Emergence
- *from the Point of View of Industrial Software*

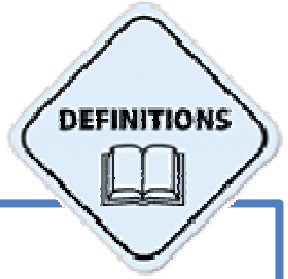
Frank J. Furrer
Prof h.c. TUD, Dr. sc. techn. ETH-Zürich

TU Dresden: Workshop «**Emergence**»
25. August 2016

V1.0 / 21.08.2016

http://www.fotocommunity.de

Emergence ?



Manifestation of a system's properties that *cannot* be anticipated *in all cases* from the properties of its constituent components or parts

Example: Emergent property: „flying“

http://www.mskgmbh.com

Constituent systems (CS) of an aircraft:

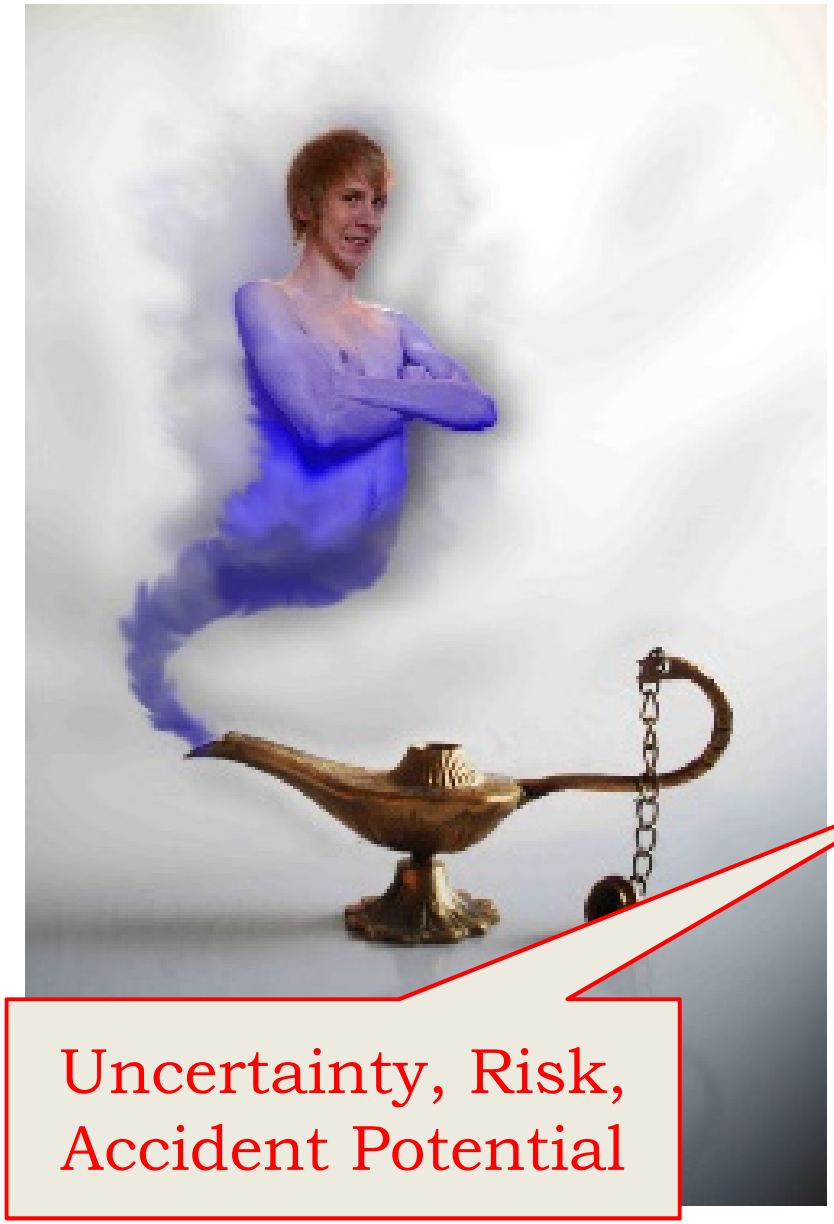
- engines
- body
- wings
- cockpit
- etc.

… none of the constituent systems is able to fly !

Assemble the essential constituent systems:

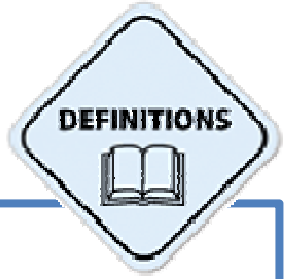
Emerging property: the assembly (= airplane) is able to **fly** !

http://en.wikipedia.org/wiki/Airbus_A320_family



http://www.fotocommunity.de

Emergence ?



Manifestation of a system's properties that ***cannot*** be anticipated in ***all*** cases from the properties of its constituent components or parts

Uncertainty, Risk, Accident Potential

Example: Emergent Property: **Interface Failure**

http://avioners.net/2011/03/airbus-a319-approach-to-landing.html

Constituent systems:

- Airplane (DC-8)
- Airport (Runway)

October 8, 1979: Swissair Flight 316 overran the Athens runway – 14 deaths

Cause: „Interface“ between the runway and the airplane

- Landing when braking action is less then good
- Crew mistakes

Research View

Biological View

SW-Eng View

Cybernetic View

Industrial SW View

System View

Objectives of this Presentation:

- Present the Industrial Software View of Emergence
- Show the potential Impact of Emergence on Industrial (= Cyber-Physical) Systems
- Investigate the Possibilities to manage Emergence
- Discuss open Questions

Content:

1. Definitions & Foundation
2. Industrial Software
3. Emergence
4. Impact Analysis
5. Managing Emergence
6. Conclusions

Definitions & Foundation

For emergent behaviour you need a *system of systems*

System-of-Systems characteristics:

1. Operational Independence of the Elements
2. Managerial Independence of the Elements
3. Evolutionary Development

Emergent Behaviour

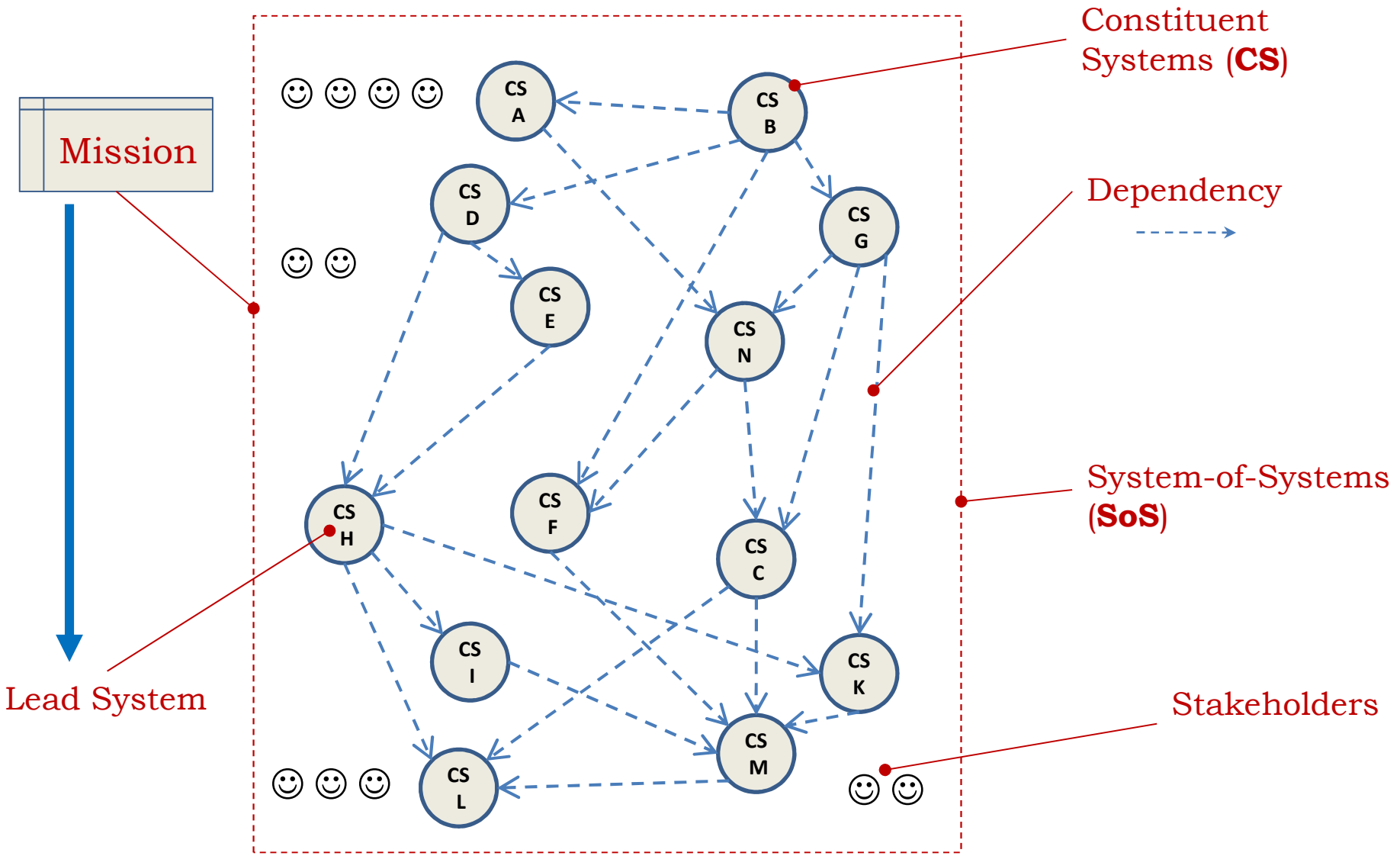
5. Geographic Distribution

[5 Maier criteria, 1998]

Total = more than the sum of its parts

Emerging behavior is the consequence of the *interactions* of the constituent components or parts of the system of systems

SoS (System-of-Systems) Terminology



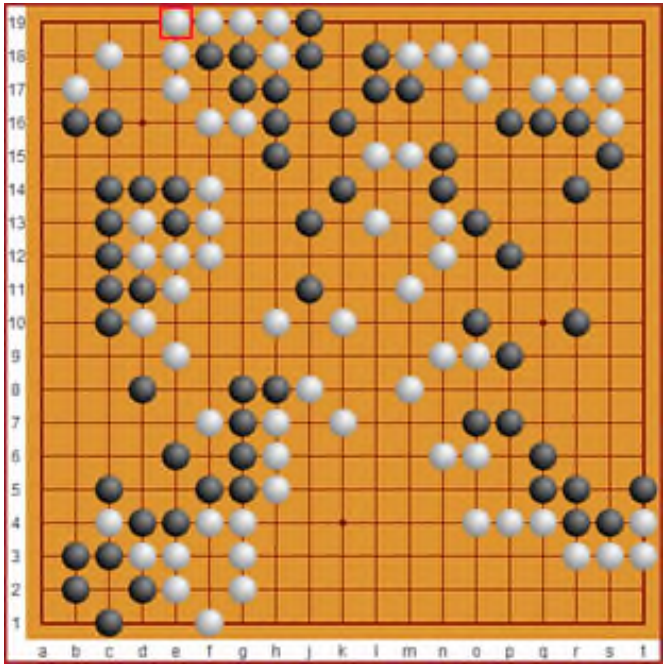
Emergent Behaviour (Example 1/3)

https://fr.wikipedia.org

«GO» is a board game which was invented before 2`500 years in China.

Board: 19 x 19 lines, infinite number of white and black stones.

Objective: gain as much terrain as possible

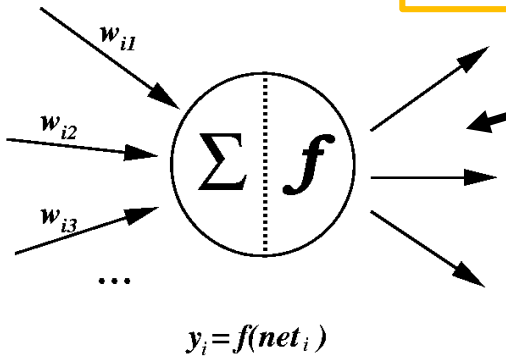


http://www.brettspielnetz.de

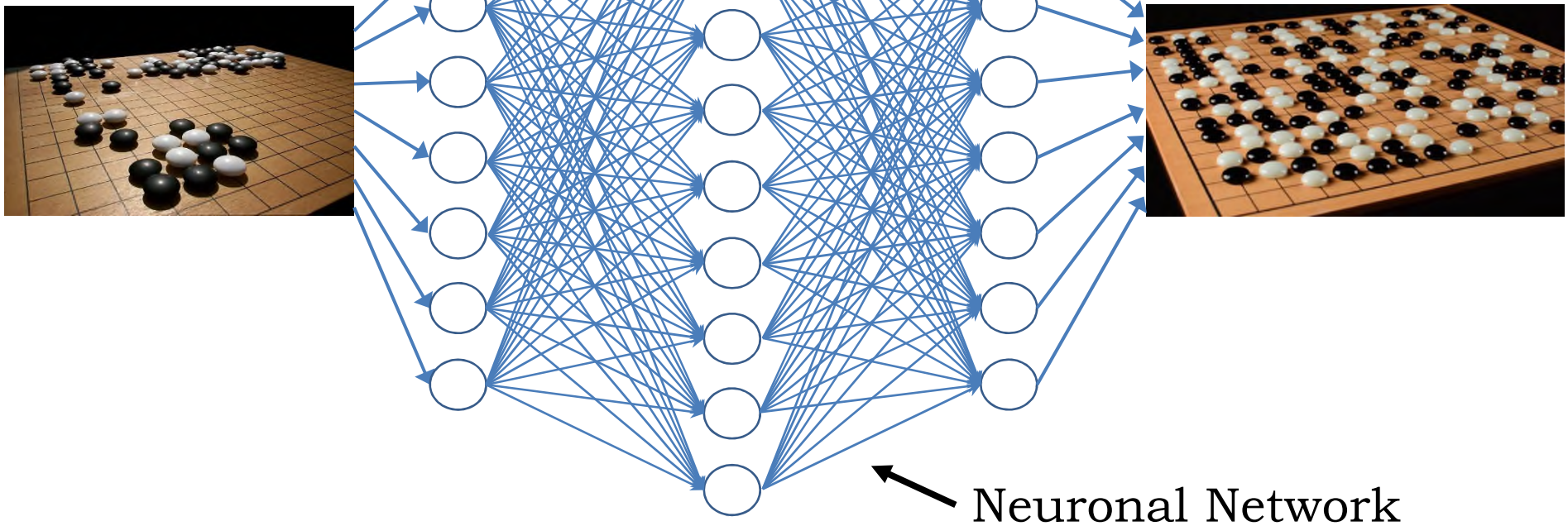
Number of possible situations on the GO-board: ~ **$4{,}63 \times 10^{170}$**

- Chess: ~10^{43}
- Number of atoms in the universe: ~10^{80}

Emergent Behaviour (Example 2/3)



Artificial Neuron



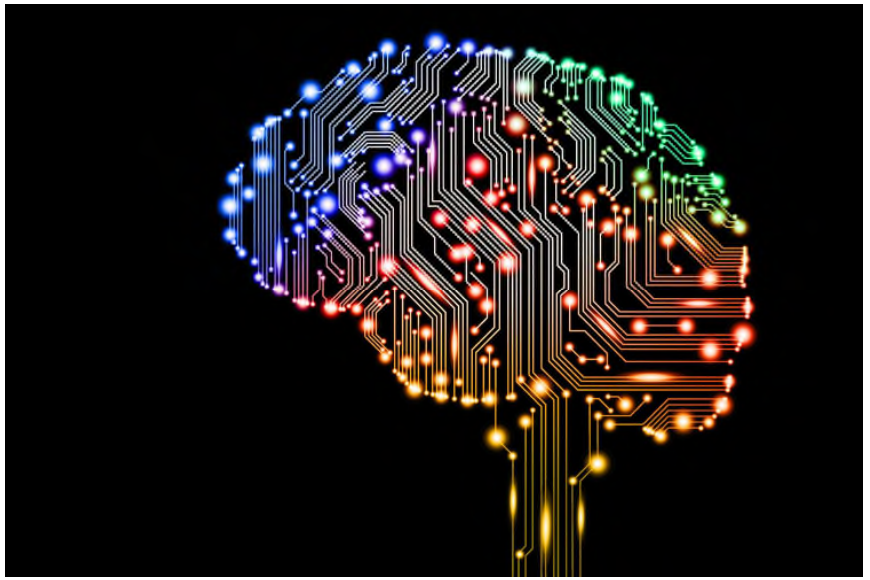
Neuronal Network

Emergent Behaviour (Example 3/3)

http://www.watson.ch

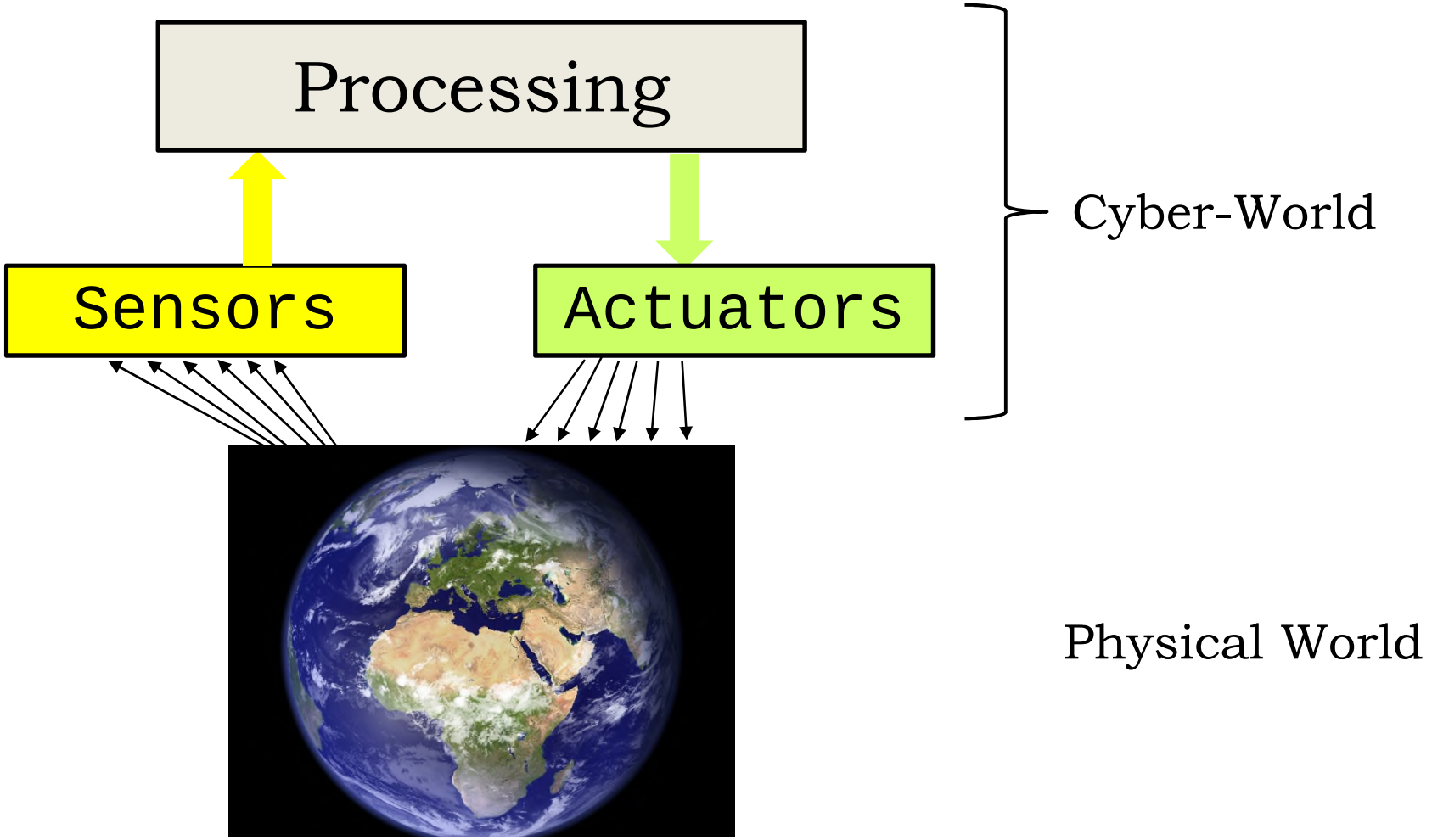
March 2016: The AI-program «AlphaGO» wins against the multiple and current world champion Lee Sedol (4:1)

Impressive/frightenining: «AlphaGO» was ***not*** programmed, but was a learning program [Deep Learning] ⇐ **Emergence!**



http://www.digitaltrends.com

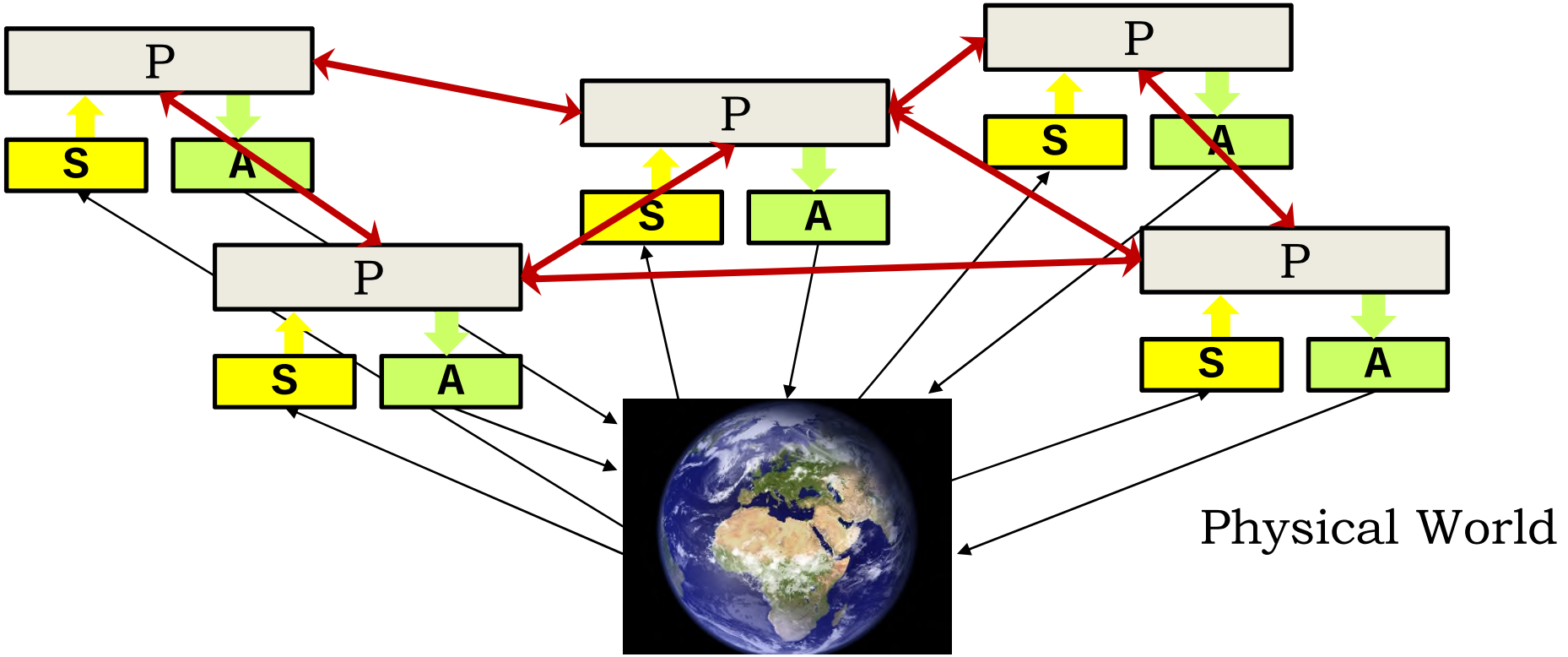
Cyber-Physical System



http://www.dangerouscreation.com

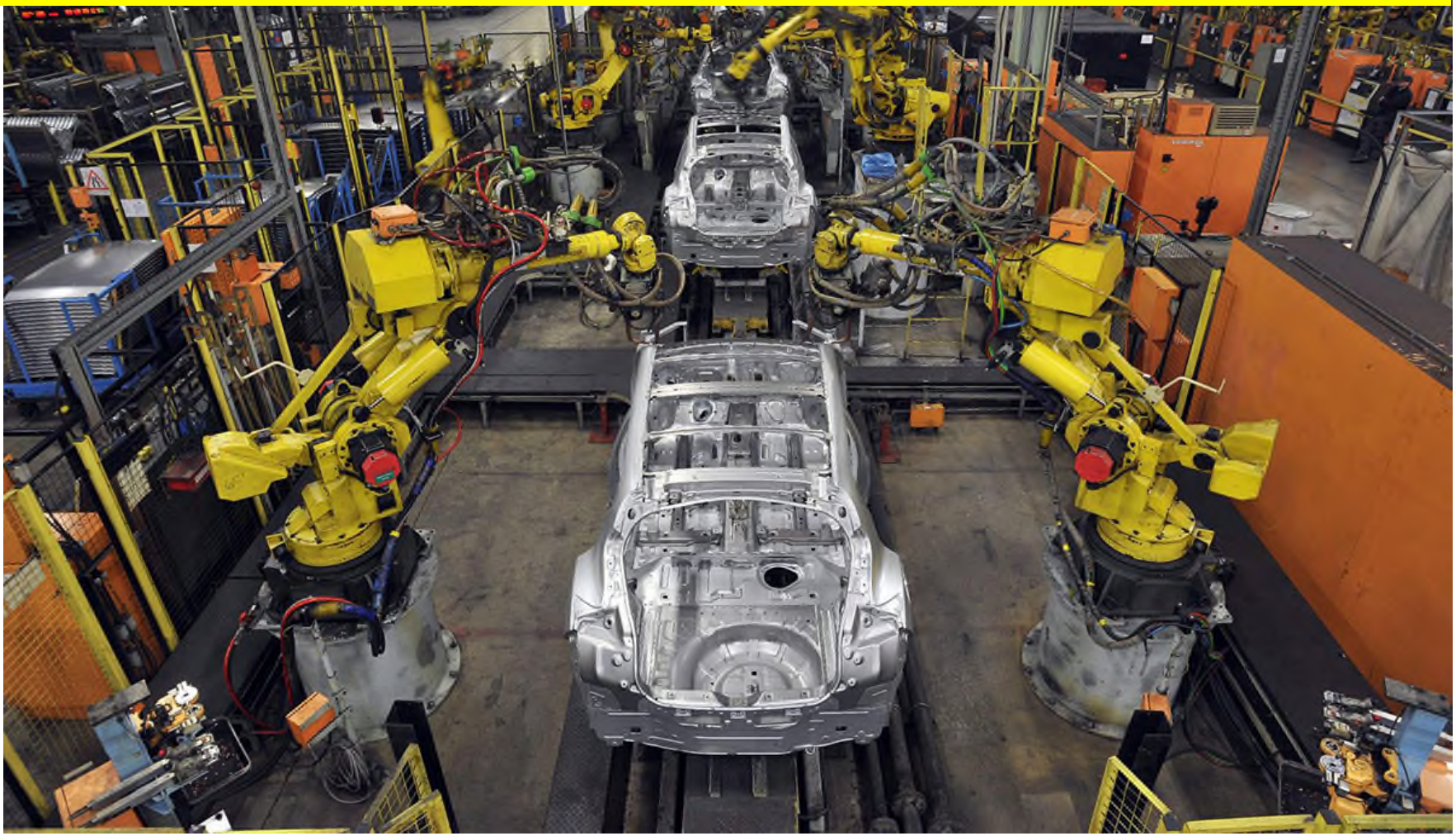
Cyber-Physical System of Systems (CPSoS)

A cyber-physical system (CPS) consists of a collection of computing devices communicating with one another and interacting with the physical world via sensors and actuators in a feedback loop



http://www.dangerouscreation.com

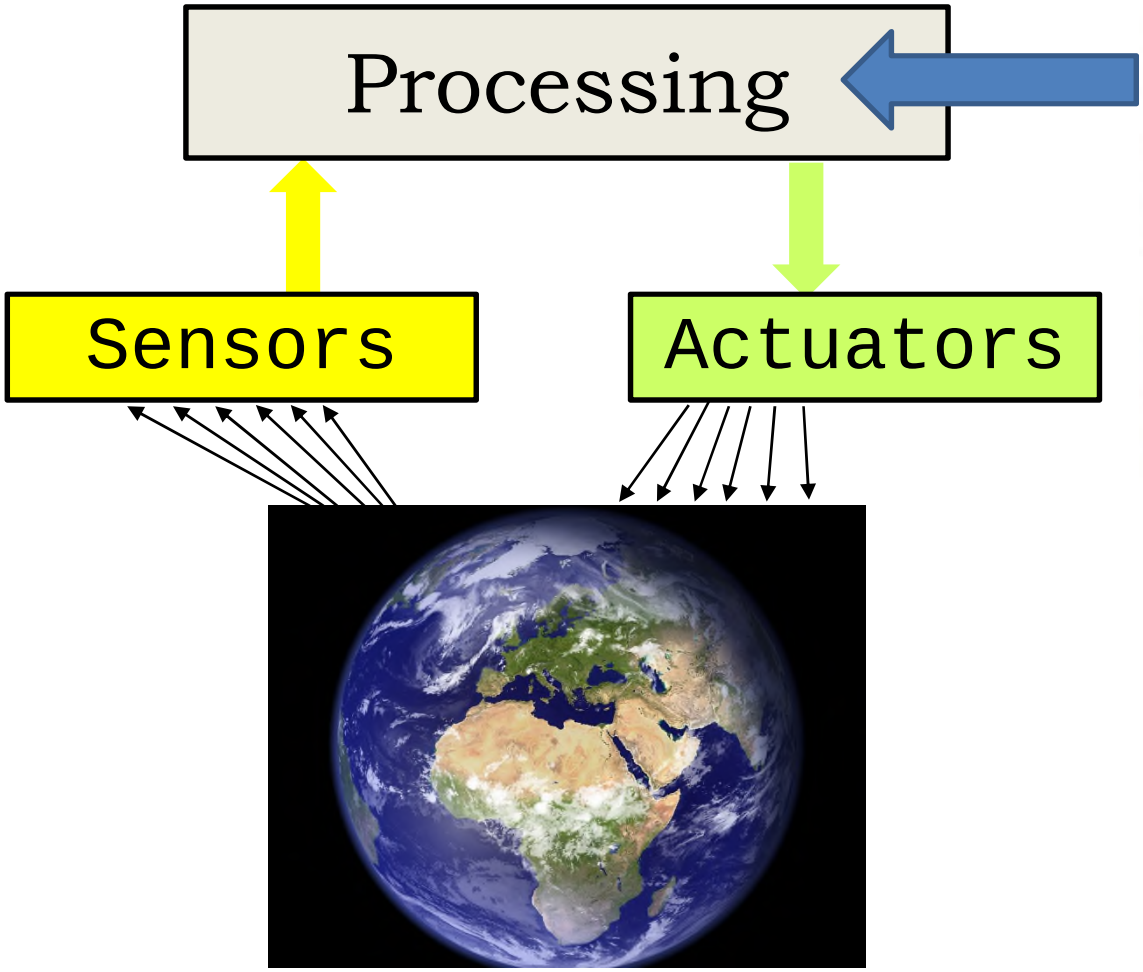
Example of a Cyber-Physical System of Systems (CPSoS): **Collaborating Robots**



http://www.hypermiler.co.uk

Industrial Software

Industrial Software




```
1 class Graph {
  private Set<Arc> arcs = new HashSet<Arc>();
3 private Set<String> nodes =
                          new HashSet<String>();
5 public void addNode(String id) {
    nodes.insert(id); }
7 public void addArc(String from, String to) {
    arcs.insert(new Arc(from,to)); }
9 public Set<Colored> compute3Coloring() {
   Set<Colored> res = new HashSet<Colored>();
11  <# in = arcs::arc,nodes::node out = res::col
      col(X,red) v col(X,green)
13                v col(X,blue) :- node(X).
      :- col(X,C), col(Y,C), arc(X,Y).
15  #>
    if_no_answerset { res = null; }
17  return res; }
  }
19 public class Arc {
    public String start;  public String end;  }
21 public class Colored {
    public String node;   public String color;}
```

Software

http://www.dangerouscreation.com

Industrial Software

Software

```
1 class Graph {
  private Set<Arc> arcs = new HashSet<Arc>();
3 private Set<String> nodes =
                          new HashSet<String>();
5 public void addNode(String id) {
    nodes.insert(id); }
7 public void addArc(String from, String to) {
    arcs.insert(new Arc(from,to)); }
9 public Set<Colored> compute3Coloring() {
   Set<Colored> res = new HashSet<Colored>();
11  <# in = arcs::arc,nodes::node out = res::col
      col(X,red) v col(X,green)
13                v col(X,blue) :- node(X).
      :- col(X,C), col(Y,C), arc(X,Y).
15  #>
    if_no_answerset { res = null; }
17  return res; }
  }
19 public class Arc {
    public String start;  public String end;  }
21 public class Colored {
    public String node;   public String color;}
```

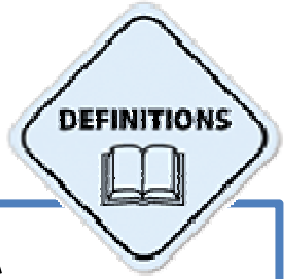
- Mission-critical
- Safety-critical
- Real-time
- Security-critical
- etc.

Characteristics of Industrial Software

Emergence

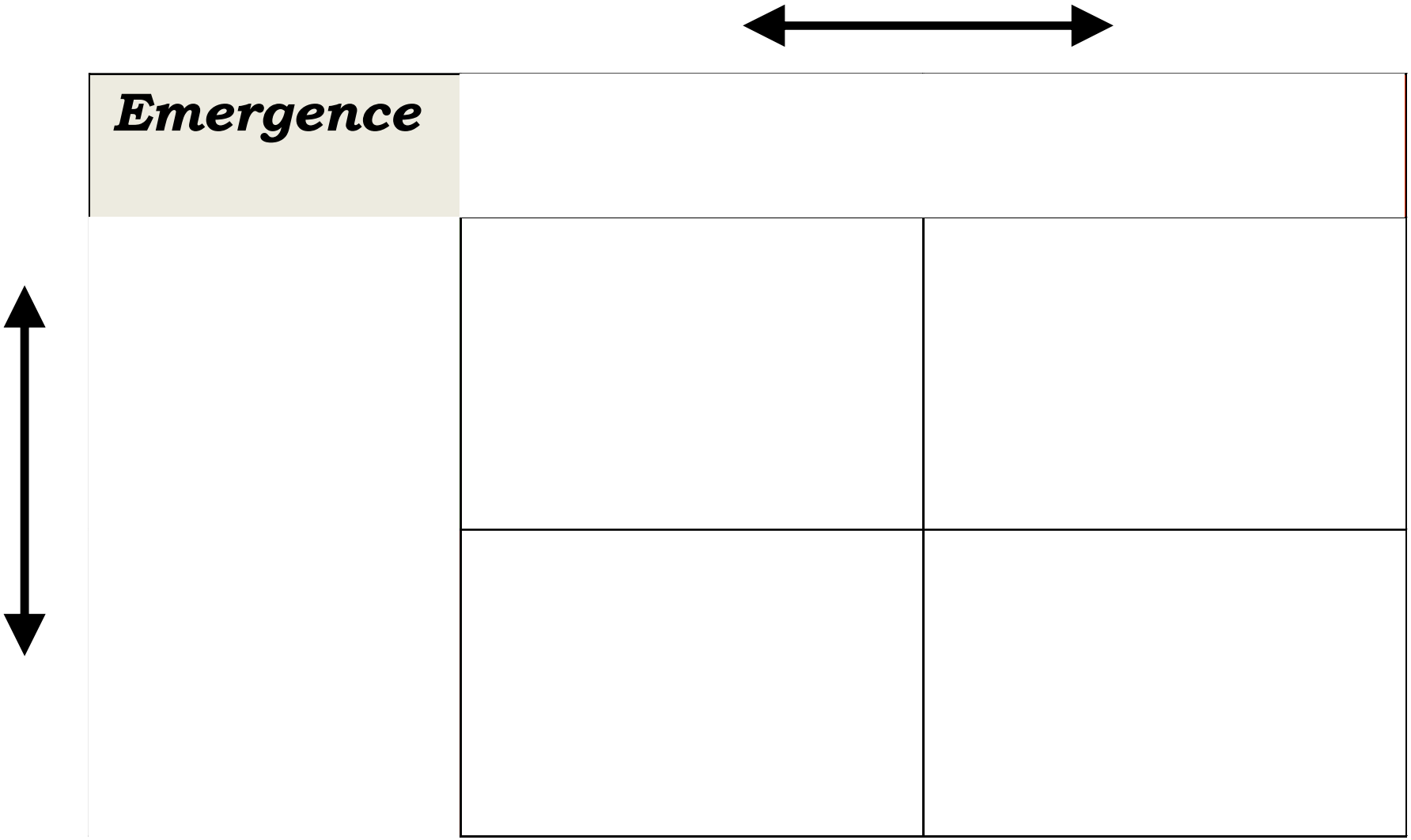
Emergence

http://www.fotocommunity.de

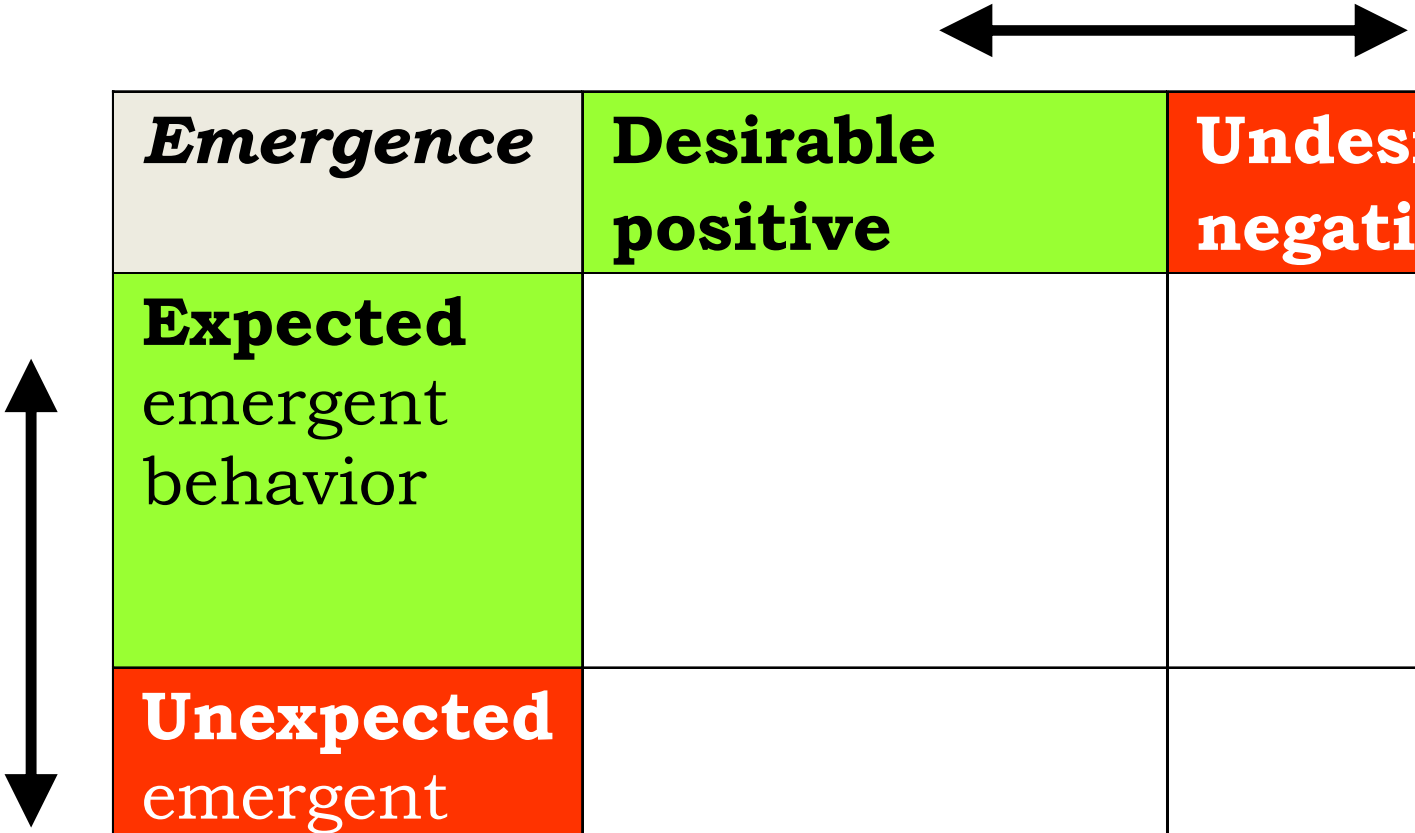


Manifestation of a system's properties that ***cannot*** be anticipated in ***all*** cases from the properties of its constituent components or parts

Emergent Behaviour Classification

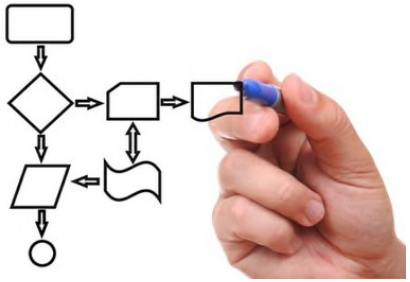


Emergent Behaviour Classification

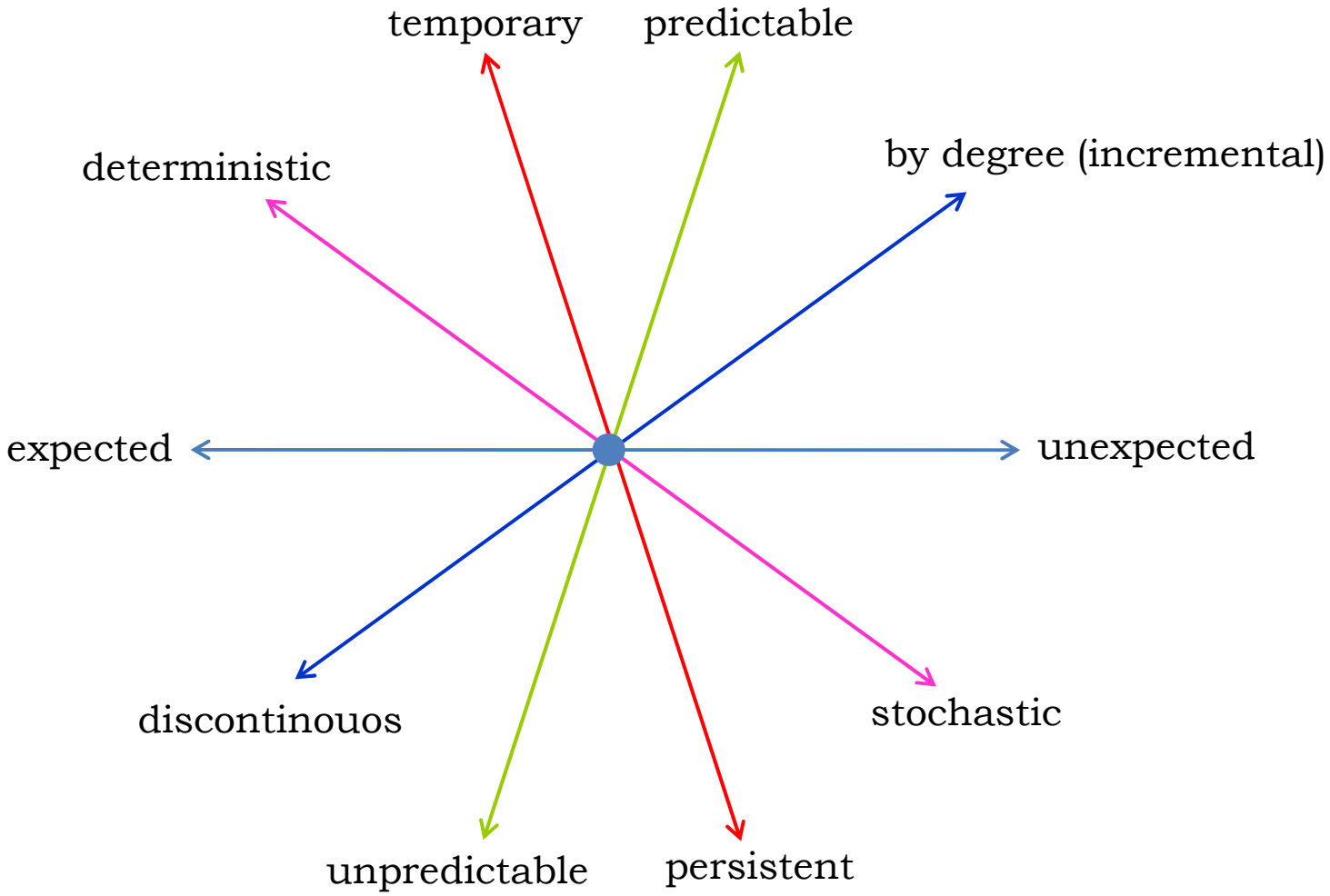


Emergence	**Desirable positive**	**Undesirable negative**
Expected emergent behavior		
Unexpected emergent behavior		

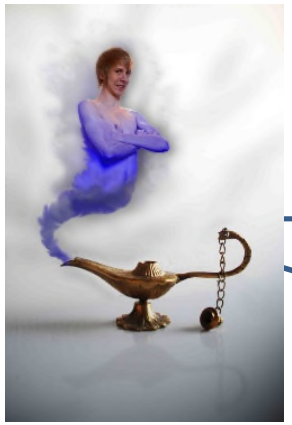
SoS Emergent Behaviour Classification

Emergence	Desirable/positive	**Undesirable/negative**
Expected emergent behavior	Reason for building the SoS (**SoS objective**) 	**Mitigate** by appropriate design measures, such as threat/risk analysis and countermeasures 
Unexpected emergent behavior	Sometimes (however, quite rarely) an SoS shows unexpected, **beneficial behaviour** 	Unexpected & undesirable negative emergent behavior is one of the **critical risks** of most SoS 

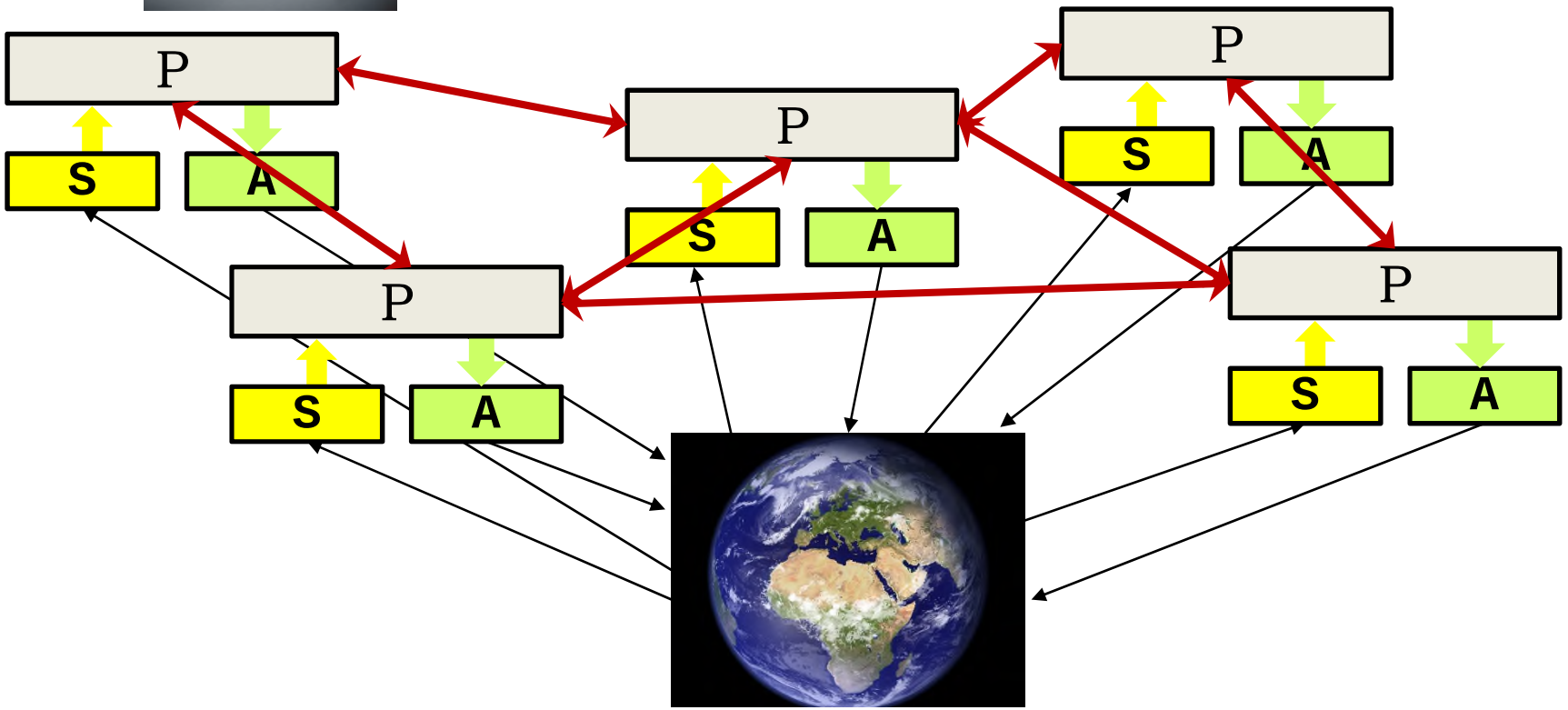
SoS Emergent Behaviour Classification

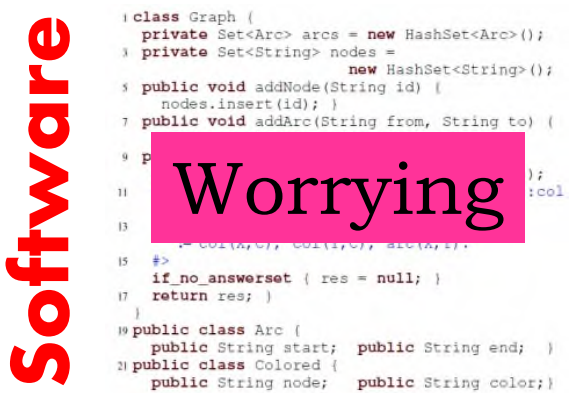
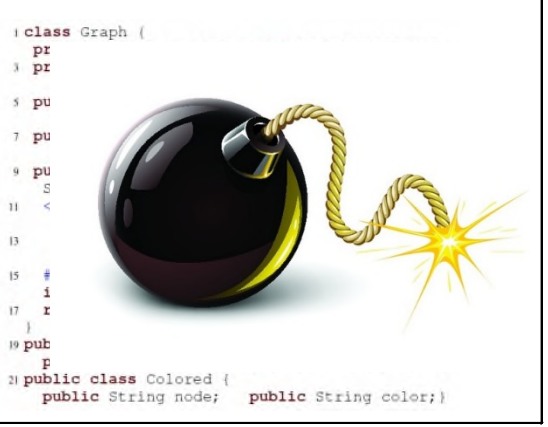


Impact Analysis



Which Impact has Emergence on our CPSoS?



Emergence	**Desirable positive**	**Undesirable negative**
Expected emergent behavior	Software  Design Goal: **Attain**	Software  Design Goal: **Avoid**
Unexpected emergent behavior	Software  Worrying	Software 

Emergence	**Desirable positive**	**Undesirable negative**
Expected emergent behavior		 Missing Care!
Unexpected emergent behavior		 **Accidents**

Managing Emergence

Managing Emergence in Industrial Software

Emergence	**Desirable positive**	**Undesirable negative**
Expected emergent behavior	√	√
Unexpected emergent behavior	✗	

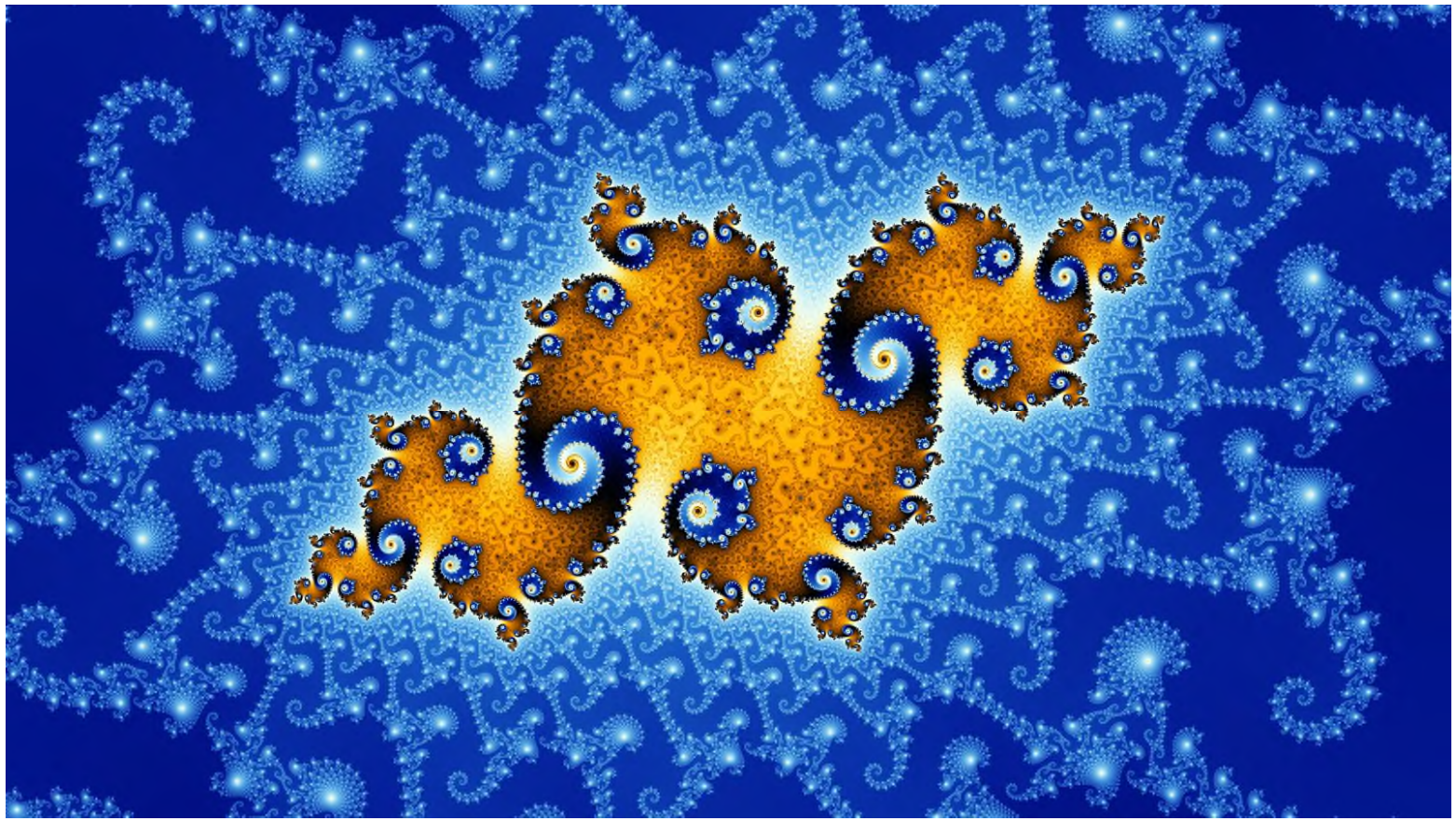
Vital Questions for SoS-Engineering:

1) How can we *avoid* **unexpected** emergent behaviour?

2) How can we *avoid* **undesirable** emergent behaviour?

3) How can we *prove* that there is no **unexpected** emergent behaviour in our system?

4) How can we *prove* that there is no **undesirable** emergent behaviour in our system?



http://seryzone.deviantart.com

Emergence

is today an active research challenge in various fields

http://www.darmarit.de

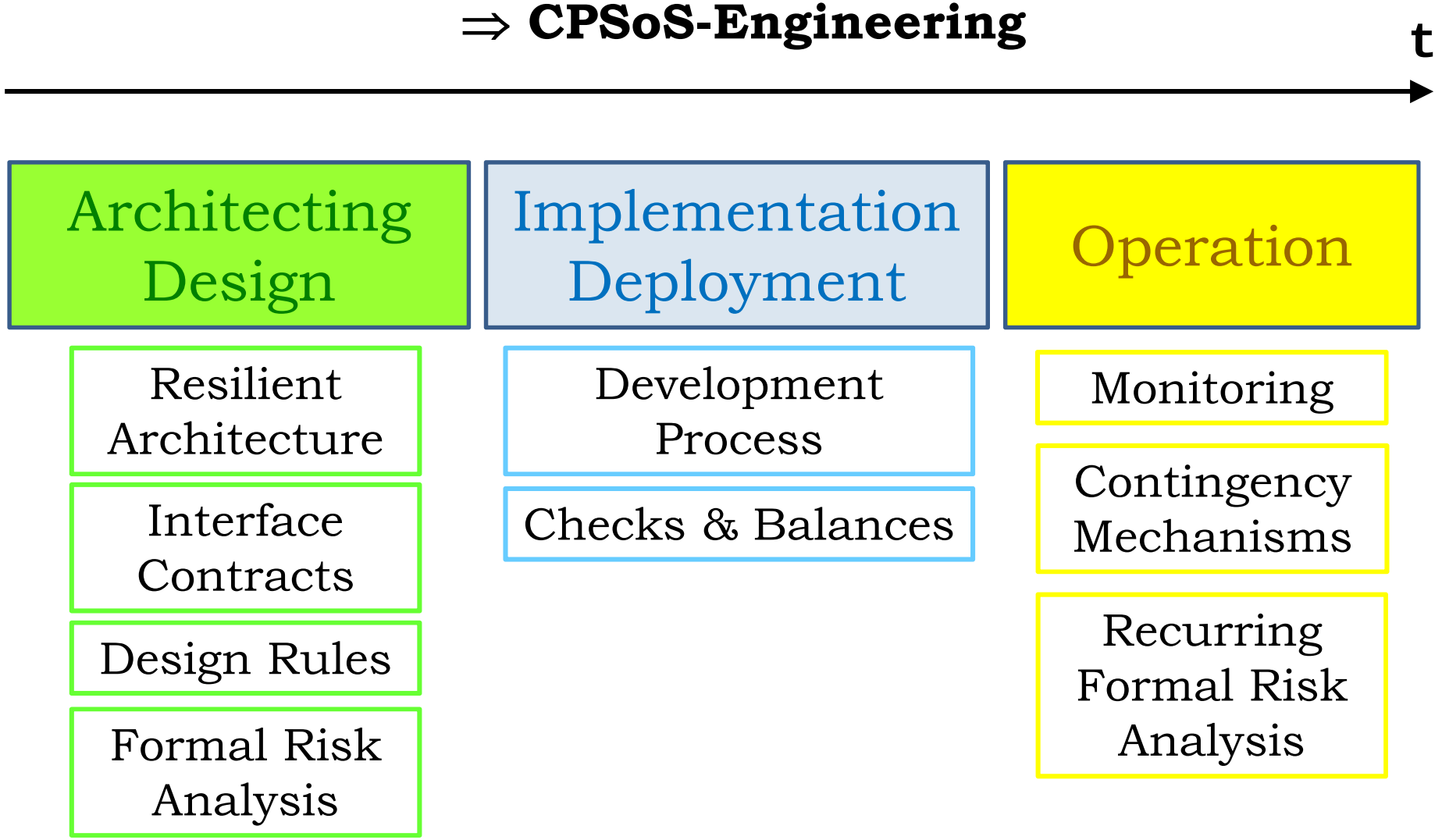
Emergenent Properties:

What can we do in Industrial Software?

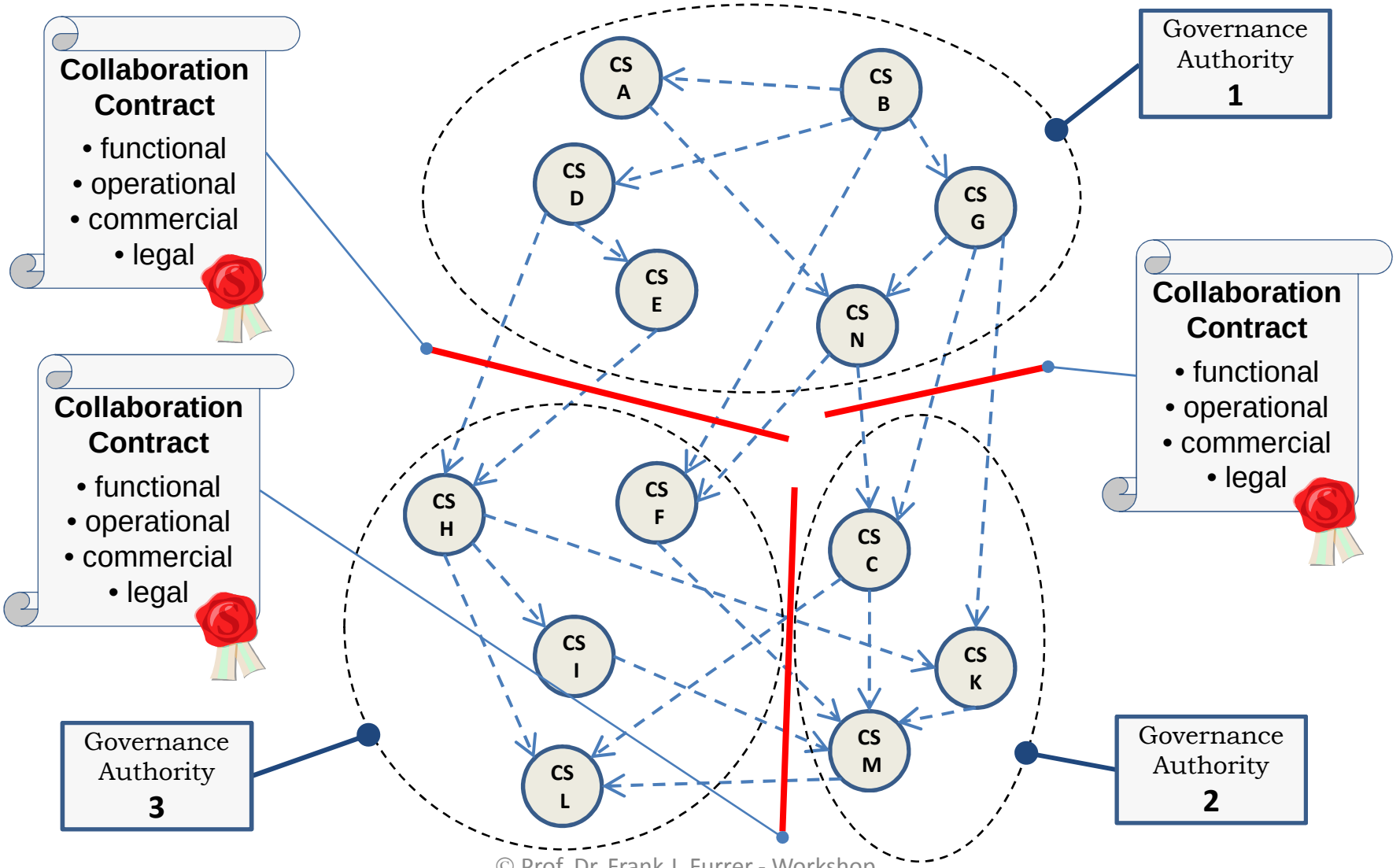
New Field: Systems-of-Systems Engineering (SoSE)

Managing Emergence in Industrial Software

⇒ **CPSoS-Engineering**



Interface Contracts



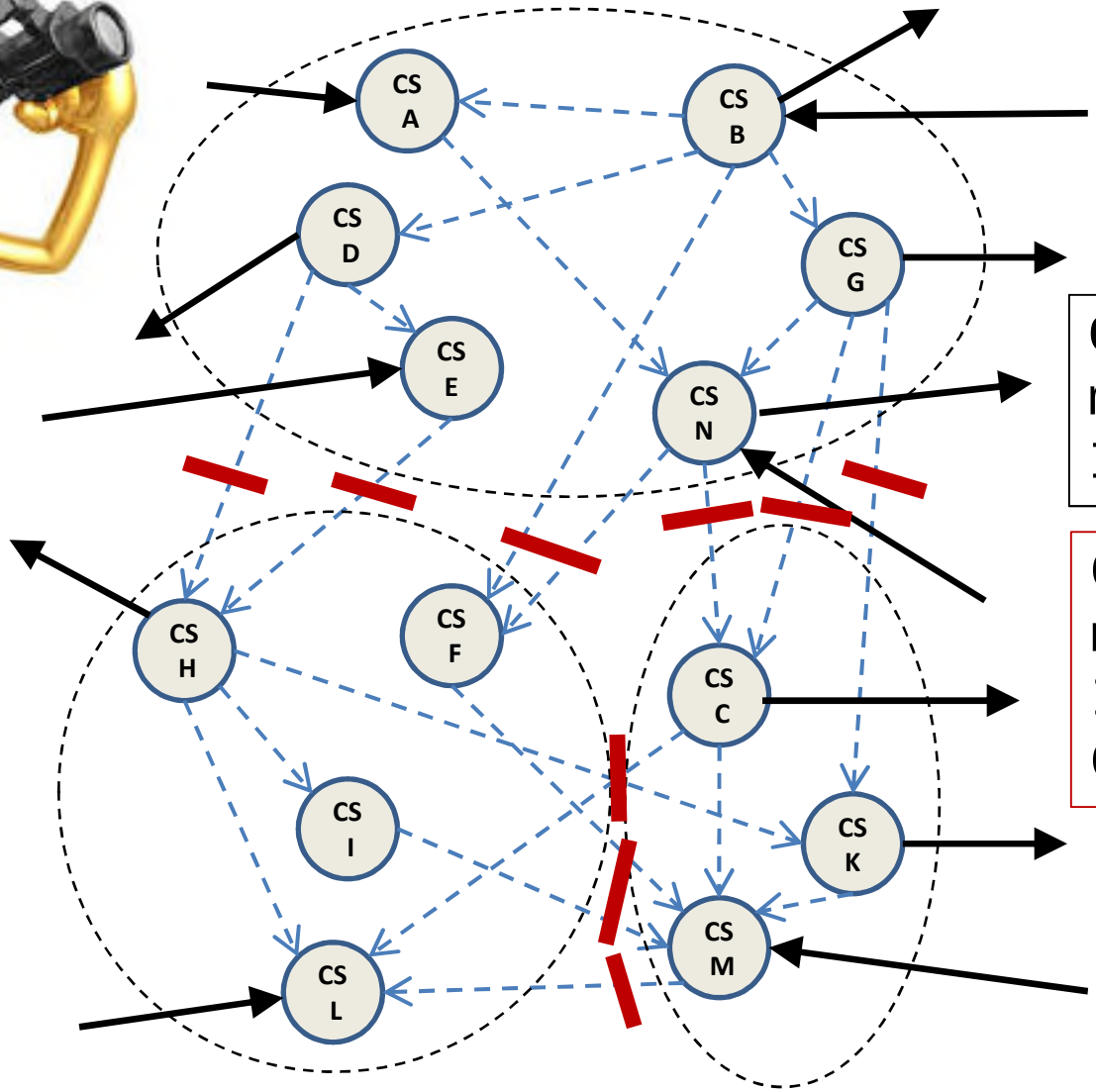
www.123rf.com

Execute a formal **risk assessment**
(various proven methodologies available)

Risk ⇒ Assessment ⇒ Mitigation (Software Development)

Monitoring

http://www.gograph.com



Continuously monitor Inputs/Outputs

Continuously monitor Interface Contracts

Conclusions

Conclusions

Emergence (emergent behaviour) can be a serious **risk** for Industrial Software

Emergence is **not** yet well understood

Potential Emergence in Industrial Software needs to be **managed**

Requires an effective **SoSE-Process**

Research in

Emergence Management

for Industrial Software

is an interesting and fruitful Topic

(Master, Diploma, PhD)

Thank You and Questions Please