whole is Greater than the Sum of its Parts*. The interactions of *Parts* can generate a *Whole* with unprecedented properties that go beyond the properties of any of its constituent *Parts*. The immense varieties of inanimate and living entities that are found in our world are the result of emergent phenomena that have a small number of elementary particles at their base.

A *System-of-Systems (SoS)* consists of a set of autonomous technical systems, called *constituent systems (CS)* that are independent and provide a useful service to their environment [18]. The purpose of building a System-of-Systems out of CSs is to realize new services that go beyond the services provided by any of the isolated CSs. *Emergence is thus at the core of SoS engineering.*

A Cyber-Physical System (CPS) is a synthesis of processes in the physical environment and computer systems that contain sensors to observe the physical environment and actuators to influence the physical environment. In most cases, the computer systems are distributed and contain computational nodes connected through networks that realize the information exchange among the nodes. A Cyber-Physical System-of-Systems (CPSoS) is an integration of stand-alone CPSs that provides services that go beyond the services of any of its isolated CPSs.

It is the objective of this chapter to investigate the phenomenon of emergence in CPSoS. In the following section we look at some prior work on emergence in the

---

This work has been partially supported by the FP7-610535-AMADEOS project.

© The Author(s) 2016

A. Bondavalli et al. (Eds.): Cyber-Physical Systems of Systems, LNCS 10099, pp. 73–96, 2016.

DOI: 10.1007/978-3-319-47590-5\_3

domains of philosophy and computer science. Since emergence is always referring to phenomena that occur at a given level of a hierarchic system model, Sect. 3 elaborates in detail on the concept of a *multi-level hierarchy*. Section 4 presents a definition of emergence in the SoS context and discusses some properties of emergent phenomena. Section 5 introduces a number of examples of emergent phenomena in computer systems. Section 6 discusses some design guidelines that help to detect the potential of emergent phenomena in a CPSoS and mitigate the effects of detrimental emergence. This Chapter terminates with a conclusion in Sect. 7.

## 2 Related Work

In philosophy the questions of how the diversity of the world emerges out of simple physical building blocks has been a topic of inquiry since the time of the ancient Greeks, leading to abundant literature about emergence, e.g., the survey articles [20, 34] or the books by [4, 6, 16]. Computer scientists got interested in the topic of emergence when it was realized that some striking phenomena that are observed at the system level of complex systems could not be explained by looking at the system's components in isolation. A well-publicized example of such a striking phenomenon is the flash crash of the stock market on May 6, 2010 [2]. Emergence can be regarded as an intriguing *part-whole relation* that investigates how the properties and the interaction of the parts lead to novel phenomena of a whole.

Holland remarks in [16]: *Despite its ubiquity and importance, emergence is an enigmatic and recondite topic, more wondered at than analyzed... It is unlikely that a topic as complicated as emergence will submit meekly to a concise definition and I have no such definition to offer.* Fromm [9, 10] elaborates on different forms of emergence and investigates the emergence of complexity in large systems. In [26], Mogul describes emergent misbehavior in a number of computer systems, discusses how emergence can manifest itself, and proposes a research agenda for studying the phenomena of emergence in complex computer systems. In the European Research Project TAREA SoS the current state of the art in the field of SoS has been captured [14] and a roadmap for future SoS research has been proposed. In this roadmap the topics of theoretical foundations of SoSs and of emergence are in a prominent position. In [19], Keating argues for the development of a firm epistemological foundation of emergence in SoSs. In the proceedings of the yearly IEEE conference on Systems of Systems Engineering and the book [18] by Jamshidi relevant contributions to the topic of emergence in SoSs can be found. Parunak and VanderBrok [28] and Huberman and Hogg [17] observed that variable temporal delays play a key role in the generation of emergent misbehavior in an SoS. In [5] Boschetti and Gray elaborate on the limits of insights gained from computer simulations when modeling emergent phenomena in natural systems.

### 3 Multi-level Hierarchy

The understanding and analysis of the immense variety of things and their behavior in the non-living and living world around us require appropriate modeling structures. Such a modeling structure must limit the overall complexity of a single model and support the step-wise integration of a multitude of different models. One such widely identified modeling structure is that of a *multi-level hierarchy*, where level-specific rules and laws govern the interdependence of entities at each level of the hierarchy. Since the phenomenon of emergence is always associated with levels of a *multi-level hierarchy* it is useful to start with a thorough discussion of *multi-level hierarchies*.

A *multi-level hierarchy* is a recursive structure where a system, the *whole* at the level of interest (the *macro-level*), can be *taken apart* into a set of sub-systems, the *parts*, that *interact* statically or dynamically at the level below (the *micro-level*). 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 internals of a sub-system is of no further interest. We call such a sub-system at the lowest level of interest (the *base* of the hierarchy) an *elementary part* or a *component*.

In his seminal paper *The Architecture of Complexity* Herbert Simon posits [32] (p. 219): If there are important systems in the world that are complex without being hierarchic, they may to a considerable degree escape our observation or understanding.

Our models of the *world of things* are organized along such a widely cited *Multi-level Material Hierarchy*, giving rise to the establishment of dedicated scientific disciplines for each level, e.g.:

- Atoms consist of elementary particles (the field of physics)
- Molecules consist of atoms (the field of chemistry)
- Cells consist of molecules (the field of biology)
- Organs consist of cells (the field of medicine).

#### 3.1 Whole versus Parts

Viewed from the macro-level, the whole is an *established entity* that encapsulates and hides its parts that interact at the lower level. If the parts at the micro-level that form the whole at the macro level are all identical we talk about a *homogeneous* structure, otherwise we talk about a *heterogeneous* structure.

At a given macro-level, we consider the whole as an entity that is surrounded by a *surface*. Interfaces located at the surface of the whole control the exchange of *matter*, *energy* or *information* among the wholes at the same level.

Koestler [21] (p. 341) has introduced the term *holon* to refer to the *two-faced character* of an entity in a multi-level hierarchy. The word *holon* is a combination of the Greek “holos”, meaning *all*, and the suffix “on” which means *part*. The point of view of the observer determines which view of a given holon is appropriate in a particular scenario.

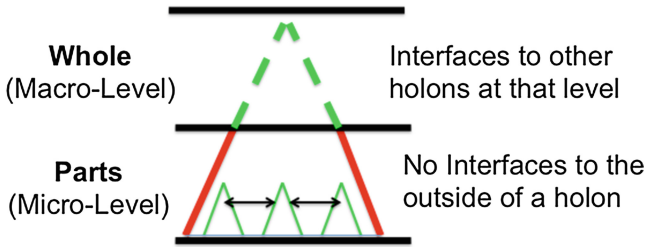


Fig. 1. Two-faced character of a holon

Figure 1 gives a graphical representation of the holon. Viewed from the outside at the macro level, a holon is a stable whole that can interact with other holons of that level by an interface across its surface. Viewed from below, the micro-level, a holon is characterized by a set of interacting parts that are confined by the boundaries of the holon. **This rigorous enclosure of the parts of a holon at the micro-level is absolutely essential to maintain the integrity of the abstraction of a holon as a whole at the macro level.**

Koester states in [21] (p. 343): Every holon has the dual tendency to preserve and assert its individuality as a quasi-autonomous whole; and to function as an integrated part of an (existing or evolving) larger whole. This polarity between Self-Assertive (S-A) and Integrative (INT) tendencies is inherent in the concept of hierarchic order and a universal characteristic of life.

There are two relations characterizing two adjacent levels of a hierarchy: (i) the *level relation* between the *whole* at the macro-level and the *parts* of the micro-level and (ii) the *interaction relation* among the *parts* of the micro-level.

### 3.2 Level Relations

The type of the *level relation* determines the character of a multi-level hierarchy. In this section we focus on three types of level relations, a *nested (or structure) hierarchy*, a *description hierarchy* and a *control hierarchy*. For the emergence of novel behavior in a CPSoS the control hierarchy is the most important.

**Structure Hierarchy.** We call a hierarchy a *structure (or nested) hierarchy* if the *whole comprises the parts* or, in different wording, the *parts are contained in the whole*, i.e., *consists of* (from the top to the bottom) or *forms* (from the bottom to the top) stand for the *level relation* of *containment*.

Structure hierarchies are formed by the identification and classification of the observation of physical structures that are existent in the world of things, irrespective of the subjective view of the observer. These physical structures are often formed by *physical force-fields* (see also Sect. 3.3, Physical Interactions).

The *Multi-level Material Hierarchy* referred to in the beginning of Sect. 3 above is an example for a *structure hierarchy*.

**Description Hierarchy.** A multi-level hierarchy that describes a *set of related entities* at different levels of abstraction is called a *multi-level description hierarchy*. A description hierarchy can be much simpler than the related *structure hierarchy* provided the structure hierarchy is highly redundant. If a complex structure is completely un-redundant, then it is its own simplest description [32] (p. 221).

We distinguish two types of descriptions, *state descriptions* and *process descriptions*. State descriptions describe the state of the world at the *instant of observation*. *Process descriptions* explain how a new state of the world unfolds as time progresses that is how the state transitions happen. A *description of behavior* is a process description.

The classification of entities in a description hierarchy is usually based on cognitive models of the observer and thus may be dependent on the subjective view of the observer. Moreover, depending on the purpose, different levels of description of the same physical structure can be introduced by the observer.

For example, the *thermodynamic description* of the behavior of a gas is at a higher level of description than the *statistical description* of the *same physical material* and the choice among them may depend on the purpose of the description.

If the redundancy of a structure is removed from its description hierarchy, then a significant simplification of the description can be realized (e.g., [32] p. 220).

In case the elements of a hierarchy are *constructs*, i.e. non-material entities that are the product of the human mind, the assignment of the constructs to hierarchical levels always results in a description hierarchy, the organization of which is determined by the purpose of the observer.

In many, but not in all cases, the description hierarchy of a structure follows the structure hierarchy.

**Control Hierarchy.** In a control hierarchy the macro-level provides some *constraints* on the structure or behavior of the parts at the micro-level thus establishing a causal link from the macro level to the micro-level. Constraints restrict the behavior of things beyond the natural laws, which the things must always obey.

In many, but not all cases, the *control hierarchy* follows the *structure hierarchy*. Ahl [1] (p. 107) provides the following example: The concept *army* denotes a structure hierarchy that *consists of the soldiers of all ranks and contains them all*. In contrast, a *general* at the top of an army (a military hierarchy) *controls* the soldiers, but *does not contain* them.

In some cases, as the example of the military hierarchy above shows, the control constraints originate from *outside*, i.e. above the macro-level. In other cases, the control constraints have their origin in the *whole*, i.e. the *collective behavior* of the parts of the micro-level. It is this latter case that is relevant for the analysis of emergence. Many equivalent examples can be found in Distributed Computing when we have centralized or decentralized control and management. Since behavior (function plus time) is a concept that depends on the progression of time, there is a temporal dimension in control hierarchies that deal with behavior.

Since the behavior of the parts forms the behavior of the whole, but the whole can constrain the behavior of the parts we have an example of a *causal loop* in such a control hierarchy.

We can observe such a causal loop in many scenarios that are classified as *emergent* in every-day language: the behavior of birds in flocks, the synchronized oscillations of fireflies or the build-up of a traffic jam at a congested highway.

Pattee [30] discusses control hierarchies extensively in *The Physical Basis and the Origins of Hierarchical Control*. In order to support the simplification at the macro-level and establish a hierarchical control level, a control hierarchy must on one side *abstract from* some degrees of freedom of the behavior of the parts at the micro-level but on the other side must *constrain* some other degrees of freedom of the behavior of the parts, i.e., a control hierarchy must provide *constraints from above*, while, in a multi-level material hierarchy the natural laws provide *constraints from below*.

**The delicate borderline between the constraints from above on the behavior of the micro-parts and the freedom of behavior of the micro-parts is decisive for the proper functioning of any control hierarchy.**

There are two extremes of control which lead to a collapse of the control hierarchy: (i) full control from above which defeats the principle of abstraction of control and leads to a full deterministic behavior and (ii) no constraints from above which can lead to unconstrained chaotic behavior (see Fig. 2).

For example, a good conductor of an orchestra will control the tempo of the performance without taking away the freedom from the musicians to express their individual interpretation of the music.

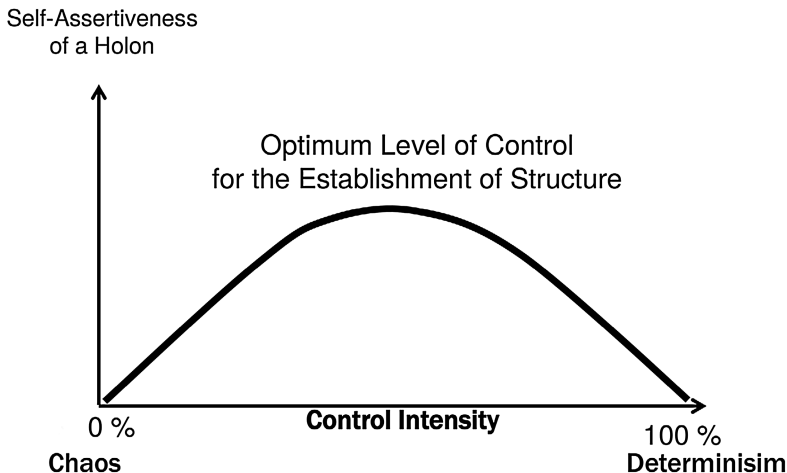


Fig. 2. Self assertiveness of a holon

### 3.3 Interaction Relations

**Formal Hierarchy.** Simon [32] (p. 195) calls a hierarchy a *formal hierarchy* if the *interaction relation* is empty, i.e., the parts are only related to the whole of the higher adjacent level. If, in the above example, the soldiers relate at a given level only to their boss, but not to each other, then we have an example of a formal hierarchy. Models that have the structure of a formal hierarchy are rare.

**Physical Interactions.** The *physical interactions* at any considered level of a material hierarchy can be classified in the following three dimensions: (i) *distance* among the parts, (ii) *force fields* among the parts and (iii) *frequency of interactions* among the parts. In general, as we move up the levels of a material hierarchy the *distance increases*, the *force-field magnitude decreases* and the *frequency of interactions* decreases [32].

Simon argues that the laws that govern the behavior at each level are *nearly independent* of the level above and below, giving rise to the principle of *near decomposability* [32] (p. 209) of levels.

**This principle of *near decomposability* states that an approximate model suffices in most cases to model the behavior at any given level of a multi-level hierarchy.**

This approximate model considers only the physical interactions at the considered level and abstracts from the behavior of the *high-frequency parts at the level below* and considers the dynamic parameters of the *low frequency parts at the level above* that provide the constraints as *constants*.

**Informational Interactions.** Informational interactions exchange information among the communicating partners. When the information exchanged consists of *data* and an *explanation of the data* we observe the exchange of *Itoms*.

**Itom:** An Itom is an atomic unit of *object data* and *meta data*. The object data represents some semantic content, and the meta data provides an explanation of the object data, i.e., how the semantic content represented by object data can be accessed. The semantic content of (or the information contained in) an Itom reports about a timed proposition relating to some entities in the world [23].

In a Cyber-Physical System-of-Systems (CPSoS) we distinguish between two types of informational interactions: (i) *message-based information* interactions in cyber space and (ii) *stigmergic information* interactions in the physical world.

Interactions in the cyber space allow in principle the exchange of explicitly defined Itoms which travel unmodified (invariant semantic content) from a sender to a set of receivers. Stigmergic interactions are indirect and involve influencing the state of the common environment of senders and receivers. Such environment may also be under the possible influence of *environmental dynamics*. Environmental dynamics are autonomous processes in the environment (physical world or cyber space) that also act on the state of the environment. Consequently, in stigmergic interactions it is – in many cases – not possible to send the same Itom from sender to receivers. Instead very often

receivers will only be able to observe object data which is (more or less closely) related to the original data sent and needs to be correctly interpreted to avoid *property mismatch*. A model of the environmental dynamics able to represent the processing and modifications performed on data would be paramount in the understanding and mastering of stigmergic information exchange.

In cyber space data is represented by a bit-pattern that can be generated by the processing of stored *Itoms* or by some data acquisition process, e.g., by a sensor. For data acquisition, the design of the sensor determines how the acquired bit pattern has to be interpreted, i.e., provides for the explanation of the object data.

Since an *Itom* is a higher-level concept than the sole *object data* in an *Itom*, we propose to use *Itoms* in the specification of *Relied-Upon Interfaces (RUIs)* among the Constituent Systems (CSs) of a CPSoS (see Chap. 2). According to [23] the full specification of an *Itom* has to provide answers to the following questions:

- **Identification:** *What entity is involved?* The entity must be clearly identified in the space-time reference frame.
- **Purpose:** *Why is the data created?* This answer establishes the link between the raw data, the refined data and the purpose of the CPSoS.
- **Meaning:** *How has the data to be interpreted by a human or manipulated by a machine?* If the answer to this question is directed towards a human, then the presentation of the answer must use symbols and refer to concepts that are familiar to the human. If a computer acquires data, then the explanation must specify how the data must be manipulated and stored by the computer.
- **Time:** *What are the temporal properties of the data?* Real-time data must include the instant of observation in the entity. In control applications it is helpful to include a second timestamp, a *validity instant* that delimits the validity of the control data as part of the *Itom* [22] (p. 4).

**Message-based Information Flows:** A *message-based information flow* is present if one CS sends a message to another CS. In many legacy distributed systems only *object data* is contained in a message while the *explanation of the data* is derived from the context.

In a CPSoS the involved CSs can be operating in differing contexts, e.g., in the *US* and *Europe*. For example, in the *US* temperature is represented by *degrees Fahrenheit*, while in *Europe* temperature is represented by *degrees Celsius*. As a consequence, the *same data (bit-patterns)* can convey a different meaning if the contexts of the sender differs from the context of the receiver of the message, causing a *property mismatch*. Such *property mismatches* have been the cause of severe accidents.

**Stigmergic Information Flows:** A *stigmergic information flow* is present if one sending CS acts on the physical environment and changes the state of the environment and later on another receiving CS observes the changed state in the environment with a sensor that captures the *sensor specific aspect* of the environment [24]. Consider, for example, the coordination of cars on a busy highway to realize a smooth flow of traffic. In addition to the direct communication by explicit signals among the drivers of the cars (e.g., the blinker or horn), the *stigmergic information flow* based on the observation of the movement of the vehicles on the road (caused by the actions of other drivers) is a primary source of information for the assessment of a traffic scenario. An



important characteristic of stigmergic information flows is the consideration of up to date *environmental dynamics*.

**Hidden Channels.** There exist many indirect information flows, in particular stigmergic ones, which remain both (i) unknown to the sender which is not aware of the flow, and (ii) are not captured by systems designers or modelers. We call such existing interaction relations *hidden channels*.

Hidden channels are problematic, because they can contribute to the generation of causal loops (and therefore take active part in the rise of emergent phenomena). In addition, these causal links may lead to a **modification of the understood holarchy abstraction**, i.e., parts of one level interact directly with parts of another level **which may establish hidden level relations** (e.g., a control hierarchy). Effects of such modification of the holarchy abstraction may cause both unintended information leakage (violations of security properties) and unexpected negative emergence.

Usually it is difficult to protect the state of the physical environment regarding observations of receivers. Additionally, in many cases a sender may be even unaware of leaking information to its environment. For example, consider security attacks based on observing the electromagnetic emissions of a processor on smart cards [11].

Still, hidden channels should be avoided by properly identifying them (see Sect. 6.1) or insulating against them (e.g., firewalls, physical insulation).

## 4 Emergence

It is quite common, as we move up a multi-level hierarchy, that novel phenomena can be observed at a given level that are not present at the level below. We call these new phenomena *emergent phenomena*. We use the term *phenomenon* as an umbrella term that can refer to *structure, behavior or property*.

In many cases the laws that explain the genesis of these emergent phenomena are formulated *post facto* because it would require a *very knowledgeable mind* to predict *a priori* all possible phenomena that can come into existence out of the interactions of many given parts. The first appearance of an emergent phenomenon is *often a surprise* to a human observer.

### 4.1 Definition of Emergence

In order to achieve a level of objectivity we aim for a definition of emergence that is based on a *property of the scenario* and not on a *relation* between the scenario and the observer.

Let us analyze the relationship between two adjacent levels of a multi-level hierarchy, the *micro-level* (the level of the *parts*) and the *macro-level* (the level of the *whole*) where emergent phenomena are observed, assuming that the *level relation is given*. We restrict our analysis to these two levels and disregard the case where some properties of the parts are themselves emergent with respect to their lower-level parts. Our definition of emergence in a Cyber-Physical Systems-of-Systems is the result of

many interdisciplinary discussions during the *AMADEOS Workshop on Emergence in Cyber-Physical Systems-of-Systems* [15].

**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.**

A phenomenon is of a *new kind* if the concepts required to *explain* this phenomenon cannot be found in the world of the isolated parts. *Conceptual Novelty* is thus the landmark of our definition of emergence.

Note that, according to the above definition, the emergent phenomena *must only be of a new kind with respect to the non-relational phenomena of the parts*, not with respect to the knowledge of the observer. If a phenomenon of a whole at the macro-level is *not of a new kind* with respect to the non-relational phenomena of any of its proper parts at the micro level then we call this phenomenon *resultant*.

The essence for the occurrence of emergent phenomena at the macro-level (the *SoS level*) lies in the *interactions of the parts* at the micro-level, i.e., in the spatial arrangement of the parts caused by physical force-fields and/or the *designed temporal informational interactions* among the parts at the micro-level.

In CPSoS, the phenomenon we are interested in is *behavior*. In a CPSoS the *observable behavior* of a system is the *temporal sequence of observable states* of the system in the *Interval of Discourse*. We are thus interested in *diachronic* emergence, where initial interactions of the parts at the micro-level precede the appearance of the emergent phenomenon at the macro level.

We assume that the temporal distance between two observation instants of an observer is a multiple of a smallest duration. This smallest temporal distance expresses the *grain of observation* of this particular observer. If the duration of a state is shorter than the *grain of observation* then this short-lived state may evade the observations of this observer. The duration of the grain of observation should be selected on the basis of the *purpose of the observer*, the *dynamics of the observed system* and the *minimal response time* of the entities at the chosen level of observation.

Some scientists posit that emergent behavior is connected with a *surprise of the observer* [31]. According to this view, emergence occurs, if the causal link between the *interactions of the parts* and the *behavior of the whole* is *non obvious* to the observer (and therefore a surprise to the observer). According to this definition, the *state of knowledge of the observer* is the decisive criterion for the classification of a phenomenon as *emergent*. As a consequence, different observers with different states of knowledge will judge the same phenomenon differently. It follows that emergence is considered a *relation* between the whole and the *observer* and not a *property* of the whole.

## 4.2 Explained vs Unexplained Emergence

At first we pose the question whether emergent properties are *reducible* to the properties of the parts considered in isolation.

The following quote about Scientific Reduction is taken from the Stanford Encyclopedia on Philosophy:

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 a CPSoS, 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. It follows that *emergent properties* in a CPSoS are not *reducible* to the parts considered in isolation.

According to our definition of emergence in Sect. 4.1, a novel phenomenon is considered *emergent*, irrespective of whether it can be *explained* how the new phenomenon at the macro level has developed out of the parts at the micro-level. Given the *present state of knowledge*, some of these emergent phenomena can be explained by existing theories while there are other emergent phenomena where at present no full explanation can be given as to how they developed. Examples for (as of today) unexplained emergence are the *generation of life* or the *generation of the mind* on top of the neurons in the brain.

But what constitutes a *proper scientific explanation*? Hempel and Oppenheim [13] (p. 138) outlined a general schema for a *scientific explanation* of a phenomenon as follows:

**Given**

*Statements of antecedent conditions*

**and**

*General Laws*

**then a logical deduction of the**

*Description of the empirical phenomenon to be explained*

**is entailed.**

The *antecedent conditions* can be initial conditions or boundary conditions that are *unconstrained* by the general laws.

The *general laws* can be either universally valid *natural laws* that reign over the behavior of things or *logical laws* describing a valid judgment in the domain of constructs. Natural laws do not change in time or have a memory of the past. A natural law, such as a physical law, must hold everywhere, no matter what level of a multi-level hierarchy is the focus of the investigations.

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 *inexorable* and *universally valid* while established rules are *structure dependent* and *local*. Rules about the behavior of things are based on more or less meticulous experimental observations. A special case is the introduction of *imposed rules*, e.g., the rules of an artificial game, such as chess. The degree of accuracy and rigor of various established rules differ substantially.

It thus follows that between the two extremes of *scientifically explained* and *not explained at all* there is a *continuum of explanations* that are more or less acceptable and are relative with respect to the general state of knowledge and the opinion of the observer at a given point in time.

### 4.3 Conceptualization at the Macro-level

According to our definition of emergence, novel concepts should be formed and new laws may have to be introduced to be able to express the emerging phenomena at the macro level appropriately. Note that the emergent phenomena and laws must be new w. r.t. the phenomena of the isolated parts, but not necessarily new with respect to the knowledge of the observer, i.e., such phenomena are emergent irrespective of the state of knowledge of the observer.

In the history of science, many novel laws that employ *new concepts* have been introduced to capture the newly observed regularities of phenomena at a macro-level. We call such a new law that deals with the emerging phenomena at a macro level *an intra-ordinal law* [27]. At a later time, some of these laws have been reduced to well-understood effects of the parts at the adjacent micro-level, e.g., the *thermodynamic theory of a gas* can be explained by the *statistical theory of gas* [3].

Since the concepts at the macro level are new with respect to the 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. Therefore, it is often necessary to formulate *inter-ordinal laws* (also called *bridge laws*) to relate the established concepts at the micro-level with the new concepts of 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.**

Let us look at the example of a transistor. The *transistor effect* is an emergent effect caused by the proper arrangement of dopant atoms in a semiconducting crystal. The exact arrangement of the dopant atoms is of no significance as long as the provided behavioral specifications of a transistor are met. In a VLSI chip that contains millions of transistor, the detailed microstructure of every single transistor is probably unique, but the external behavior of the transistors (the holons) is considered the *same* if the behavioral parameters are within the given specifications. It is a tremendous simplification for the designer of an electronic circuit that she/he does not have to consider the unique microstructure of every single transistor.

### 4.4 Downward Causation

In classical physics, the concept of causation links an *effect* to an earlier *cause*. If in the domain of Newtonian mechanics precisely defined initial conditions (the *cause*) are given, an object will move along a trajectory (the *effect*) that is fully determined by the differential equations that express the laws of macro-mechanics. However, in the domain of micro-mechanics, where quantum-physical laws reign, it is not possible to observe the initial conditions of an object without influencing the object of observation. This is one of the reasons, why the concept of unidirectional causation is highly debated in the modern sciences. Another reason pertains to the multitude of parameters, captured in the notion of a *causal field* that characterizes the causes of real-life

phenomena. It is often up to subjective judgment to determine which one of these many causes is considered the *most prominent cause*.

On the other side, the *unidirectional cause-effect* relation plays a prominent role in our subjective models of the world in order to realize intended effects or to avoid the *causes of undesired effects*. To quote Pattee [29] (p. 64 onwards): *I believe the common everyday meaning of the concept of causation is entirely pragmatic. In other words, we use the word cause for events that might be controllable... the value of the concept of causation lies in its identification of where our power and control can be effective. ... when we seek the cause of an accident, we are looking for those particular focal events over which we might have had some control. We are not interested in all those parallel subsidiary conditions that were also necessary for the accident to occur, but that we could not control... .*

Along this line of reasoning the *term downward causation* denotes the concept that the whole at the macro-level can *constrain* or even *control* the behavior of the parts at the micro-level (the level below).

*Downward causation is a difficult concept to define precisely, because it describes the collective, concurrent, distributed behavior at the system level. ... Downward causation is ubiquitous and occurs continuously at all levels, but it is usually ignored simply because it is not under our control. ... The motion of one body in an n-body model might be seen as a case of downward causation [29] (p. 64).*

Downward causation establishes a *causal loop* between the *micro-level* and the adjacent *macro level*. The interaction of the parts at the micro-level causes the whole at the macro-level while the whole at the macro-level constrains the behavior of the parts at the micro-level (see also Sect. 5.2). We conjecture that in a multi-level hierarchy emergent phenomena are likely to appear at the macro-level when there is a causal-loop formed between the micro-level that forms the whole and the whole (i.e., the ensemble of parts) that constrains the behavior of the parts at the micro-level.

In a system that exhibits downward causation the degrees of freedom of the parts that can be exploited at the micro-level, e.g., by mechanisms of self-organization are limited by:

1. Constraints on the degrees of freedom of material parts at a micro-level coming from below, i.e., *upward causation* deriving from applicable *natural laws*, e.g., *the laws of physics*.
2. Constraints on the degrees of freedom of a part at the micro-level coming from above, the *whole* at the macro-level by *downward causation*.

Note that in a concrete system, some of these categories can be empty. For example, in a hierarchy of *constructs* there is no *upward causation*, i.e. constraints on the parts from below caused by natural laws.

In our opinion the *exclusion argument* by Kim [20] —that in a system with downward causation *macro causal powers compete with micro causal powers* and, if this is the case, micro causal powers will always win, needs to be reconsidered since the macro causal powers and the micro causal powers restrict different degrees of freedom of the parts and are thus not in conflict.

Another different way in which emergence is observed in practice in the real world also is the one caused by a *Cascade effect* [8]. A cascade effect exists, if in a system

with a multitude of parts at the micro level a state change of a part at the micro-level causes successive state changes of many other parts at the micro level. The cumulative effect of the totality of these state changes results in a novel phenomenon, such as an *avalanche* or a *nuclear explosion*. An *epidemic* is also a good example for a cascade effect. Cascade effects are *diachronic*, since they develop over time.

There may be other mechanisms that lead to emergent phenomena that we have not yet identified.

### 4.5 Supervenience

The principle of *Supervenience* [25] establishes an important dependence relation between the emerging phenomena at the macro-level and the interactions and arrangement of the parts at the micro-level. *Supervenience* states that

*Sup\_1*: a given emerging phenomenon at the macro level can emerge out of many different arrangements or interactions of the parts at the micro-level while

*Sup\_2*: 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 the many different arrangements or interactions of the parts at the micro level that lead to the same emerging phenomena at the macro level—see the example of the *transistor* above. *Sup\_1* entails a *significant simplification* of the higher-level models of a multi-level hierarchy.

Because of *Sup\_2* any difference in the emerging phenomena at the macro level can be traced to some significant difference at the micro level. *Sup\_2* is important from the point of view of *failure diagnosis*.

### 4.6 Classification of Emergence

Figure 3 depicts a schema for the classification of emergent phenomena.

In a CPSoS the CSs interact, i.e., via message-based channels in cyber space in which they exchange *Itoms*, and interact also via stigmergic channels information flows in the

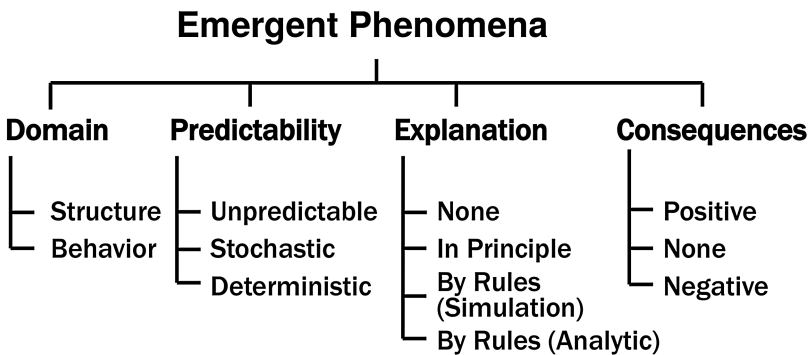


Fig. 3. Classification of emergent phenomena

physical world. These interactions can give rise to *emergent behavior* at the level of CPSoS. Although this behavior is explainable in principle, we may not be able to explain or predict this behavior in practice due to our ignorance about the full scope of the CPSoS, the precise temporal interactions among the CS (see e.g. the deadlock example in Sect. 3.5) and hidden communication channels behind the interfaces of a CS.

## 5 Examples of Emergence in Computer Systems

In this Section we discuss a number of examples of emergent behavior in computer systems. The first four examples can be explained, while the fifth example, the *Flash Crash* of the stock market on May 6, 2010 [2], although *explainable in principle* has not been *explained in practice* up to today.

### 5.1 Deadlock in Computer Systems

In some publications, the occurrence of a *deadlock* in a computer system is called an *emergent phenomenon* [12]. With the advent of multi-programming computer systems, the following event has been occasionally observed: when executing a number of processes concurrently, the system comes to a *permanent halt*, although each process, executed in isolation executes flawlessly. At first, this phenomenon could not be explained and was considered a *surprise*. Later on (around the year 1970) a *full explanation* of this phenomenon, called *deadlock*, was given [7]. The following simple example of Fig. 4 explains the essence of the phenomenon deadlock.

Let us consider the execution of a seat reservation system (cf. Fig. 4) in an *ideal world*, where no failures of the computer hardware will ever occur. As long as only a

Process Type A	Process Type B
1 $S^{\text{money}} = 1, S^{\text{seat}} = 1$	1 $S^{\text{money}} = 1, S^{\text{seat}} = 1$
2 Client selects seat and provides credit card	2 Client selects seat and provides credit card
3 Wait ( $S^{\text{money}}$ )	3 Wait ( $S^{\text{seat}}$ )
4 Get Money	4 Get Seat
5 If No-Money Then Signal ( $S^{\text{money}}$ ) Print No Money Goto 2	5 If No-Seat Then Signal ( $S^{\text{seat}}$ ) Print No Seat Goto 2
6 Wait ( $S^{\text{seat}}$ )	6 Wait ( $S^{\text{money}}$ )
7 Get Seat	7 Get Money
8 If No-Seat Then Return Money Signal ( $S^{\text{money}}$ ) Signal ( $S^{\text{seat}}$ ) Print No Seat Goto 2	8 If No-Money Then Return Seat Signal ( $S^{\text{money}}$ ) Signal ( $S^{\text{seat}}$ ) Print No Money Goto 2
9 Signal ( $S^{\text{money}}$ ) Signal ( $S^{\text{seat}}$ )	9 Signal ( $S^{\text{money}}$ ) Signal ( $S^{\text{seat}}$ )
10 Print Seat Ticket	10 Print Seat Ticket
11 Goto 2	11 Goto 2

Fig. 4. Example of deadlock

finite number of reservation processes of Type A are executed concurrently, the system will operate flawlessly forever. The same will happen if only a finite number of reservation processes of Type B execute concurrently. However, if a finite number of processes of Type A and processes of Type B operate concurrently, the system will sometimes *stop forever (deadlock)*. *Stopping forever* is the novel phenomenon that is not happening if processes of Type A or processes of Type B operate in isolation.

In the program sketch of Fig. 4 there are two semaphore variables,  $S^{\text{money}}$  and  $S^{\text{seat}}$  initialized with the value 1. Whenever a process executes a *Wait* operation on a semaphore variable, the process is only allowed to enter the following *Critical Section* if the value of the semaphore variable is positive at the start of execution of the *atomic operation Wait*. The atomic operation *Wait* tests the value of the designated semaphore variable. In case the test gives a positive value, it decreases the value of the semaphore variable by 1 and enters the Critical Section. Otherwise it waits until the value of the semaphore variable gets positive. The semaphore operation *Signal*, executed at the end of a *Critical Section*, increases the value of the designated semaphore variable by 1 and thus enables another waiting process to enter the Critical Section.

In Fig. 4, the semaphore  $S^{\text{money}}$  ensures that in the following *Critical Section*, dealing with the *money* only a single process is allowed to execute at an instant. Likewise, the semaphore variable  $S^{\text{seat}}$  ensures that in the following *Critical Section* dealing with the *seat allocation* only a single process is allowed to execute at a time. As long as processes of type A execute concurrently, the execution of  $\text{Wait}(S^{\text{money}})$  is always followed by  $\text{Wait}(S^{\text{seat}})$ .

However, if the executions of processes of *Type A* and *Type B* are interleaved, then it can happen that a process of Type A enters the Critical Section protected by  $S^{\text{money}}$  and, before the process of Type A executes the operation  $\text{Wait}(S^{\text{seat}})$  a process of Type B enters its critical Section protected by  $S^{\text{seat}}$ . From now on, a deadlock is unavoidable if the *money* and the *seat* are available, since both processes have to *wait forever* on the release of the respective following *Critical Section*.

The observed phenomenon of *deadlock* fulfills the requirement of an emergent phenomenon:

- The phenomenon *deadlock—halting forever*—is novel with respect to the simple world of an individual processes, where the notion of *halting forever* is not present.
- There is *downward causation*. The system of concurrently executing processes *constrains* the execution of an individual process by *indirect communication channels* established by the semaphore variables.

It is important to note that although this phenomenon is *fully explainable* it is *not predictable*, even in theory. If two processes try to execute the same semaphore operation exactly simultaneously, the underlying hardware enters into a state of *meta-stability* [33] (p. 77). It is not predictable, *even in theory*, which one of the two *simultaneous* processes will win this race.

It is also revealing to look at the problem of deadlock from the point of view of determinism. Although each one of the individual processes, the parts, behaves *deterministically* the behavior of the overall system, the whole, is *non-deterministic*.



## 5.2 Distributed Fault-Tolerant Clock Synchronization

In a time-triggered distributed computer system computational and communication processes are triggered by the progression of a global notion of physical time. This global notion of physical time must be *fault-tolerant* in order to mitigate the effects of a failing physical clock.

A distributed fault-tolerant synchronization algorithm constructs the fault-tolerant global time. Such an algorithm comprises the following three phases [22] (p. 69):

1. Periodic exchange of the time value of the local clock of each computing node among all the nodes of the system.
2. Distributed calculation of a global fault-tolerant time value, taking the local readings of the clock as inputs.
3. Adjustment of the local clock to come into agreement with the calculated global fault tolerant time value.

According to the theory of clock synchronization the number  $N$  of clocks in a system must be larger than  $3k$ , where  $k$  is the number of faulty clocks i.e.,  $N \geq (3k + 1)$ .

A physical clock is a device that contains a physical oscillator (e.g., a crystal) and a counter that counts the number of ticks of the oscillator and thus contains the *state of the clock*. The frequency of the physical oscillator is determined by the *laws of physics* and depends on the size of the crystal and environmental conditions, such as temperature or pressure—a case of *upward causation*. The speed of the oscillator cannot be modified by *downward causation*. However, the state of the clock is modified by *downward causation* in step iii of the algorithm.

The phenomenon fault-tolerant clock synchronization fulfills the requirement of an emergent phenomenon:

- The phenomenon *fault-tolerant time*, which does not fail if a single clock fails, is novel with respect to the behavior of a single clock that can fail.
- There is *downward causation*. The system of concurrently executing clocks constrains the execution of an individual clock by adjusting the state of the counter of the local clock to a value that has been determined by the ensemble of clocks.

This example of emergence is interesting from the point of view of how *upward causation* (the frequency of a physical clock) and *downward causation* (the periodic correction of the state of a clock caused by the time value calculated by the ensemble of clocks at the macro level) interact and form a causal loop.

## 5.3 Alarm Processing

In an industrial plant an *alarm* is triggered when the value of a significant state variable exceeds a preset threshold limit. There may be thousands of significant state variables that are monitored in a large industrial plant. Since a single serious fault may cause a *correlated alarm shower* an alarm processing system must reduce the alarm rate at the operator interface to a manageable level in order to avoid an operator overload. The

alarm processing system establishes the causal dependencies of alarms and decides which alarms can be hidden from the operator.

An alarm processing system consists of distributed sensors that can detect alarms and send alarm messages, a communication system that transports the alarm messages to an alarm processing center and the alarm analysis software that decides which alarm to hide.

Alarms are events that happen infrequently in normal operation. Many communication protocols for the transport of the alarm messages are of the PAR (Positive Acknowledgment of Retransmission) type for the transmission of event messages. The PAR protocol contains a retransmission mechanism to resend a message in case the previously sent message is not acknowledged in due time. Under heavy load, this mechanism can lead to a cascade effect

In the case of a correlated alarm shower that arises from a single serious fault, the event-triggered communication system slows down because the increased load on a finite capacity channel causes a delay of some messages. This slow-down induces the retransmission mechanism to kick in and to increase the load on the communication system even further. This can lead to a collapse called *thrashing*—an emergent phenomenon.

- The phenomenon *thrashing*, is novel with respect to the behavior under normal operation.
- There is *downward causation*. The high-load on the communication causes a slowdown of the communication system that causes the retransmission mechanism to increase the load even further.

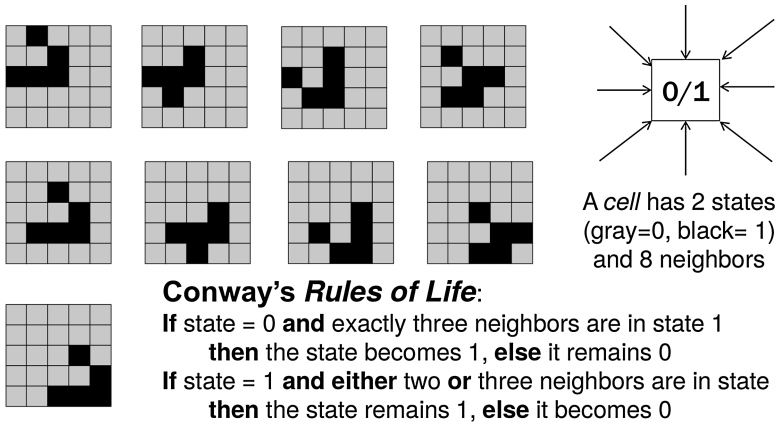
## 5.4 Conway's Game of Life

Conway's Game of Life is a simple cellular automaton. It is played on a set of cells organized in a square array. Since there are no *things* involved, there is no upward causation from natural laws.

The simple rules of Conway's game of life are shown in Fig. 5. A player can select the initial conditions, i.e. the initial marking of the cells on the square array, as he/she pleases. After a round of updating all cells according to the transition rules, a new marking on the square array comes into sight. This marking forms the initial conditions for the following round, etc. Given defined initial condition, the series of states that develop is deterministic.

Let us choose the pattern for the initial conditions as shown in the left upper corner of Fig. 5. If all other cells of the square array are empty, then a phenomenon called *glider* appears.

If we select a *grain of observation* that observes the evolving patterns on the square array only after every four rounds then we clearly see the *glider* moving down diagonally along the square array. Holland calls this an emergent phenomenon [16].



**After four cycles, the pattern has moved along the diagonal.**

**Fig. 5.** Conway's game of life

- The *moving glider* is a *deterministic consequence* of the selected initial conditions and the rules of the *game of life* at the micro-level. If the moving glider meets on its passage a *non-empty cell* of the square array then the *moving glider* disappears.
- The phenomenon of the *moving glider* that is observable on the selected macro level of a description hierarchy (Sect. 3.2) is *novel* and a surprise to a human observer. It is very difficult for the human mind to predict the patterns that will evolve deterministically from an initial condition in the course of many rounds.
- There is *downward causation* (a *feedback loop*) from one round to the next round, because the pattern that comes to sight after *all cells* have executed a round forms the initial condition for *each cell* in the following round.

## 5.5 Stock Market Crash on May 6, 2010

In today's electronic financial markets, an electronic trader can execute more than 1000 trades in a single second. The actions of a multitude of human traders and automated trading systems at the micro-level cause the valuation of the assets at the macro level which in turn influences the actions of the human traders and the algorithms of the automated trading systems, thus forming causal loops and cascade effects that can result in emergent misbehavior.

Aldrich et al. [2] reports about such a misbehavior of the stock market, called the Flash Crash on May 6, 2010: "... in the span of a mere four and half minutes, the Dow Jones Industrial Average lost approximately 1,000 points."

"As computerized high-frequency traders exited the stock market, the resulting lack of liquidity causes shares of some prominent companies to trade down as low as a penny or as high as \$100.000" (N.Y Times, October 1, 2010)

About half an hour after the start of the Flash Crash, the stock market stabilized at a level that was significantly below the pre-crash valuation, destroying billions of dollars of equity.

The Flash Crash raises difficult, policy-relevant questions of causation. As is the case with most market events, the circumstances of the Flash Crash cannot be reconstructed because a detailed record of the precise temporal order of all relevant events is not available. *This “Flash Crash” occurred in the absence of fundamental news that could explain the observed price pattern and is generally viewed as the result of endogenous factors related to the complexity of modern equity market trading* Aldrich et al. [2].

Analysts lack access to the specifications of the automated trading algorithms that were active in the markets prior to and during the crash, and cannot replicate the strategies implemented by human traders active during the relevant period. Intense investigations and congressional hearings followed, but conclusive evidence is still missing six years after the crash. Although the sequence of events that caused the Flash Crash is explainable in theory it cannot be reconstructed in practice due to the concurrency and ignorance about the immense multitude of interacting transactions.

## 6 Consequences for CPSos Design

In CPSoS design not all the combinations allowed by Fig. 3 are of interest, in fact we are particularly interested in the behavior domain, i.e., behavioral emergence. Figure 6 classifies the emergent behavior of a CPSoS from the point of view of the consequences of this behavior on the overall mission of a CPSoS and from the prediction or awareness we may have on the appearance of emergent behavior.

*Expected and beneficial* emergent behavior is the normal case (quadrant 1) that results from a conscious design effort. *Unexpected and beneficial emergent* behavior is a positive surprise (quadrant 3). *Expected detrimental emergent* behavior can be avoided by adhering to proper design rules (quadrant 2). The problematic case is quadrant 4, *unexpected detrimental emergent behavior*.

In safety-critical CPSoSs, an unexpected detrimental emergent behavior can be the cause of a catastrophic accident. But how can we detect and avoid an *unknown* and therefore *unexpected* emergent phenomenon?

	Beneficial	Detrimental
Expected	1 Normal Case	2 Avoided by Design Rules
Unexpected	3 Positive Surprise	4 Problematic Case

Fig. 6. Contribution of emergent behavior

Clearly a conscious and aware design discipline aims to move, as knowledge progresses, more and more emergent phenomena from quadrant 4 to quadrant 2, in which provisions can be taken to mitigate, eliminate or prevent detrimental emergence. To exemplify just observe that while at its first manifestation deadlock was a problematic issue in distributed systems, today every computer student is though many of the different ways we have developed to properly address it.

Still our knowledge regarding CPSoS may remain limited and our ignorance about them can hardly be sufficiently reduced especially when we consider COTS components and legacy constituent systems. In fact, most CPSoS are built incorporating such LEGACY and COTS on which very little is known and where the information flow is often quite hidden.

In the remainder of this section we will focus on quadrant 4, the problematic case of *detrimental unexpected emergent with special* regards to undiscovered emergent phenomena never seen before.

### 6.1 Exposure of the Direct and Indirect Information Flow

In a CPSoS emergent behavior is the result of direct or indirect *flow of information* among the constituent systems.

At design time, the planned message-based, stigmergic and sometimes human information flow patterns should be analyzed in order to find *potential causal loops* and *cascade effects*. However, this analysis has limits where part of the information flow is hidden behind the interface of a CS whose interface model is incomplete because it abstracts from the details of the world behind the interface.

At run time, the actual information flow should be observed without the probe effect and documented with precise timestamps such that the temporal order of events can be reconstructed in a *post hoc* analysis of a scenario to establish the precise sequence that led to detrimental emergent behavior. This POST MORTEM analysis would be particularly useful to discover and explain new (just encountered) emergent phenomena. Actually such analysis, coupled with disclosure of the internal algorithms used for automatic trading would have allowed to explain the Stock Market Crash (Sect. 5.5).

### 6.2 Safety-Critical Systems

The behavior of a safety-critical system should conform to the *design model* that is the basis of the safety argument. The design model *does not* and *cannot* take into account *unknown emergent effects* that can cause a deviation of the actual behavior from the intended behavior.

Since in safety-critical CPSoS even a very small probability for a detrimental emergent phenomenon cannot be tolerated, it is proposed that the evolving state of a safety-critical CPSoS is meticulously monitored by an independent monitor component in order to detect the *onset* of an unexpected deviation of *the actual state* from the *intended state*. This deviation can be an indication for the start of an unknown (and therefore unexpected) detrimental emergent behavior. The system internal information

flow to the monitoring system must operate in real-time in order that the monitor can act promptly. Since emergent behavior is *diachronic*, (i.e. it develops over time) an independent *meta (monitoring) system* that continually observes the evolving state of the *object system* can detect the early onset of a deviation and thus provide an immediate warning of a forthcoming disruption due to an emergent phenomenon. Based on this immediate warning, mitigating actions can be activated that bring the object system *back to normal operation* or at least to a *safe state*.

It is important to note that the monitoring system should be *state-based*, and not *process-based*. A state-based monitoring system acts on a higher-level of abstraction than a process-based system since it is concerned with the *properties of the states* of a system only and not with the much more involved processes that *generate the state changes*. A state-based monitoring system is thus much simpler than a process-based monitoring system. This fundamental difference between a state based and a process-based system is also important from the point of view of *design diversity* to detect hidden software errors.

Taking again the example of the Stock Market Crash (Sect. 5.5), if an independent monitoring system (without knowledge of the trading algorithms) had continually observed significant parameters that are relevant indicators of the market state and it had acted in the sub-millisecond range to stop the trading activities (safe state) the flash-crash that disrupted the market and wiped out billions of dollars of equity could have been avoided.

## 7 Conclusions

The purpose of building a Cyber-Physical System-of-Systems out of Constituent Systems (CSs) is to realize new services that go beyond the services provided by any of the CSs in isolation. *Emergence is thus at the core of CPSoS engineering*. In this Chapter we have surveyed some of the abundant past literature on emergence from the fields of philosophy and computer science, looked at the characteristics of multi-level hierarchies, developed a CPSoS definition of emergence and analyzed some examples of emergent behavior in computer systems.

We identified the basic mechanism that can lead to emergent phenomena: *causal loops* between the macro-level and the micro-level of a multi-level hierarchy (with the variant of *cascade effects*) that result in conceptually novel phenomena. We came to the conclusion that due to the ignorance about the scope of CPSoS even a thorough design analysis cannot uncover all potential mechanisms that can result in unexpected emergent phenomena at run-time. Unexpected emergent phenomena manifest themselves in a CPSoS by a diachronic deviation of the actual behavior from the intended (design) behavior.

Since *unknown emergent effects* can be the cause of a deviation of the actual behavior from the intended behavior, the meticulous observation of the behavior of a safety-critical CPSoS by an independent monitoring system can detect the onset of diachronic emergence and initiate mitigating actions before the detrimental emergent phenomenon has fully developed.

## References

1. Ahl, V., Allen, T.F.H.: *Hierarchy Theory. A Vision. Vocabulary and Epistemology*. Columbia University Press, New York (1996)
2. Aldrich, E.M., Santa Cruz, U.C., Grundfest, J.: Stanford university law school, Laughlin, G., Santa Cruz, U.C.: *The flash crash: a new deconstruction*, 25 January 2016, Revised 2 February 2016. [http://papers.ssrn.com/sol3/papers.cfm?abstract\\_id=2721922](http://papers.ssrn.com/sol3/papers.cfm?abstract_id=2721922)
3. Beckerman, A., et al. (eds.): *Emergence or Reduction—Essays on the Progress of Nonreductive Physicalism*. Walter de Gruyter, Berlin (1992)
4. Bedau, M.A., Humphreys, P.: *Emergence, Contemporary Readings in Philosophy and Science*. MIT Press, Cambridge (2008)
5. Boschetti, F., Gray, R.: *Emergence and computability*. *ECO* **9**(1), 120–130 (2007)
6. Clayton, P., Davies, P.: *The Reemergence of Emergence*. Oxford University Press, New York (2006)
7. Coffman, E.G., et al.: *System deadlocks*. *ACM Comput. Surv.* **3**(2), 67–78 (1971)
8. Fisher, D.A.: *An emergent perspective on interoperation of system-of-systems*. Technical report CMU/SEI-2006-TR-003, Carnegie Mellon University (2006)
9. Fromm, J.: *The Emergence of Complexity*. Kassel University Press, Kassel (2004)
10. Fromm, J.: *Types and forms of emergence*. [arXiv.org/pdf//nlin/0506028.pdf](https://arxiv.org/pdf/nlin/0506028.pdf). Accessed 12 May 2016
11. Gandolfi, K., Mourtel, C., Olivier, F.: *Electromagnetic analysis: concrete results*. In: Koç, Ç. K., Naccache, D., Paar, C. (eds.) *CHES 2001*. LNCS, vol. 2162, pp. 251–261. Springer, Heidelberg (2001). doi:10.1007/3-540-44709-1\_21
12. Gligor, V.: *Security of emergent properties in ad-hoc networks (transcript of discussion)*. In: Christianson, B., Crispo, B., Malcolm, J.A., Roe, M. (eds.) *Security Protocols 2004*. LNCS, vol. 3957, pp. 256–266. Springer, Heidelberg (2006). doi:10.1007/11861386\_30
13. Hempel, C.B., Oppenheim, P.: *Studies in the logic of explanation*. *Philos. Sci.* **15**(2), 135–175 (1948)
14. Henshaw, M., et al.: *The Systems of systems engineering strategic research agenda*. Document Number: TAREA-PU-WP5-R-LU-26. Issue 1, Loughborough University, United Kingdom 3, 17 June 2013
15. Hoeflberger, O.: *Report on the AMADEOS Workshop on Emergence in Cyber Physical-Systems of Systems*. Vienna University of Technology, May 2016
16. Holland, J.H.: *Emergence, from Chaos to Order*. Oxford University Press, New York (1998)
17. Huberman, B.A., Hogg, T.: *The Ecology of Computation*. *Studies in Computer Science and Artificial Intelligence*, vol. 2, pp. 73–115. North Holland, Amsterdam (1988)
18. Jamshidi, M.: *Systems of Systems Engineering*. Wiley, Hoboken (2009)
19. Keating, C.H.: *Research foundations for systems of systems engineering*. In: *Proceedings of the International Conference on Systems, Man and Cybernetics*, vol. 3, pp. 2720–2725 (2005)
20. Kim, J.: *Emergence: core ideas and issues*, 9 August 2006. [http://cs.calstatela.edu/~wiki/images/b/b1/Emergence-Coreideas\\_and\\_issues.pdf](http://cs.calstatela.edu/~wiki/images/b/b1/Emergence-Coreideas_and_issues.pdf)
21. Koestler, A.: *The Ghost in the Machine*. Hutchinson, London (1967)
22. Kopetz, H.: *Real-time Systems-Design Principles for Distributed Embedded Applications*. Springer, Heidelberg (2011)
23. Kopetz, H.: *A conceptual model for the information transfer in systems of systems*. In: *Proceedings of the 17th ISORC*, pp. 17–24. IEEE Press (2014)
24. Kopetz, H., et al.: *Direct versus stigmergic information flow in systems-of-systems*. In: *Proceedings of SoSE 2015*, pp. 36–41. IEEE Press (2015)

25. McLaughlin, B., Bennet, K.: Supervenience. *Stanford Encyclopedia of Philosophy*, Stanford (2011)
26. Mogul, J.: Emergent (Mis)behavior vs. Complex Software Systems. In: *Proceedings of EuroSys*, pp. 293–304 (2006)
27. O'Connor, T.: Emergent Properties. *Stanford Encyclopedia of Philosophy*, Stanford (2012)
28. Parunak, H.V., VanderBok, R.S.: Managing emergent behavior in distributed control systems. In: *Proceedings of ISA Tech 1997*, Anaheim, CA (1997)
29. Pattee, H.H.: Causation, control, and the evolution of complexity. In: Anderson, P.B., et al. (eds.) *From: Downward Causation: Mind, Bodies, Matter*, pp. 63–77. Aarhus University Press, Aarhus (2000)
30. Pattee, H.H.: The physical basis and origin of hierarchical control. In: Pattee, H.H. (ed.) *Laws, Language and Life*, vol. 7, pp. 91–110. Springer, New York (2012)
31. Ronal, E.M.A., et al.: Design, observation, surprise! a test of emergence. *Artif. Life* **5**, 225–239 (1999)
32. Simon, H.: The architecture of complexity. In: Simon, H. (ed.) *The Science of the Artificial*. MIT Press, Cambridge (1969)
33. Sparso, J., Furber, S.: *Principles of Asynchronous Circuit Design*. Kluwer Publisher, Dordrecht (2002)
34. Stephan, A.: Emergence—a systematic view on its historical facets. In: Beckerman, E., et al. (eds.) *Emergence or Reduction?*, pp. 25–48. Walter de Gruyter, Berlin (1992)

**Open Access** This chapter is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, duplication, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the work's Creative Commons license, unless indicated otherwise in the credit line; if such material is not included in the work's Creative Commons license and the respective action is not permitted by statutory regulation, users will need to obtain permission from the license holder to duplicate, adapt or reproduce the material.

