Lecture

Organic Computing II

Summer term 2020

Chapter 2: Self-organised Order

Lecturer: Jörg Hähner

Slides have been created in cooperation with Prof. Dr. Sven Tomforde (Uni Kassel),
Prof. Dr. Müller-Schloer (Uni Hannover) and Prof. Dr. Jörg Hähner (Uni Augsburg)

Content

- A first example: water temples in Bali
- A second example: ants
- Emergence
- Term definition
- Quantification of emergence
- A refined approach to emergence quantification
- Conclusion and further readings

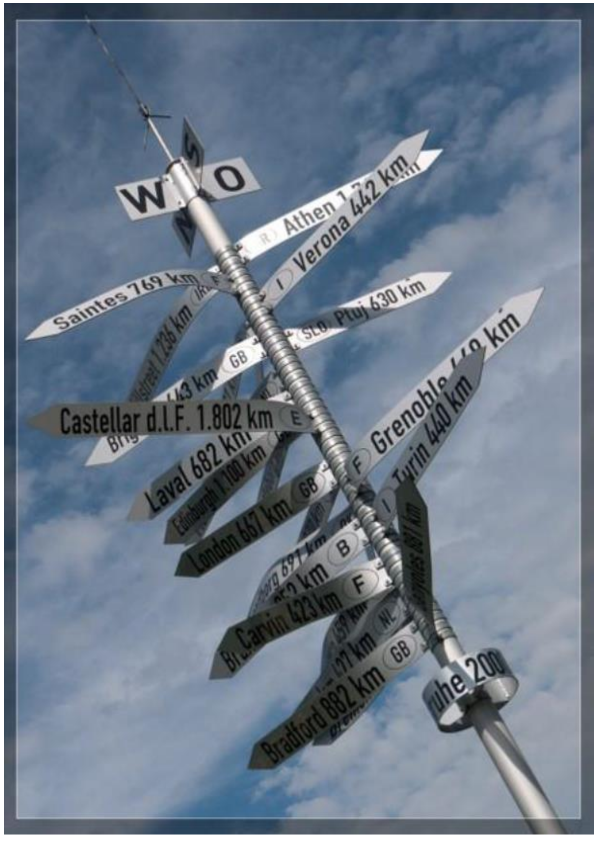
Goals

Students should be able to:

- Explain the relation between self-organisation and emergence.
- Briefly summarise the term emergence.
- Give examples for emergent phenomena, e.g. in nature.
- Quantify emergence in technical systems based on discrete attributes.
- Outline how emergence detection is done for systems with continuous attributes.

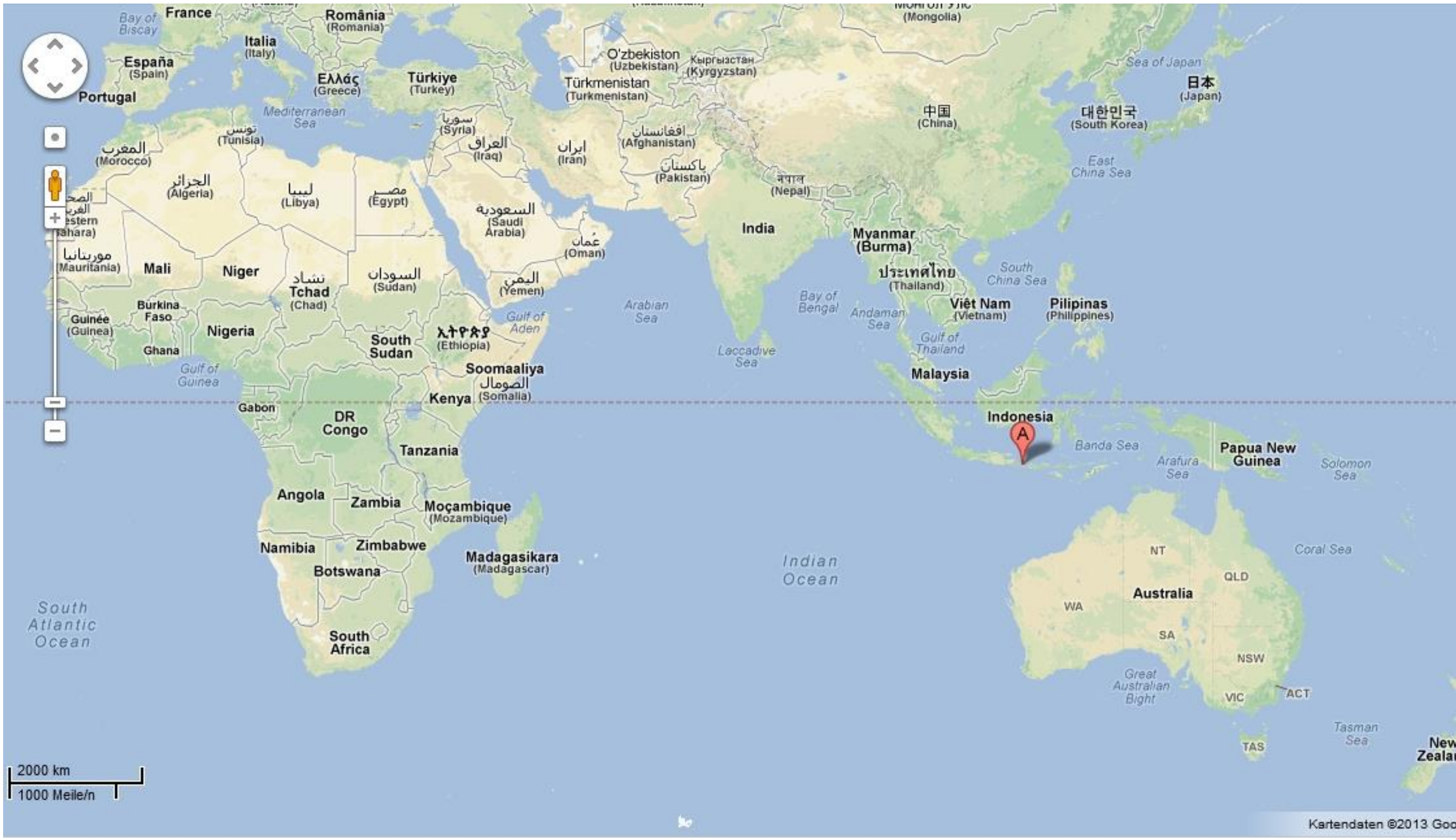
Agenda

- A first example: water temples in Bali
- A second example: ants
- Emergence
- Term definition
- Quantification of emergence
- A refined approach to emergence quantification
- Conclusion and further readings



A journey to Bali

ORGANIC
COMPUTING



Water temples in Bali

Water temples in Bali (2)

Watering system for the cultivation of rice

- Main factors:
  - Water circulation
  - Alternation between dry and wet periods
- Objectives
  - PH-values
  - Activity of micro-organisms
  - Distribution of mineral nutrient
  - Herbicide
  - Pest control (for large areas)
  - Stabilisation of temperature
- Problem: Synchronous watering leads to peak demand of water!

Water temples in Bali (4)

Problem for each farmer:

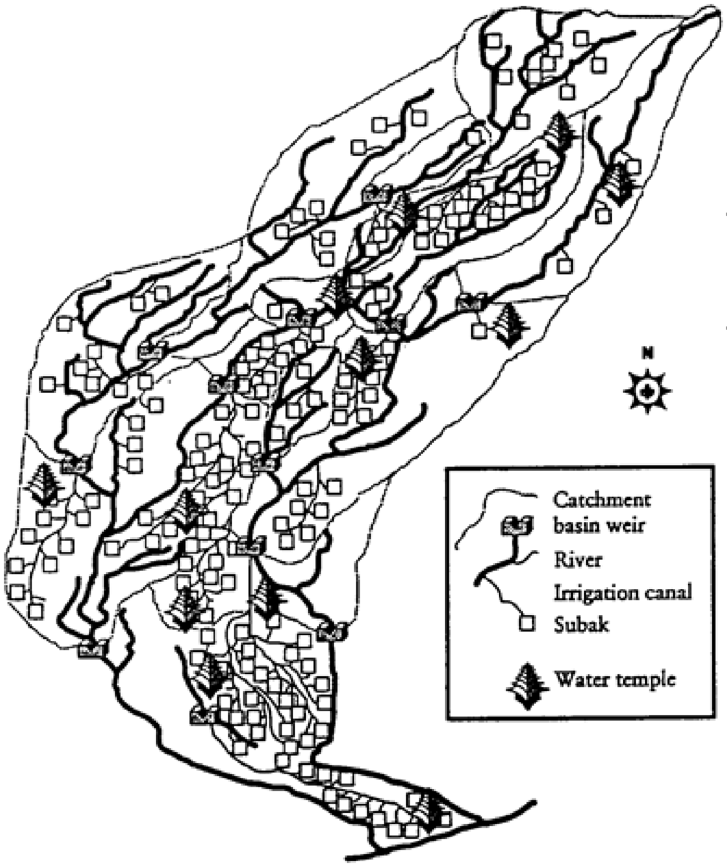
- Determine cultivation sequence

Goal:

- Maximise crop

Attributes of cultivation sequence:

- Phases of cropping (date)
- Cultivar (kind of plant)
- Watering
- Drying



The Oos and Petanu rivers in south-central Bali (not to scale).[16]

Source: Lansing and Kremer

Water temples in Bali (5)

How to determine the optimal sequence?

- Trial and error vs. planning
- Coordination: global or local?
- Is the solution suitable for the local problem?
- Is the solution adaptable?

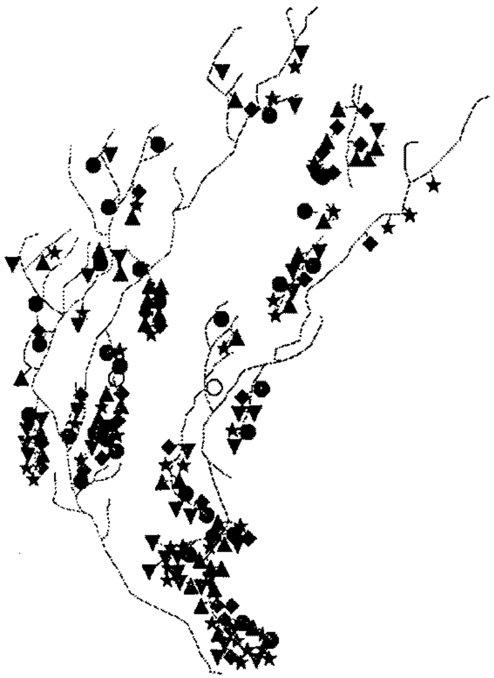
Hypothesis:

- Coordination algorithm
- Synchronous, local, like the best neighbour
- Co-adaptation

Verification of hypothesis: simulation

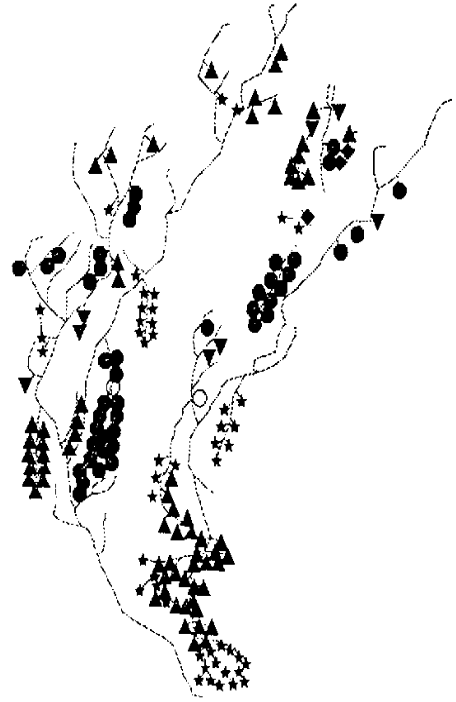
- Crop is modelled as a function of cultivation sequence and environment.
- Start: randomised initialisation

Water temples in Bali (6)



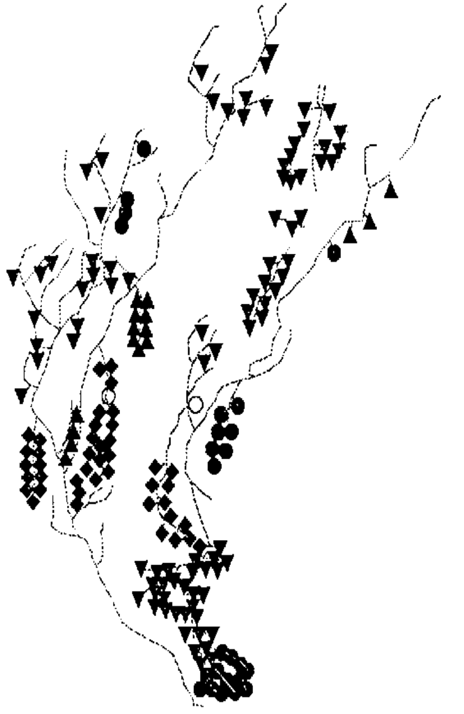
Randomised distribution at start-up

4.9 tons/ha



Simulated pattern after co-adaptation cycles

8.6 tons/ha



Traditional system of cave temples at Bali

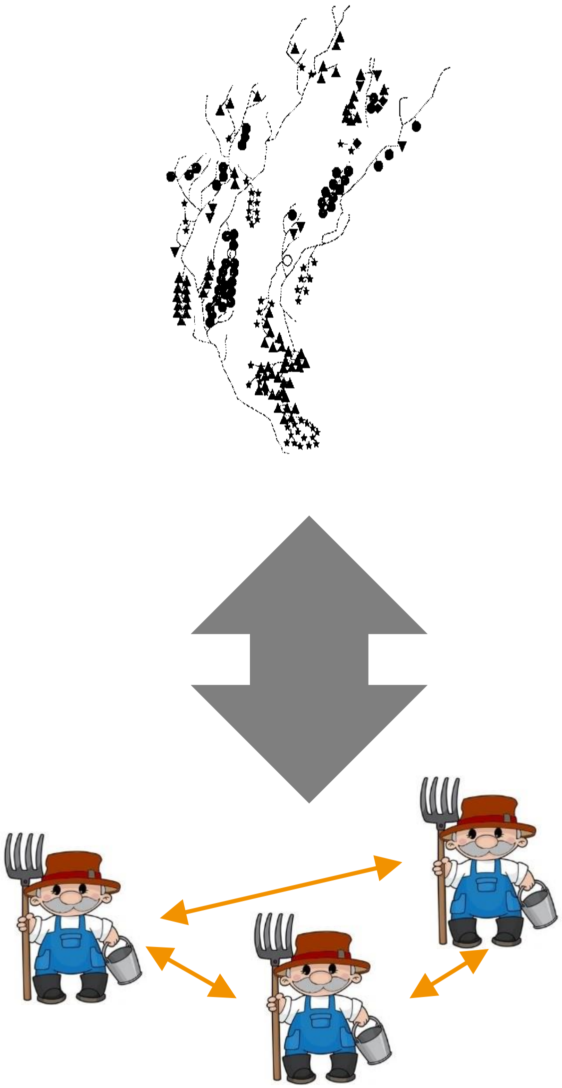
Cultivation sequences

Conclusions to be drawn from the water temple networks:

- Contains locally behaving and self-motivated farmers (nodes, agents).
- Cooperation leads to globally optimal (or „good enough“) patterns.
  → An emergent effect.
- Bottom-up evolved problem-solving networks are adaptive: react to changing environmental conditions (e.g. reduced rainfall).
- Success of the networks depends on: (1) ability of local nodes to collect local information and (2) react to it locally.
- There is no central authority needed, the system is decentralised.
- There is no external authority needed, the system is self-organised.
- Co-adaptation requires large populations of interacting agents.
- Agents decide on their own but in close and regular coordination with their neighbours. They are semi-autonomous.
- The system learns and adapts in evolutionary cycles.
- Evolutionary steps are subject to random variations.

Insight: self-organised order

- A global pattern emerges
- System is structured
- Nobody is in charge
- Nobody has a global view
- Nothing is planned
- Distinguish between micro- and macro-level
  → actions at micro-level, effects at macro-level
  → “good enough” solutions at macro-level

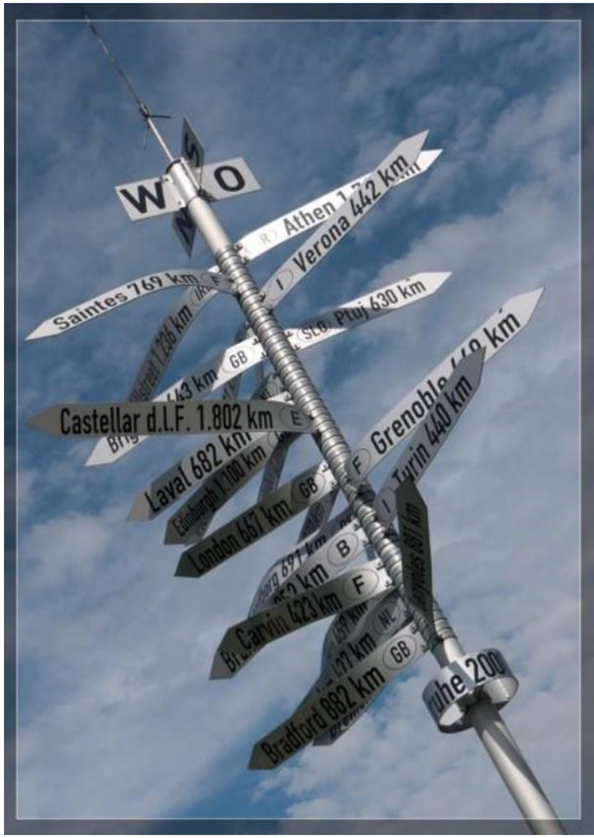


Macro-level:
- Global view
- Pattern
- Structure

Micro-level:
- Interaction
- Local view

Agenda

- A first example: water temples in Bali
- A second example: ants
- Emergence
- Term definition
- Quantification of emergence
- A refined approach to emergence quantification
- Conclusion and further readings



Ants

- Related to wasps and bees
- About 100 million years old
- Extremely successful:
  - About 30% of bio mass in the Amazonas are (with termites)
  - About 9% of bio mass world-wide
  - Comparable to bio mass of humans
- No or only limited vision
- Distribution of labour:
  - soldiers,
  - construction workers,
  - gardeners,
  - reproduction
  - …

Source: Ants at Work (1999) by Deborah Gordon

Abilities of ants

- Find shortest paths
- Build bridges
- Sort
- Efficient logistics
- Farming / food production
- Construction of complex structures
- Caring about brood and offspring
- Caring about useful entities
- Distribution of labour
- ...

Ants (3)

Building bridges

- Chains of ants clamping together
- To bridges gaps in path
- Other ants use these chains as path
- E.g. to better reach food
- Chains of ants to pull leaves down
- Alternative: use silk from larvae as materia

Ants (4)

Food production

- Ant species that cultivates fungi
- Ground and substrate is processed for fungi.
- "Gardens" are laid out and cultivated.
- Fungal spores are planted.
- Competing fungi (i.e. light or water) is eliminated.
  → Like weeding the garden!
- Fertilisation of chewed larval cases.

Distribution of labour

- Different types of ants in a colony:
  - Worker
  - Medium-sized ants
  - Small ants
  - Queen
  - Male ants

- Workers search for food (leaves) in the surroundings (up to hundreds of meters from nest).
- Workers organise ant trails for transport of food.
- Medium-sized ants carry leaves.

Ants (6)

Symbiosis

- Ants cannot digest cellulose.
- But fungus can!
  → Eat hyphal tips
- Ants provide leaves as breeding ground for fungus.

Gardens

- Sculptured with many furrows ("Furche") and cavities ("Höhle") for brood.
- System of lower passages to drain wet chambers.
- Other tunnels for temporary control
- Ants swab floor clean.
- If foreign fungus develops, it is removed.
- Garden lasts 3-4 weeks / in various stages
- Founding queen brings fungal spores from old colony.
- Up to 500,000 per colony.

Alternative: Some ant species (e.g. Lasius niger, common black ant in Britain) herd aphids ("ant cattle"), protecting and even constructing them shelters.

Ants (7)

Protection

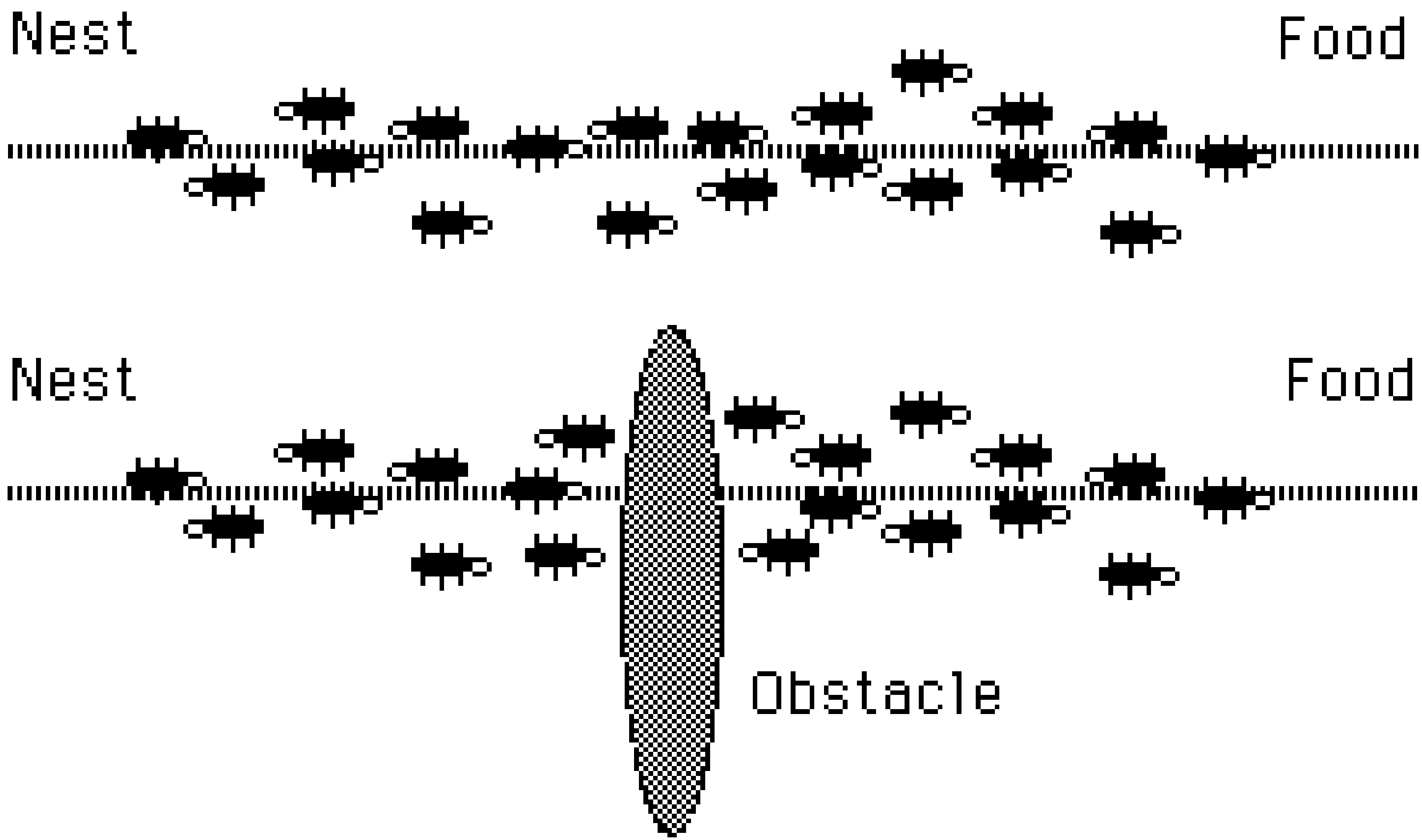
- Problem: large ants are victims of parasites.
- I.e. small flies try to deposit their eggs at head or neck of ants.
- Causes illness.
- Solution: small ants “ride“ piggy-back and chase attackers.

ORGANIC COMPUTING

Finding shortest paths:

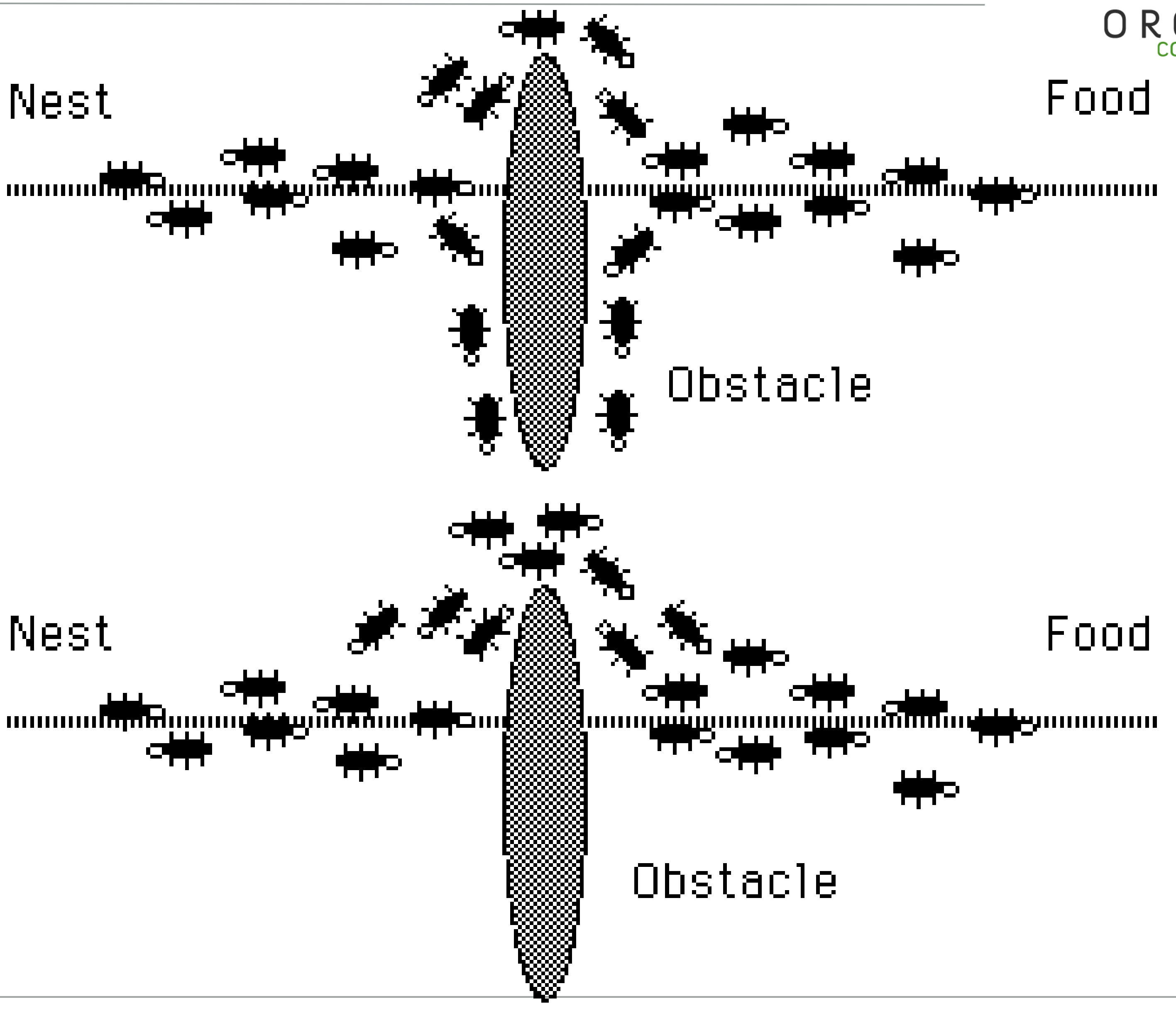
- Prioritisation of food sources based on distance and reachability
- Dynamic adaption of participating ants, e.g. depending on:
  - Size of colony (in number of ants)
  - Amount of stored food
  - Available food sources in vicinity
  - Other colonies and their location

Adaptive path optimisation



[Source: iridia.ulb.ac.be/~mdorigo]

Ants (10)



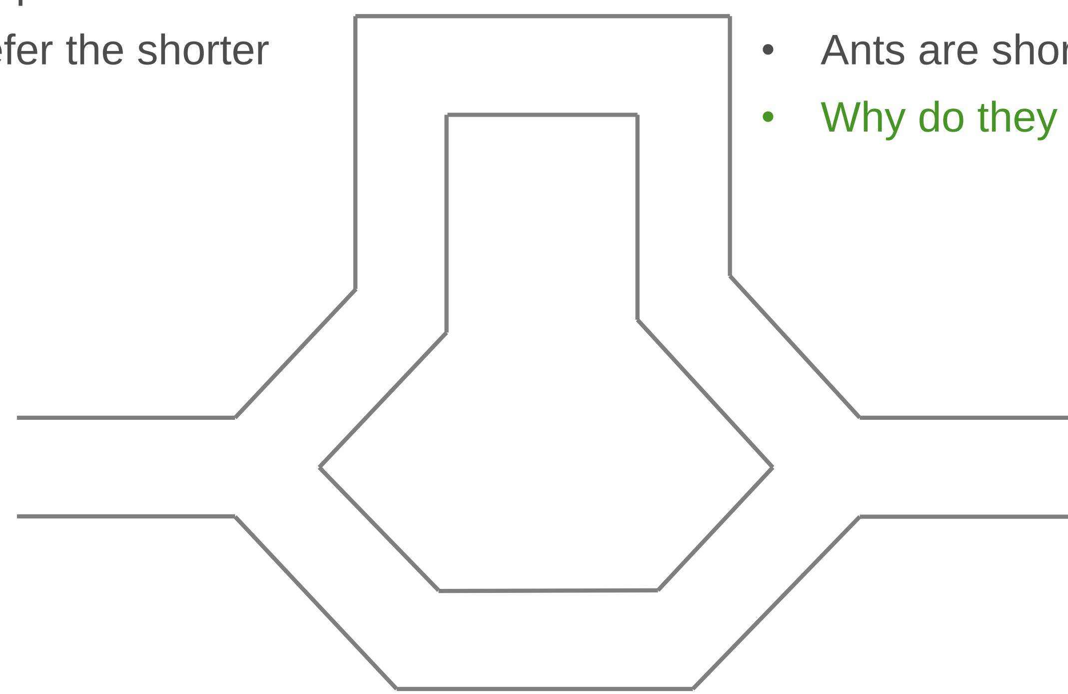
Ants (11)

Behaviour of ants

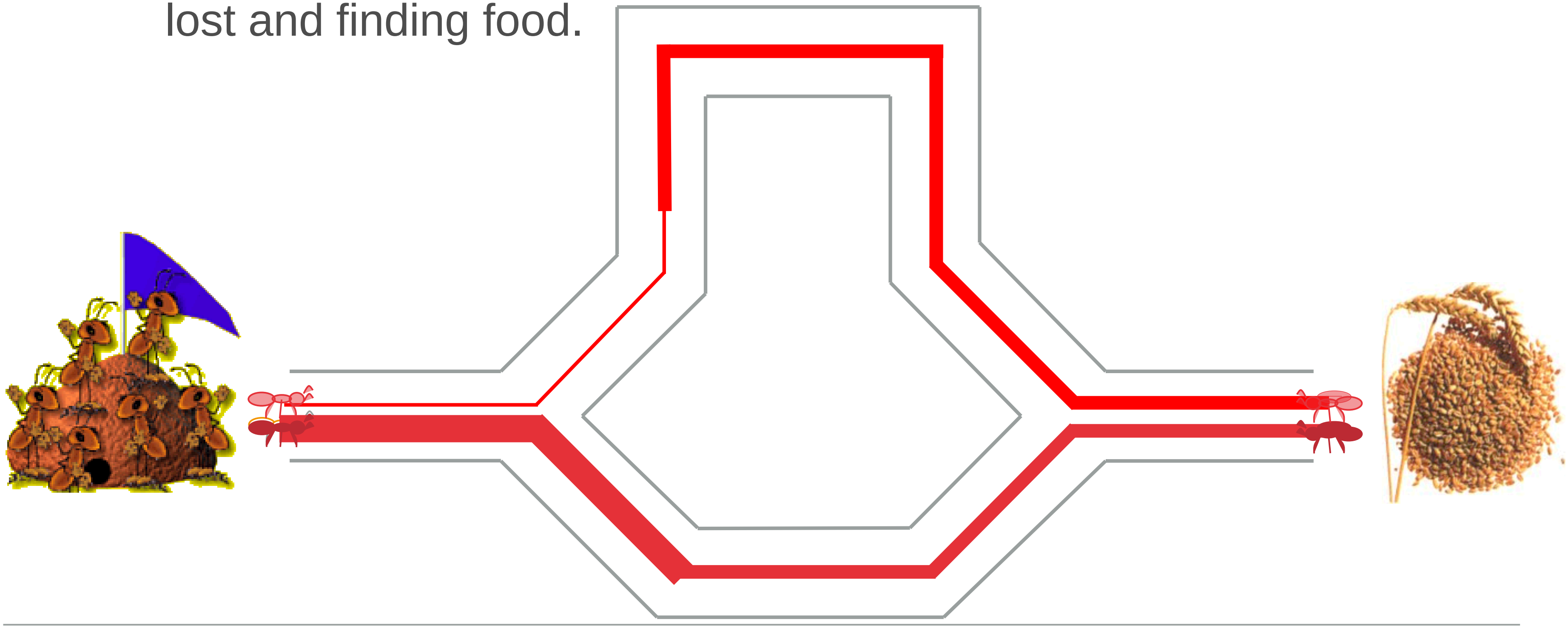
[Deneubourg et al., Dorigo, around 1990]

- Ants explore both paths.
- They find and prefer the shorter path.

- Distances are unknown!
- Ants are short sighted!
- Why do they find the shortest path?



- Ants deposit pheromones on their path.
- Ants prefer paths having higher pheromone concentration.
- Pheromones evaporate.
  → Ants do not care about shortest paths, only about not getting lost and finding food.



Question: What governs here?

→ Centralised control is impossible!

Characteristics

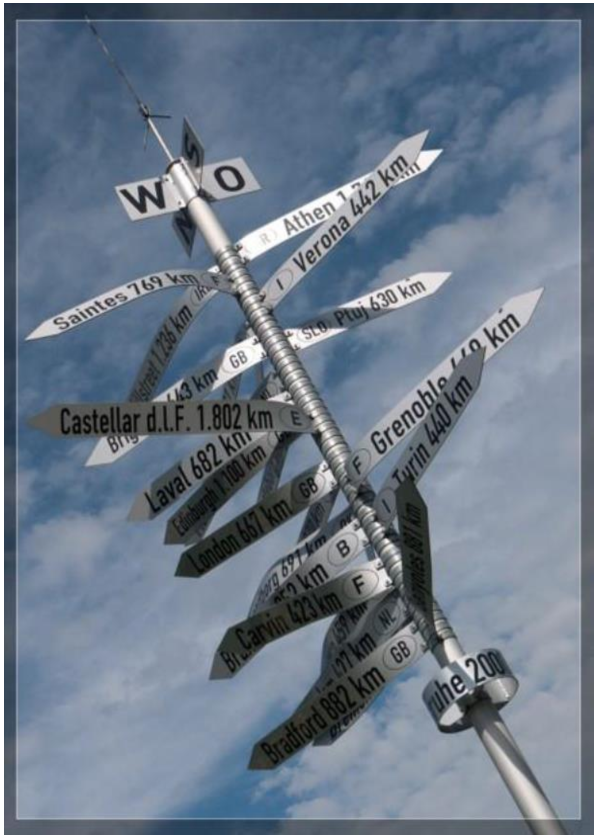
- Colony size ~ 〚8∗10〛^6
- Without central authority: no one is “in charge”!
- Colony lifetime ~ 15 years (about the lifetime of one queen)
- Colonies have a “life cycle”
- Older ants behave differently from younger:
  - Older are more fixed in ways.
  - Younger are more responsive to environmental conditions.
  - Younger, though smaller, are more persistent & aggressive.
- But ants live no longer than one year!
  - Males live one day (fight & mate).
  - Colonies in an area (which may be as large as Southern England) coordinate their Nuptial Flight to one single evening.

ORGANIC COMPUTING

Insight: Emergence

- Something just appears:
  - A shortest path
  - Protection
  - Role assignment
  - Gardens
- Not predictable from individual entities
- Ingredients
  - Self-organisation
  - Autonomous decisions
  - Interaction


- A first example: water temples in Bali
- A second example: ants
- Emergence
- Term definition
- Quantification of emergence
- A refined approach to emergence quantification
- Conclusion and further readings



- Emergence is a phenomenon that can be observed in various systems (especially in nature).
- In principle, it can be found if the following “ingredients“ exist:
  - A set of multiple (homogeneous) individuals
  - Individuals are self-motivated and self-organised
  - Individuals interact with each other
  - No centralised authority controls the process
- “The whole is more than the sum of its parts!”
- Let‘s have a look at some examples:
  1. From nature
  2. From technical systems
  3. From social systems

Examples from nature: swarms

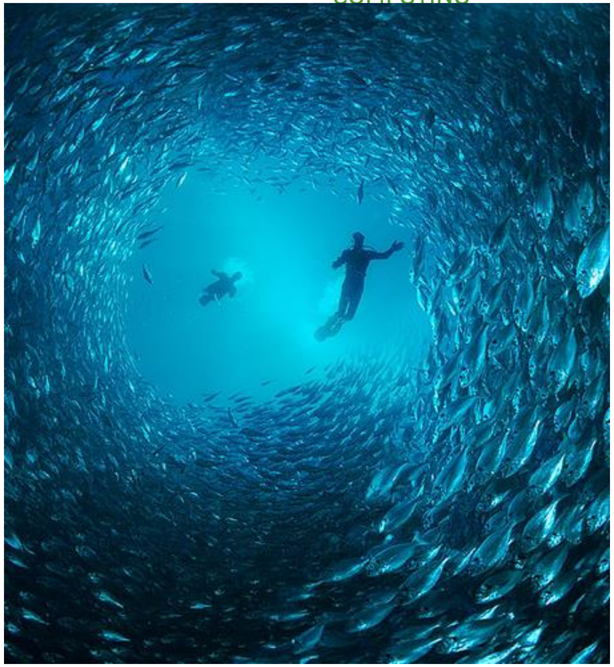
Swarms

- Flock of birds
- School of fish

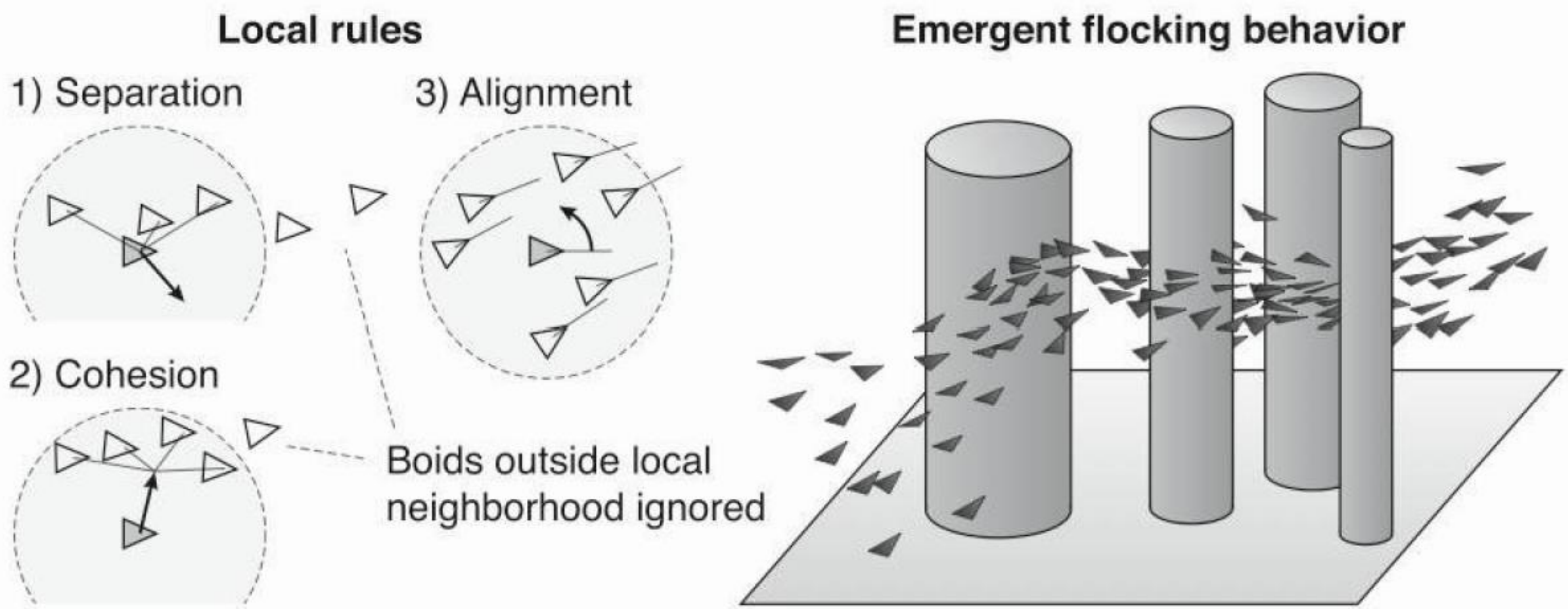
- Behave as a unified organism
- No leader, no control
- Very simple rules

Advantageous for participants:

- Avoid attackers (appear as huge fish)
- Highly resource efficient (wind for birds)



Examples from nature: swarms (2)



1. Separation: Boid maintains a given distance from other boids
2. Cohesion: Boid moves towards center of mass of neighboring boids
3. Alignment: Boid aligns its angle along those of neighboring boids

Examples from nature: termites

Termite colonies

- Are able to build large “cathedral” structures.
- Structures consist of cone-shaped outer walls and ventilation ducts („Kanäle“).
- Brood chambers are situated in central hive.
- Spiral cooling vents („Öffnung“), support pillars.

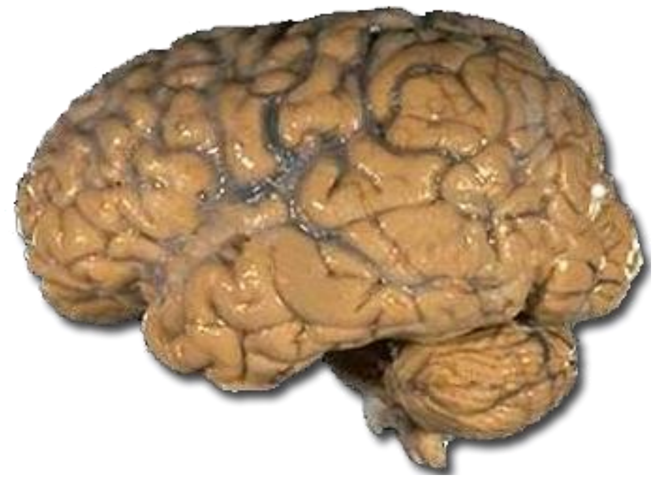
Examples from nature: termites (2)

Characteristics

- No central plan!
- No intelligence required from the individual termites, just simple individual behaviours.
- Local and global interaction between termites achieve emergence.
- To ensure that the “cathedral” adapts to local conditions, a randomness to the individual’s behaviour is necessary.
- Central, top-down control would actually suppress the positive effect of emergence in termite cathedrals.

Examples from nature: the human brain

ORGANIC COMPUTING



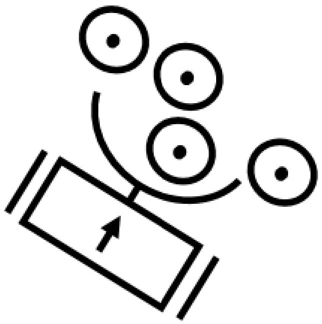
The human brain:

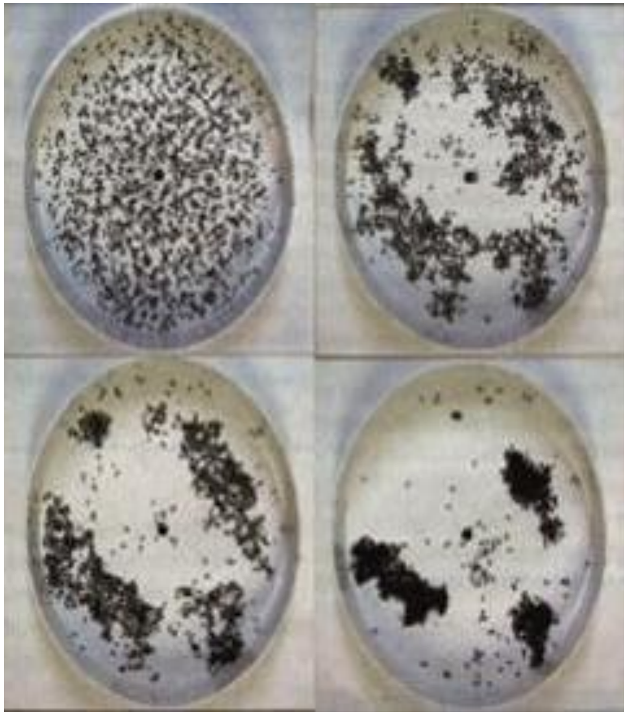
- Micro-level: the neuron
  - exhibits very simple behaviour
  - equals single-bit memory
  - has some stochastic characteristics
- Macro-level: the brain
  - About 1.5 kg, volume of around 1130 (f) / 1260 (m) cm^3
  - Consists of billions of neurons
  - Displays an infinitely sophisticated and complicated behaviour
  - I.e. language, visual, aural, and tactile I/O, the arts, culture, emotions, as well as logical thought and processing
  - Is robust, adaptive, innovative
- An examination of individual neurons cannot predict the behaviour of billions of them working together!

Technical examples: the candle mover

"Candle Mover"

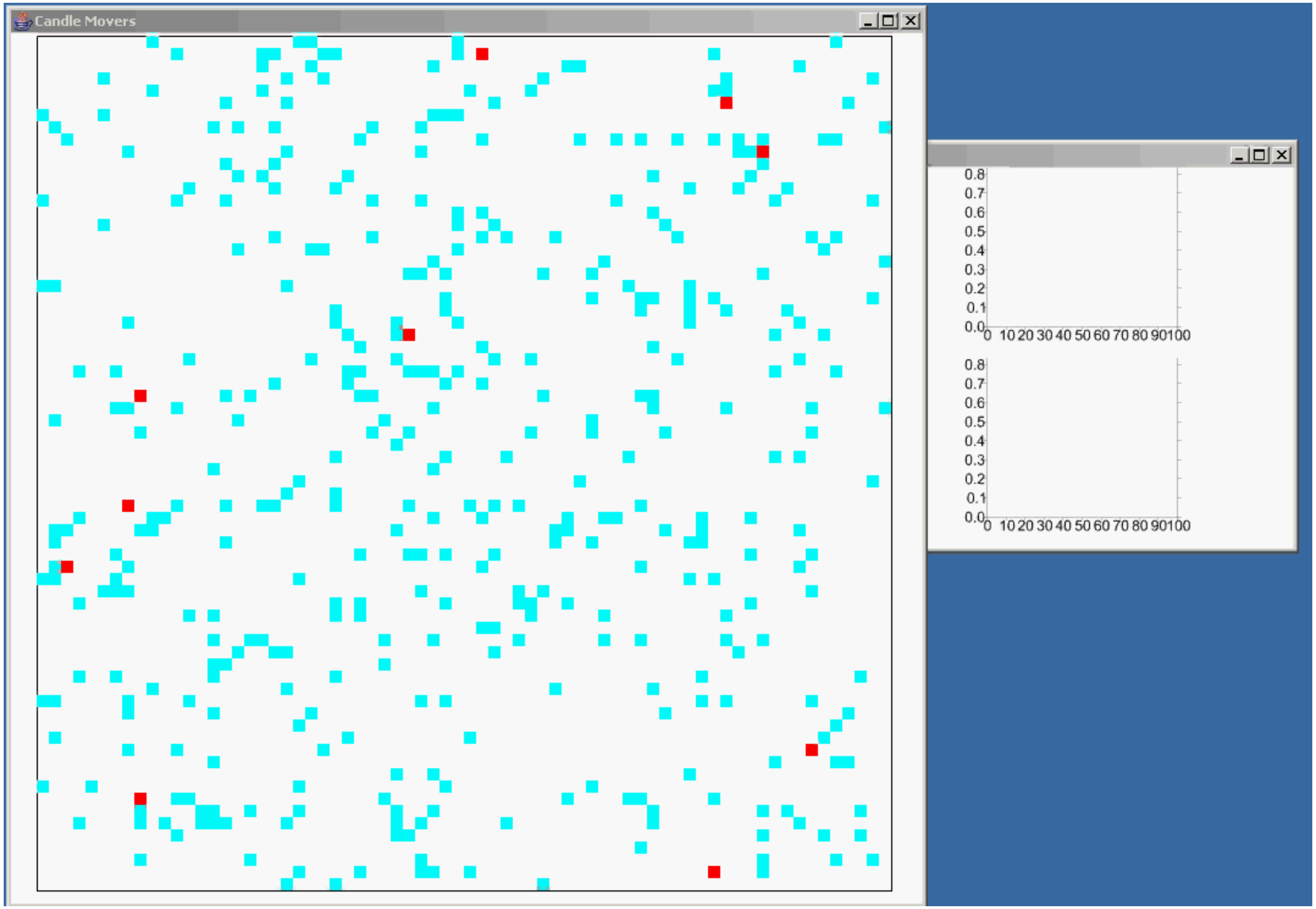
- 1 sensor (pressure)
- 1 actuator (direction)
- In case of 2 candles: move backward, turn, continue to move forward





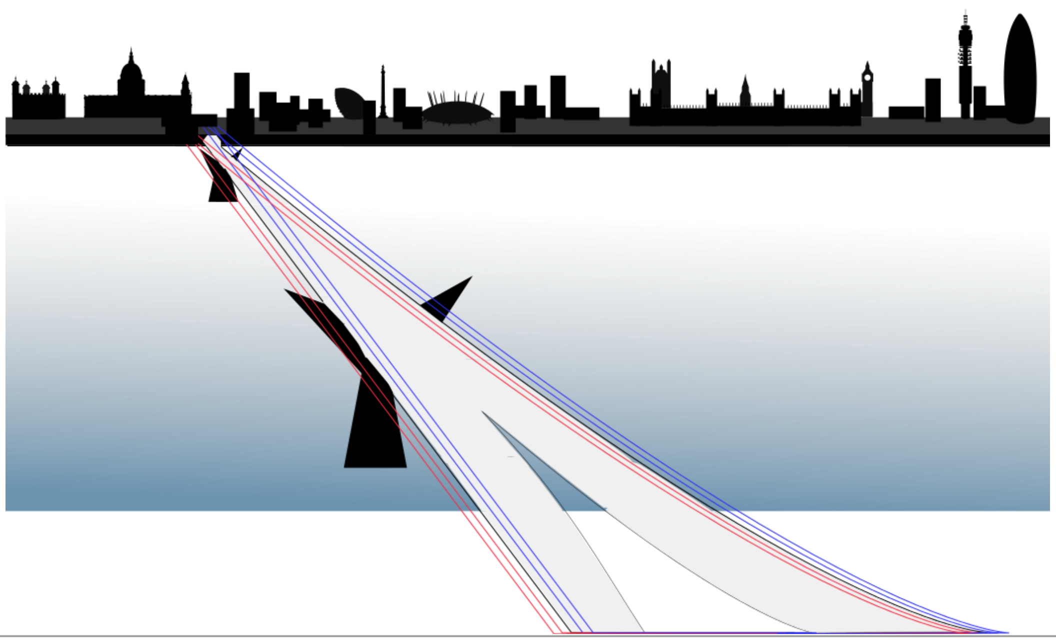
Technical examples: the candle mover (2)

What happens?



London Millennium Foot Bridge

- Built according to classical engineering.
- Analysis showed that the bridge was sufficiently strong and rigid.
- Immediately after opening, it had to be closed due to strong lateral swinging caused by a number of walking pedestrians.



Technical examples: London Millennium Bridge (2)

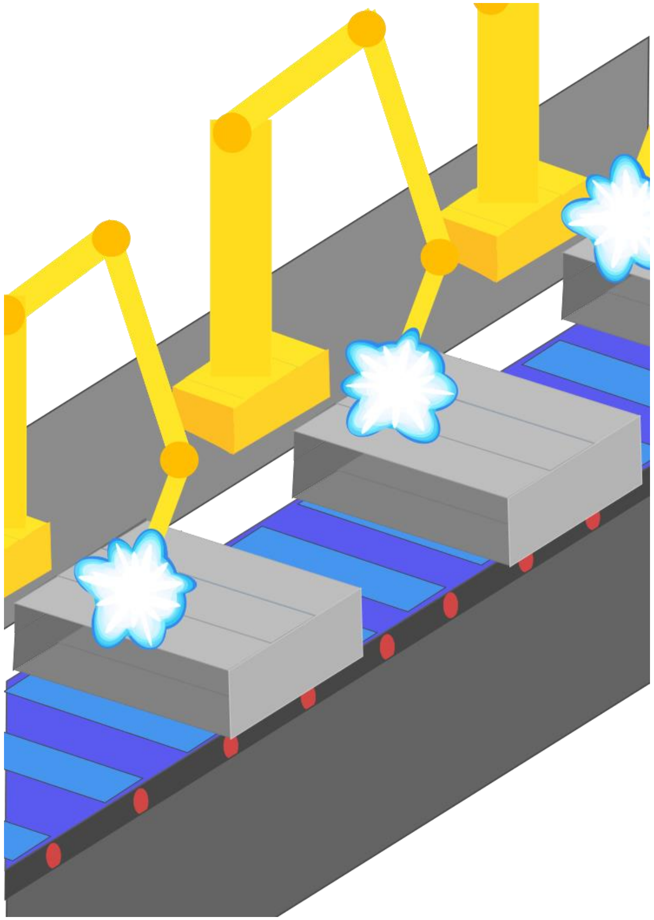
ORGANIC COMPUTING

- Analysis of the emergent swinging effect
  - Natural lateral frequency of the bridge was close to the normal walking frequency of pedestrians.
  - Pedestrians were becoming synchronised in phase and frequency to each other.
    → Humans on a swaying surface tend to subconsciously synchronise their footsteps to the sway.
  - The bridge designers did not anticipate this phenomenon.
- Individual behaviour responded to the common network (the swaying bridge), thereby resulting in an unexpected top-level system behaviour.
  → This is communication, not control!

ORGANIC COMPUTING

Automotive Welding Robots
(“Schweißroboter”)

- The weld’s quality depends upon the line voltage.
- Set of robots were installed in a factory.
- Random irregularities/defects were observed.
- Quality management techniques did not work.



Technical examples: welding robots (2)

Failures

- Caused by line voltage drops.
- Due to simultaneous welds from several robots.
- Design assumed random (non-synchronised) operation.
- All robots share the same networked voltage line.

Approach to alleviate the problem:

- Robot monitors the line voltage and waits until it is high enough.
  → Problem became worse: increased synchronisation.
- Result: No synchronisation due to random delays.
  → Note the importance of stochastic behaviour.

Technical examples: hard drives

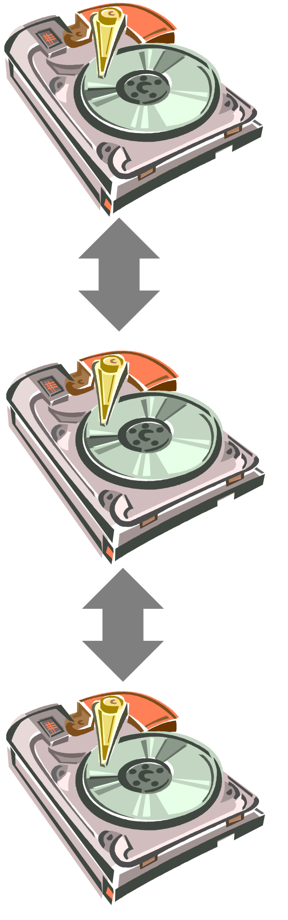
“Enterprise” Server Disk Drives

- Sensitive to vibration
- Especially in case of synchronised seek activity
- Set of disk drives mounted together → Data faults were experienced.
- I.e. computer system database searches caused several disk drives to seek simultaneously, thereby building up synchronised vibration that disturbed each other’s operation.

Technical examples: hard drives (2)

Emergent behaviour

- Simultaneous disk drive seek operations are inherent to large enterprise servers – inducing emergent behaviour among multiple disk drives.
- The disk drives have to be especially designed for such applications.
  → Much more expensive than "ordinary" disk drives.



Technical examples: traffic jams

Freeway Traffic Jams

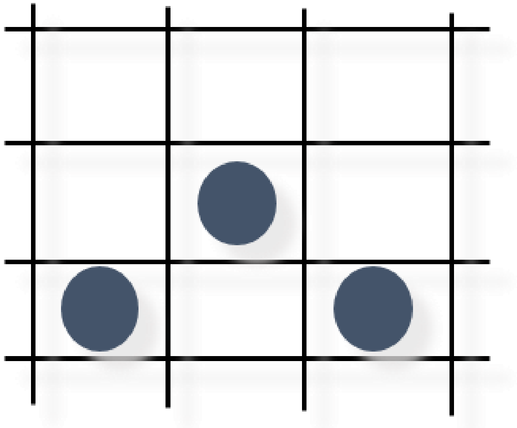
- Drivers act egoistically (avoid traffic jams).
- Observation: Minor perturbation in high traffic periods cause traffic jams.
  - Areas of light and areas of jammed traffic appear periodically.
  - Especially at fast lanes.
  - Total effective flow rate of cars is significantly reduced.

v=const 85km/h -> full stop	v=const 85km/h -> break hard	v=const 85km/h -> break	v=const 85km/h -> slow down	v=const 80km/h

Technical examples: traffic jams (2)

- Result of emergent behaviour!
  - Individual drivers attempt to respond to minor perturbations, but with delay.
  - Drivers in the fast lane have more “gain” in their behaviour.
  - Combination of non-linear delays and high gain leads to oscillations.
    → Result: congestion and reduced flow rate.
- Solution: relax, go with the flow (i.e., reduce your driving “gain”).
  → Leads to variability in separation distances.
  → Net flow rate is only slightly decreased.

- Field of Finite State Machines (FSM)
- Automaton changes its state depending on the states of its neighbours
- Example: Game of Life (John Conway) http://www.bitstorm.org/gameoflife/

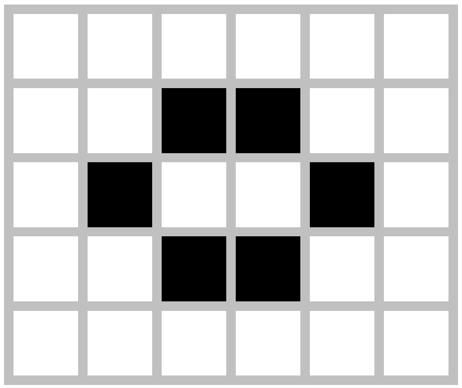


- Example rule set:

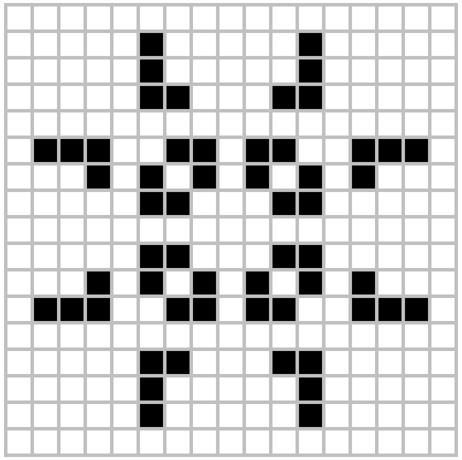
≤ 1	neighbour	dead
2	neighbours	const
3	neighbours	alive
≥ 4	neighbours	dead

ORGANIC COMPUTING

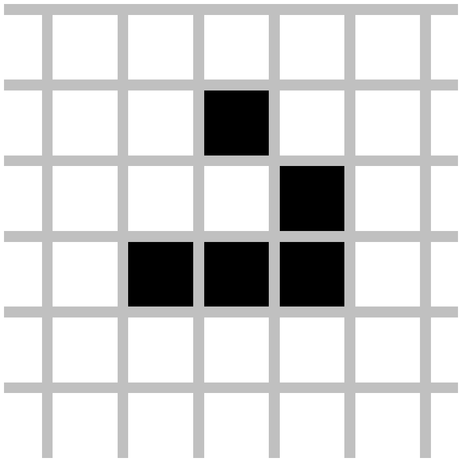
The “Game of Life“



Static



Oscillating

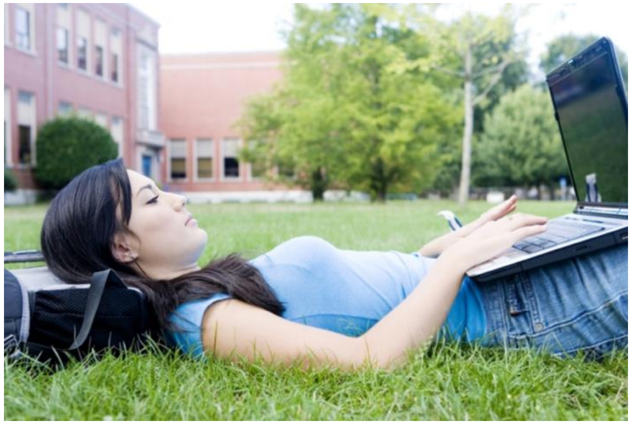


Moving

- Generalisation based on: form, position, colour, communication, etc.
  → Used to simulate united cell structures.

Source: http://en.wikipedia.org/wiki/Conway%27s_Game_of_Life

Social examples: sidewalk structures



- Optimised sidewalk structure
  - University of Michigan built a new campus.
  - Concrete sidewalks were to be placed in the Quad between buildings, but how to design?
  - Once installed they are difficult to change.
  - Previous approaches turned out to be not useful.
- Concept:
  - Plant grass and allow students to walk as they want.
  - Concrete sidewalks were installed according to emerging patterns.
  - Students and faculty are cognitive and adaptive elements in a larger system.
  - At a system level, their patterns of walking could not be accurately predicted.
  - Later analysis showed that as a group, an optimum sidewalk structure was derived.

Social examples: social networks

ORGANIC COMPUTING

Social network:

- Nodes = people
- Edges = relations between people
  → E.g.: friends, relatives, colleagues
- Paths:
  - Chain of people
  - E.g.: friend of my friend of my friend

Interesting properties:

- Decentralised, self-organising, robust, scalable network
  → E.g.: people are born, die, get to know new people
- Efficient (short) communication paths & decentralised routing
  - See Milgram's Experiment

Social examples: Milgram's experiment

- Analysis of paths in social networks
  - Conducted in the 1960s by Stanley Milgram
  - Stanley Milgram (1933 –1984) was an American social psychologist.
- Milgram sends a letter to 160 randomly selected persons from Omaha and Nebraska (USA).
- Letter contains task: Deliver letter to a certain stock broker in Boston, Massachusetts, USA.
- Constraints: Persons must only send letter to someone they know at a "first name basis" (i.e.: friends, colleagues).
- Results:
  - 44 letters reached the target
  - Average number of "hops": 6
    I.e., short paths in network of 200 million US citizens!

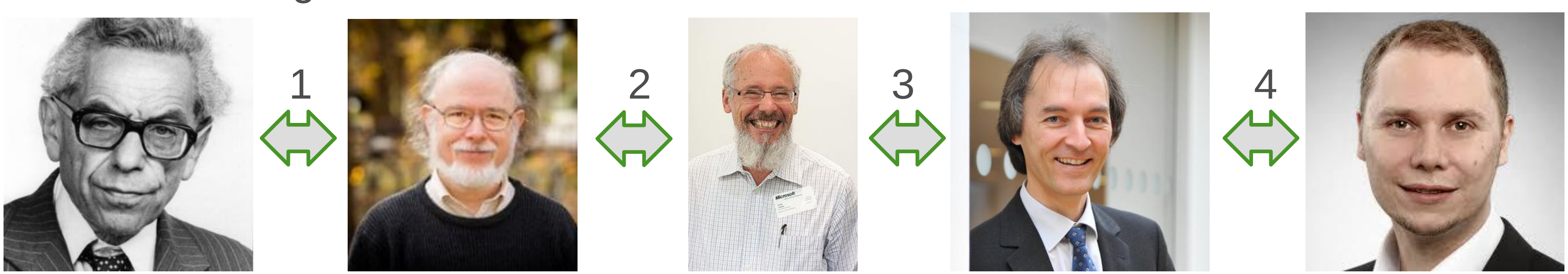
Emergent property: This is a small world!

Further examples for small world networks:

- Network derived from movie database
  - Nodes: actors
  - Edge between two actors if they have acted in same movie
- Bacon number of actor:
  Shortest path between an actor and Kevin Bacon
  - Average bacon number: 2.9
  - Via Kevin Bacon, any actor can be linked to any other in 6 “hops”
- 6 is a typical distance between pairs of nodes in such networks.
  → 6 degrees of separation

Further social examples

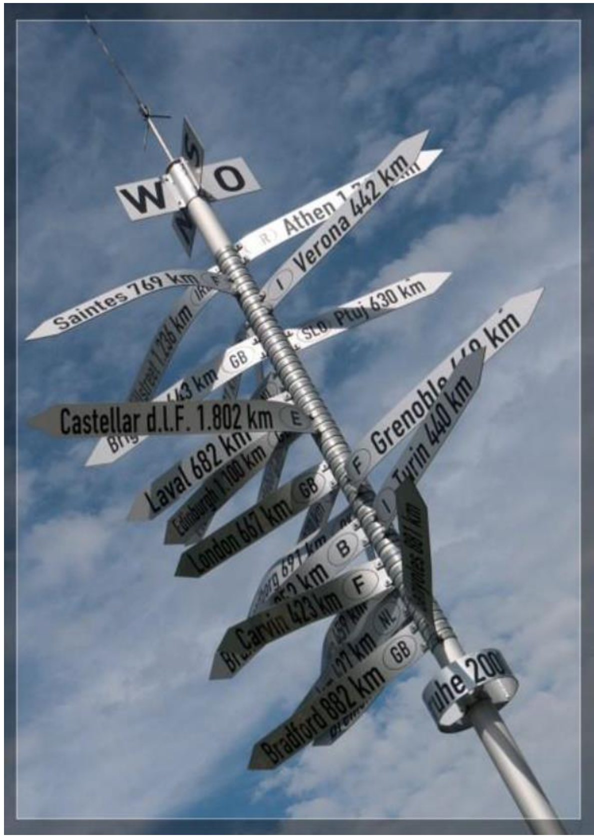
- “Die Zeit” linked a Turkish Kebab-shop owner in Frankfurt, Germany, to his favorite actor Marlon Brando in 6 hops:
  - Shop owner has a friend in California who works together with the boyfriend of a woman who is in the same student’s union of the daughter of the producer of the movie “Don Juan” starring Marlon Brando.
- Erdös number: distance in graph of paper (co-) authors
  - Paul Erdös: famous mathematician
    (> 1500 papers with > 500 co-authors)
  - Average Erdös number: 4.7
  - Average distance between authors: 7.3



Paul Erdös | Brendan D. McKay | Peter Eades | Jürgen Branke | Sven Tomforde

Agenda

- A first example: water temples in Bali
- A second example: ants
- Emergence
- Term definition
- Quantification of emergence
- A refined approach to emergence quantification
- Conclusion and further readings

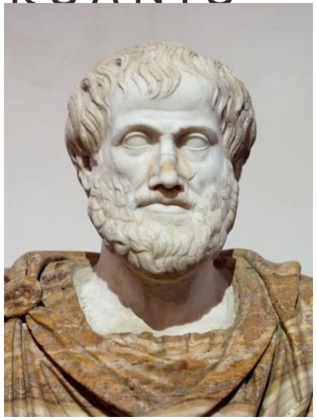


Term definition: “emergence”

- How macroscopic behaviour arises from microscopic behaviour.
- Emergent entities (properties or substances) ‘arise’ out of more fundamental entities and yet are ‘novel’ or ‘irreducible’ with respect to them.
  [Stanford Encyclopedia of Philosophy: http://plato.stanford.edu/entries/properties-emergent/]

- “It is unlikely that a topic as complicated as emergence will submit meekly (“widerspruchslos”) to a concise definition, and I have no such definition to offer.“
  [John Holland: „*Emergence: From Chaos to Order*“]

Emergence (2)



Emergence

- Comes from “*to emerge” (in German: „auftauchen“)*
- Verbal description:
  - “A system is more than the sum of its parts.“
  - From Aristotle (384 BC – 322 BC); Greek philosopher and polymath.
- Is a characteristic of the whole system, not part of the subsystems.

Definition:

- „An emergent system characteristic is a property, which is not only defined by the elements contained in the system, but by the interaction between these elements.“
- Emergent system characteristics are not computable by *summarising* the characteristics of the contained parts.
- Emergent behaviour is the result of interactions between processes.

Emergence (3)

Sources of Emergent Behaviour

- Unwanted/unintended synchronisation, or oscillation
- Local or global networks allowing wanted/unwanted communication (often unintentional networks)
- Non-linear interaction between simple elements
- Thrashing: competition over a scarce resource
- Chaotic (even if deterministic) behaviour
  → Emergence is nature's way of dealing with chaos.
- Intentional, or unintentional, feedback loops with poor gain margins.
- Intelligent, adaptive elements in the system.
  → This means that the behaviour of the elements as well as the system architecture varies with time, depending upon conditions.

Emergence (4)

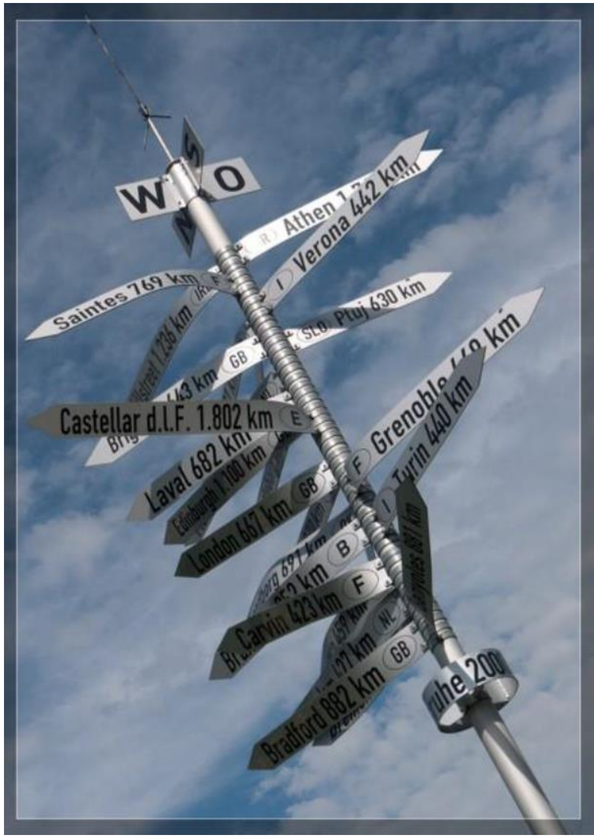
Management of emergent behaviour

- Prevent or mitigate the sources of emergence.
- Design limits into systems to lessen the negative effects of emergence.
- Add **extra stability and robustness** to the system.
- Use simulations to detect and design for emergence (caution, very sophisticated simulations required; beware of chaos theory).
- Reduce non-linearity.
- Increase scarce resources to minimise thrashing.
- Use the evolutionary (also called incremental or spiral) development life cycle to discover emergent behaviour.

- Goal: Promote its positive effects and suppress its negative effects!

Agenda

- A first example: water temples in Bali
- A second example: ants
- Emergence
- Term definition
- Quantification of emergence
- A refined approach to emergence quantification
- Conclusion and further readings



ORGANIC COMPUTING

Emergence in Organic Computing / intelligent systems

- Why do we have to consider emergence in OC systems?
- Do we want do explicitly design emergent effects?
  → This is hardly impossible…
- Emergence is **not** something we want to design, but something that will appear automatically!
  - Emergence is the result of interactions between a set of self-organised entities.
  - OC systems consist of a set of self-organised interacting entities.
    → We have to be aware of emergence: positive and negative!
  - In technical systems: How to be aware of something?
    → We need to measure it!

Emergence in Organic Computing systems (2)

- Structural emergence in the sense of collective self-organisation shows as:
  - patterns in time and/or space
  - patterns (order) at the system level.
- Patterns at system level are realised by:
  - interaction
  - of (a large number of) similar individuals.
- These patterns have properties not existent in the individuals.

- How can we measure emergence?
  - Emergence measures the result (in terms of „order“) of some unknown process.
  - Order per se says nothing about self-organisation.
  - Entropy is a measure of order. How is emergence related to entropy?

Quantification of emergence

Goal:

- Assign a high emergence value to a system, which is perceived as emergent!
- Quantification of emergence in OC systems.

Approach:

- Basis: verbal definitions
- Emergence is always associated with patterns (symmetry breaks).
- This corresponds to structural emergence.
- Patterns represent order.
- Order can be measured in terms of entropy.
- Therefore, we must: 1) define entropy and 2) relate it to emergence!

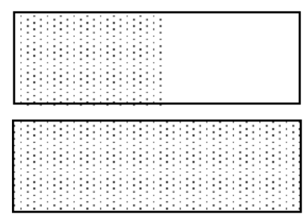
What is order?

Dissipative structures (Prigogine)

Where is more order? Left or right?

Right: Higher entropy

Left: More structure



Gas molecules: distribution with low probability

Gas molecules: thermodynamic equilibrium

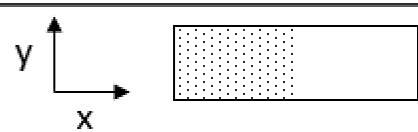
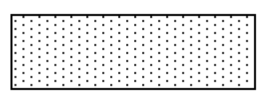
higher entropy

Order is subjective!

- The perception of order depends on the view of the observer.
- The purpose and the sensory equipment of the observer determine the view.
- A system can be rated as orderly or disorderly dependent on the utility.

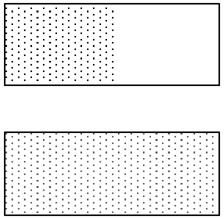
Views and order

- „Order“ or „disorder“ depends on
  - the purpose and
  - the view (aspect).
- A view is determined by the selection of certain attributes (or a group of attributes) of an object.
- Example:

View	x position	colour
	higher order	same order
	lower order	same order

- The view is influenced by the pre-processing of sensory data.

- Entropy is a thermodynamic state variable.
  - High entropy ⇔ high probability.
  - Clausius: Entropy of a closed system will always increase.
- Entropy: measure of (dis)order (high entropy = low order).
- Statistical definition of entropy (S): $\boxed{S = k_B \cdot \ln(\Omega)}$
  - $k_B = 1{,}38 * 10^{23} \frac{J}{K}$ (the *Boltzmann* constant)
  - $k_B =$ average kinetic energy of an ideal gas particle at a temperature of 1 Kelvin.
  - $\Omega =$ probability of a macroscopic state (= the number of possible states of the particles in a system / total number of possible states)
- Example: irreversible diffusion



Lower Ω, lower entropy

Higher Ω, higher entropy

Shannon's entropy

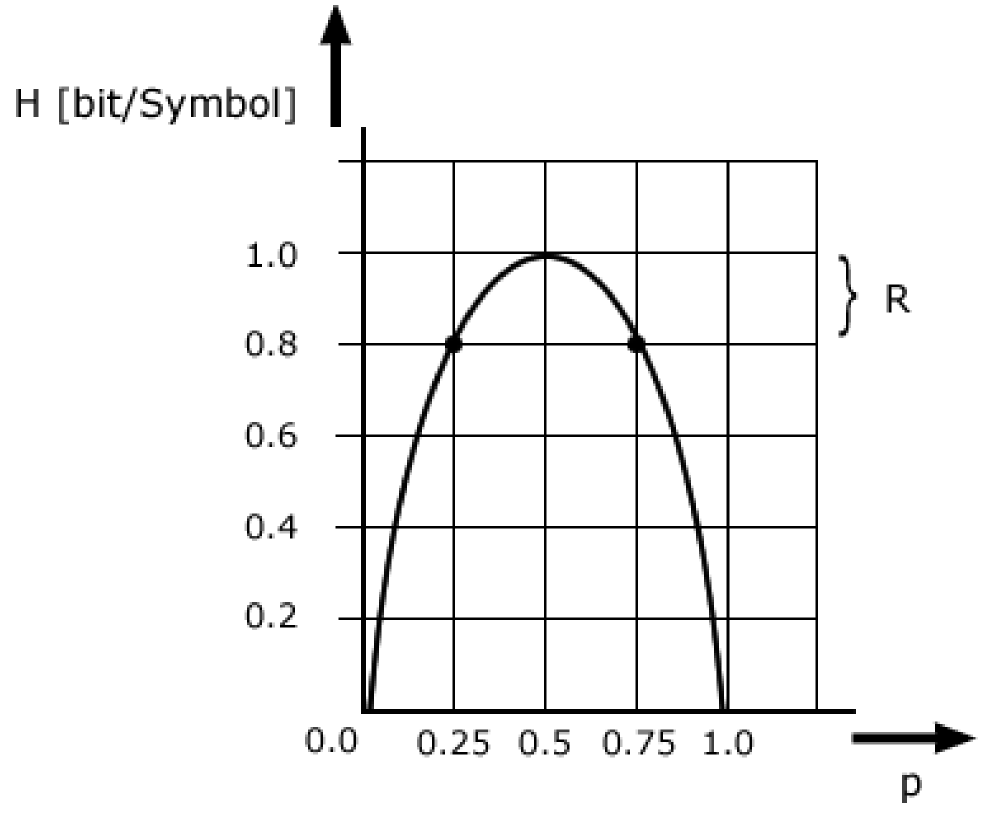
- Definition from information theory (Shannon):
  - Entropy is a measure of information.
  - Message source M, alphabet Z:

$$\boxed{Entropy\,H(M) = -K \cdot \sum_{j=1}^{|Z|} p_j \cdot ld\, p_j}$$

  - p_j = probability for occurrence of symbol $z_j \in Z$ in message source M
  - Entropy H is a measure for random information in a system (or a message source M).
  - K is a constant (can be neglected).
  - High content of random information $\Rightarrow$ low predictability $\Rightarrow$ low probability $\Rightarrow$ high information content.

Shannon's entropy (2)

- Example:
  - 2 symbols (0 and 1)
  - with probabilities p_0 and p_1
  - $H = - p_0 \operatorname{ld} p_0 - p_1 \operatorname{ld} p_1$



Shannon's entropy (3)

- $H = H_{max}$ is desirable, if a channel must transport the maximal „newness“ value per (physical) step.

- In case of: $H < H_{max}$: $R = H_{max} - H$
  - The channel transports useless information (redundancy R).
  - Known information burdens the channel but does not increase the knowledge of the receiver.

- A Shannon channel is "good", if it transports the maximum amount of information:

  $\rightarrow R = 0$!

ORGANIC COMPUTING

Approach: Use the statistical definition of entropy!

- Procedure:
  - Select an attribute A of the system (discrete, enumerable) with values a_j.
  - Observe all elements e_i and assign a value a_i to each e_i.
  - Transform into a probability distribution over the attribute values a_j.
  - Determine:

$$\text{entropy } \mathrm{H}_\mathrm{A} = -\sum_\mathrm{j} \mathrm{p}_\mathrm{j} \,\mathrm{ld}\, \mathrm{p}_\mathrm{j}$$

- Each attribute X has its entropy H_X.
- System entropy:

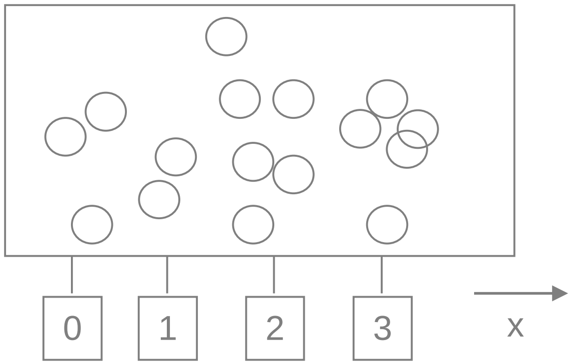
$$H_S = \sum_X H_X$$

- Characterisation of a system by:

a) System entropy (low specificity)
b) Vector of attribute entropies („fingerprint“): (H_A, H_B, H_C...)

Example for the quantification of emergence

- Discrete values of x coordinate: 0, 1, 2, 3

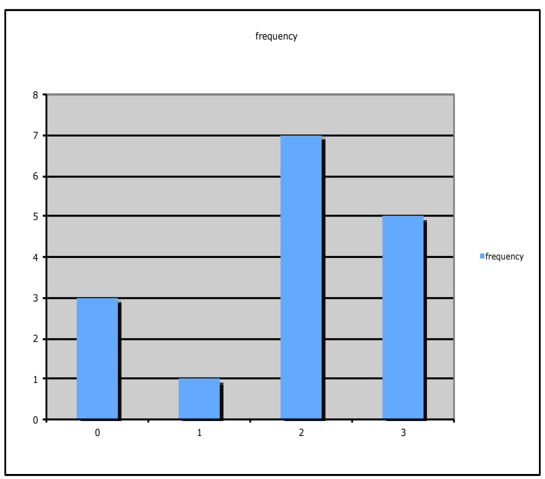


N = 16

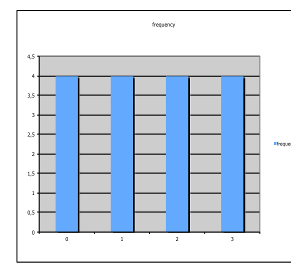
Position:	0	1	2	3
Frequency:	3	2	6	5
p	3/16	2/16	6/16	5/16

$$H_{xcoordinate} = -\left(\frac{3}{16} ld \frac{3}{16} + \frac{2}{16} ld \frac{2}{16} + \frac{6}{16} ld \frac{6}{16} + \frac{5}{16} ld \frac{5}{16}\right)$$

$$= 1{,}72 bit / element$$



Uniform distribution:



$$H_{x\,coordinate} = -\left(\frac{4}{16}\, ld \frac{4}{16} + \frac{4}{16}\, ld \frac{4}{16} + \frac{4}{16}\, ld \frac{4}{16} + \frac{4}{16}\, ld \frac{4}{16}\right)$$

$$= 2\ bit / element$$

Emergence: definition

- Entropy ≠ Emergence!
- First try of a definition: Emergence M is the decrease of entropy H from a start state to an end state:

  $M = \Delta H = H_{Start} - H_{End}$
- This is a necessary, not a sufficient condition for a process to be called emergent. Emergence manifests itself by an increase of order,

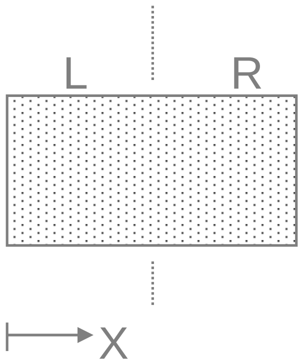
  i.e. $H_{End} < H_{Start}$ and $\Delta H > 0$.
- In addition, the process that leads to the increase of order must be self-organised (not e.g. human-induced).
- Problem:
  The observation of emergent phenomena frequently involves a change of abstraction level.

Emergence: definition (2)

- A change of view to a higher abstraction level leads to a positive ΔH, which is **not** due to an emergent process.
- $\Delta H = \Delta H_{view} + \Delta H_{emergence}$
- Emergence $M = \Delta H_{emergence} = \Delta H - \Delta H_{view}$
- If the two observations (start, end) are made on different abstraction levels, the increase of order due to the change of view (ΔH_{view}) must not be counted as entropy stemming from emergence.
- Or: the entropies H_{Start} and H_{End} are only comparable if they are observed at the same abstraction level ($\Delta H_{view} = 0$).

Emergence: abstraction level

- Example:



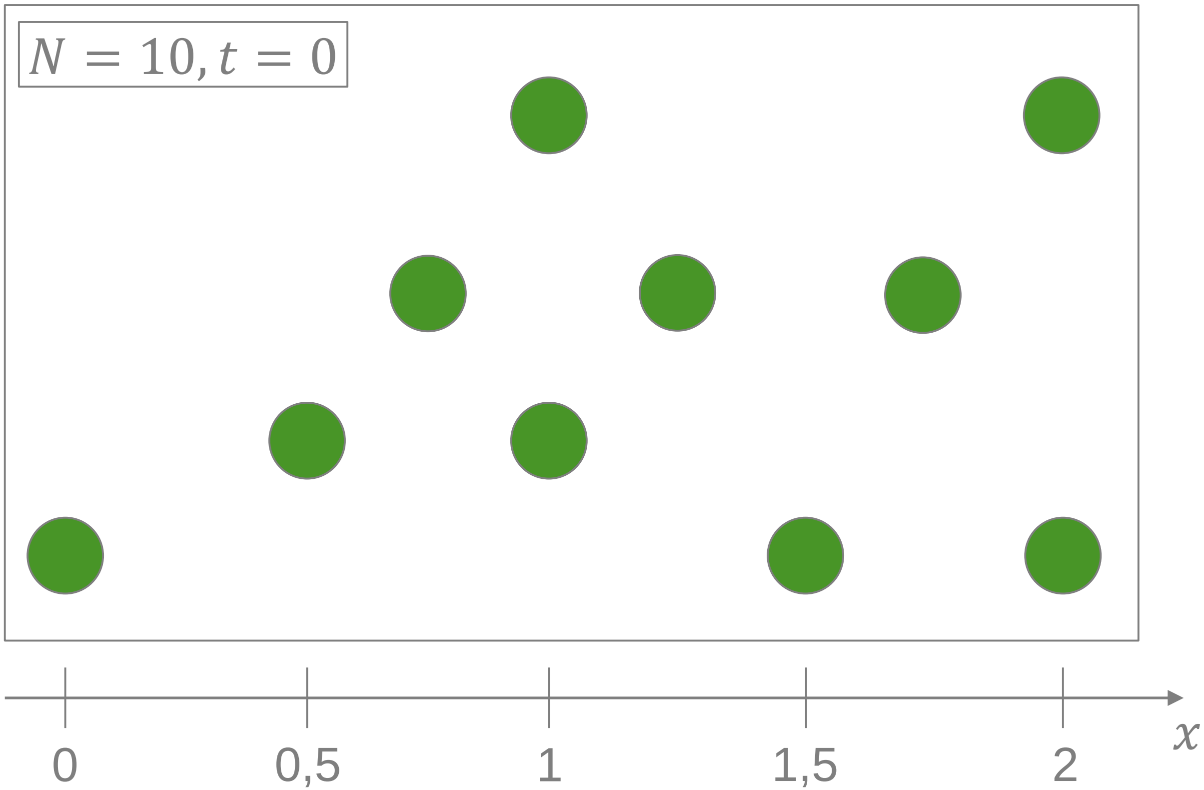
Observation 1 of x coordinate:
32-bit floating point

Observation 2 of x coordinate:
quantisation to 256 values (8 bit integer)

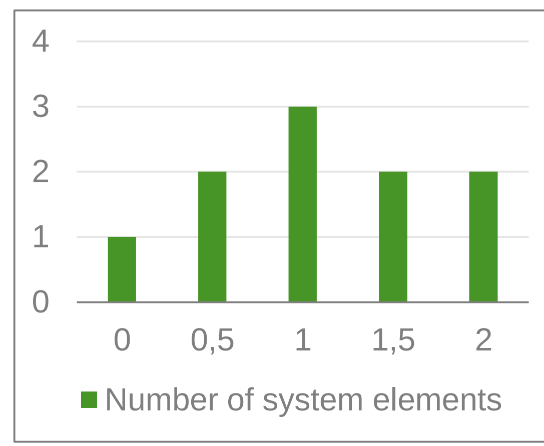
- Quantisation:
  - $\Delta H = 24 \frac{bit}{element}$
  - $\Delta H = \Delta H_{view}$
  - $Emergence\ M = \Delta H_{emergence} = 0$

Quantification of abstraction change

Consider a system S at time $t = 0$



attribute value	frequency
0	1
0,5	2
1	3
1,5	2
2	2



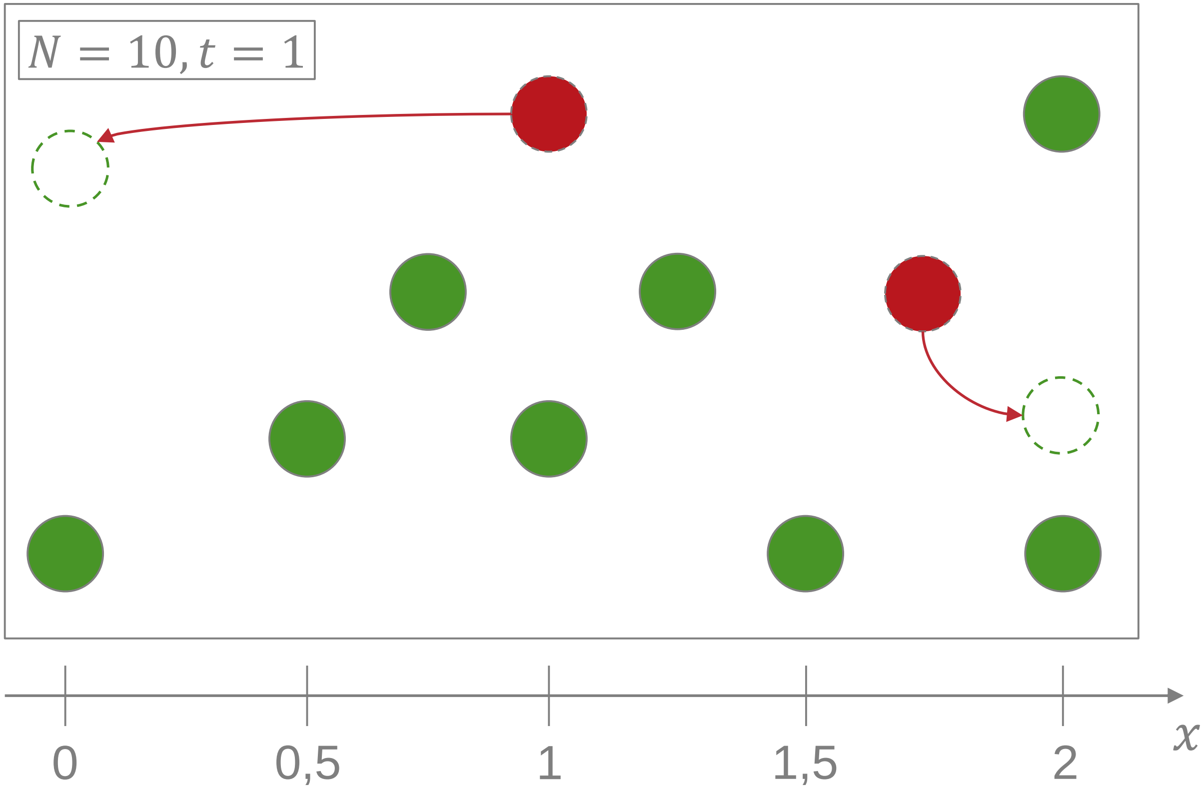
$$H_x^0 = -(\frac{1}{10} * ld\,\frac{1}{10} + 3 * \left(\frac{2}{10} * ld\,\frac{2}{10}\right) + \frac{3}{10} * ld\,\frac{3}{10})$$

$$H_x^0 = 2{,}24643934467$$

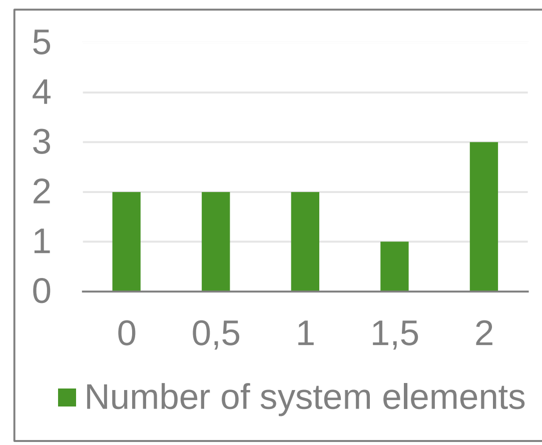
Quantification of abstraction change (2)

ORGANIC COMPUTING

Something happened from $t = 0$ to $t = 1$



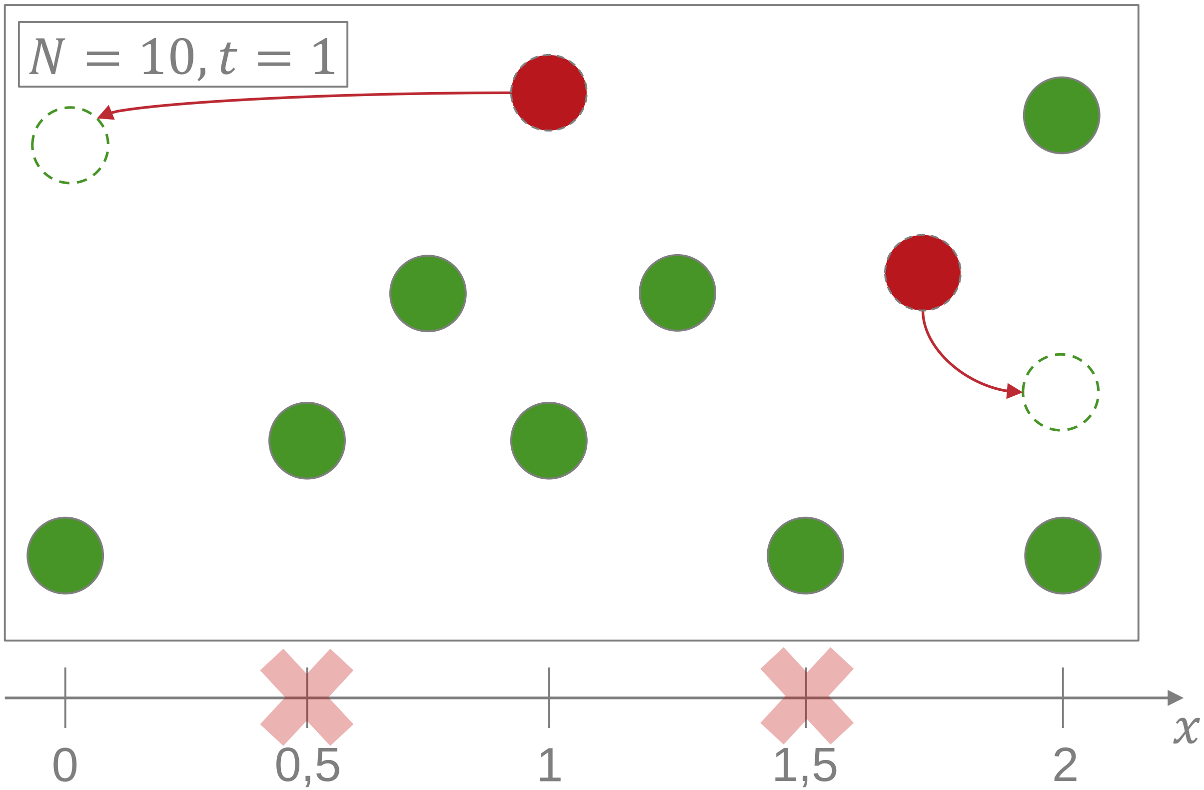
attribute value	frequency
0	2
0,5	2
1	2
1,5	1
2	3



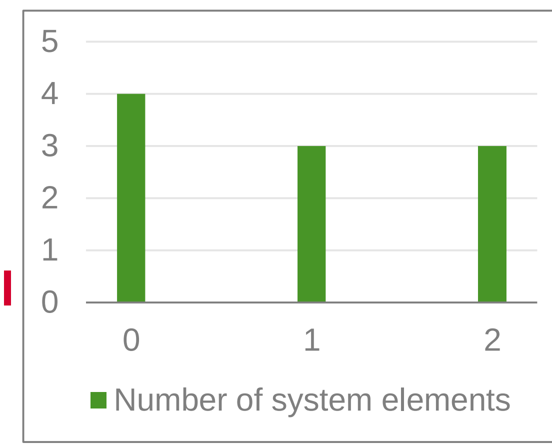
Different state due to self-organised process!

Quantification of abstraction change (3)

Numerical precision changed from *double* to *int*



attribute value	frequency
0	4
~~0,5~~	
1	3
~~1,5~~	
2	3

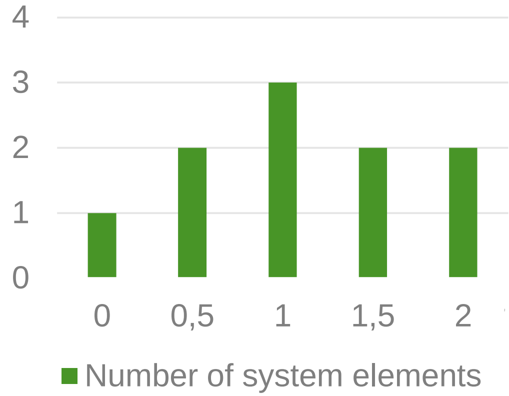


Different state (self-organisation) AND abstraction level

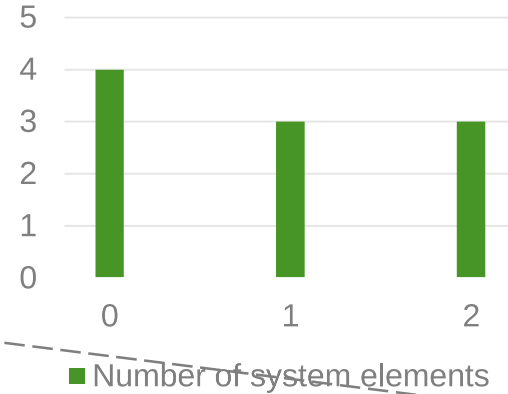
$$H_x^1 = -\left(\frac{4}{10} * ld\,\frac{4}{10} + \frac{3}{10} * ld\,\frac{3}{10} + \frac{3}{10} * ld\,\frac{3}{10}\right)$$

$$H_x^1 = 1{,}57095059445$$

What happened? Do we have a higher degree of order?



S at $t = 0$



S at $t = 1$

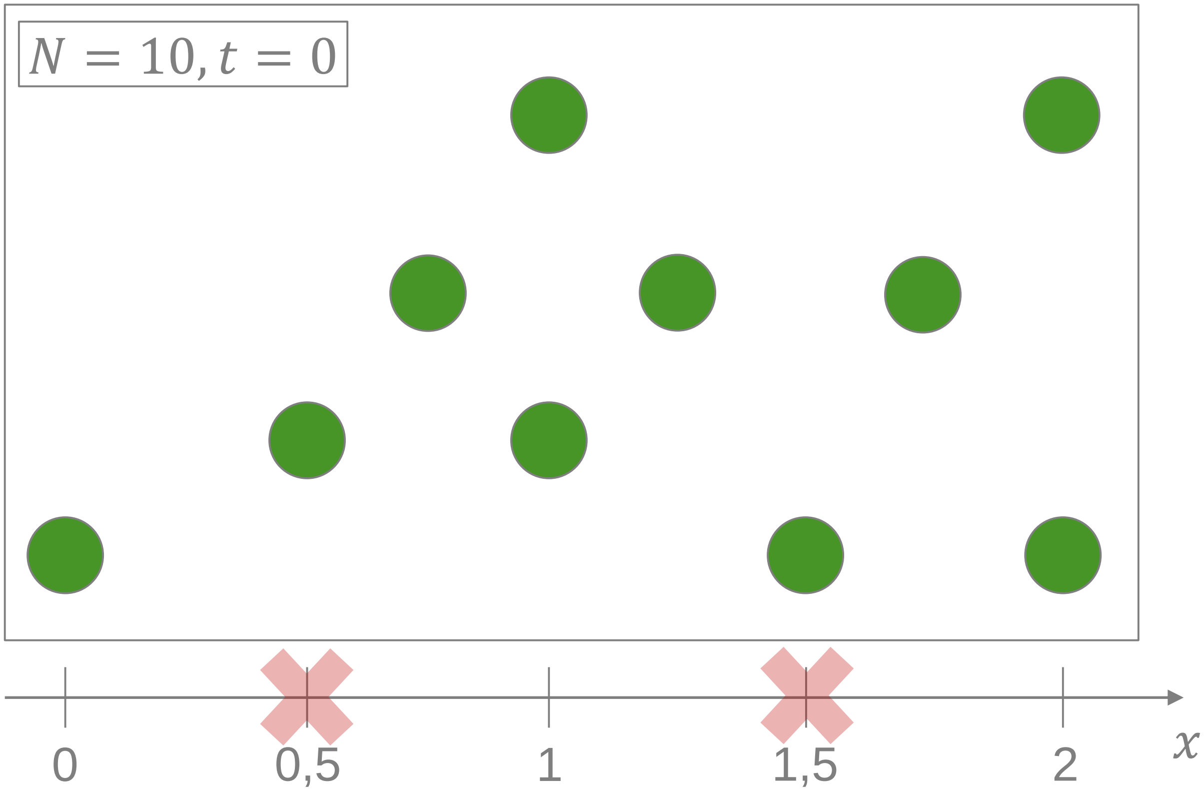
Let's calculate the emergence M:

$$M = H_x^0 - H_x^1$$
$$M = 2{,}24643934467 - 1{,}57095059445$$
$$M = 0{,}67548875022$$

Result: *increase* in terms of *order* (*decrease* of *entropy*)!
But: Subtract influence of ΔH_{view} (i.e. abstraction change)!

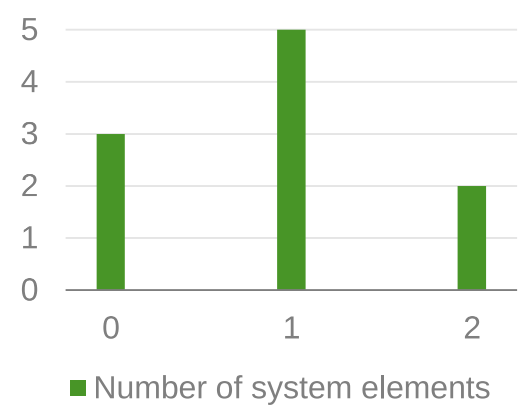
Quantification of abstraction change (5)

First step: adjust the abstraction at time $t = 0$:



Change of abstraction level (double → int)

attribute value	frequency
0	3
~~0,5~~	
1	5
~~1,5~~	
2	2



$$H'^{0}_{x} = -\left(\frac{3}{10} * ld\,\frac{3}{10} + \frac{5}{10} * ld\,\frac{5}{10} + \frac{2}{10} * ld\,\frac{2}{10}\right)$$

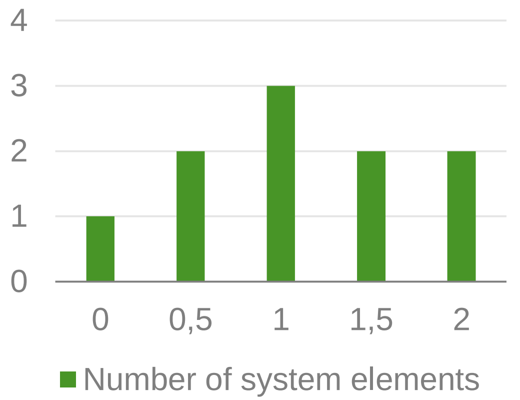
$$H'^{0}_{x} = 1{,}48547529723$$

ORGANIC COMPUTING

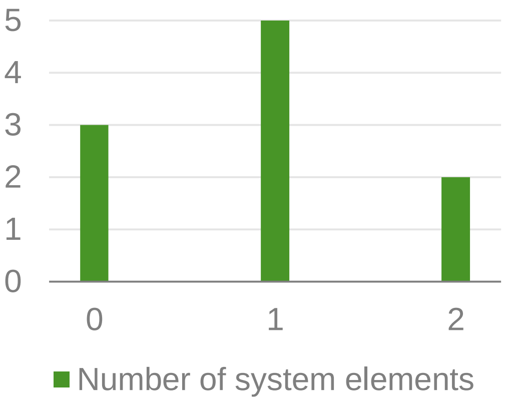
Now, ΔH_{view} calculates:

- $\Delta H_{view} = H_x^0 - H_x'^0$
- $\Delta H_{view} = 2{,}24643934467 - 1{,}48547529723$
- $\Delta H_{view} = 0{,}76096404744$

- Thus, we have an *increase* in terms of *order*, caused by the abstraction level change (less uniformly distributed)!



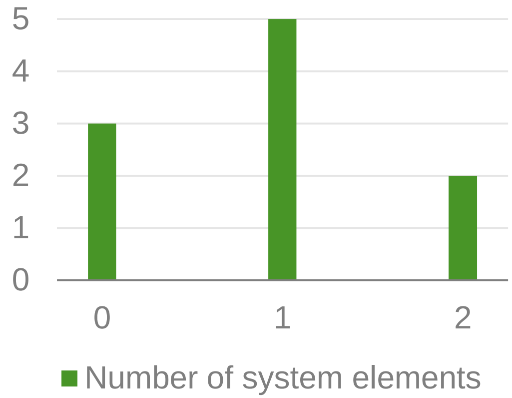
S at $t = 0$



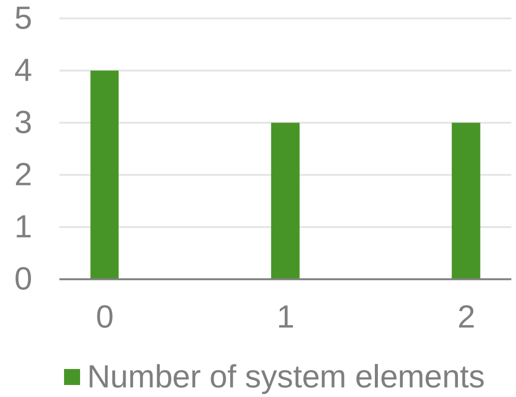
S at $t = 0$

Quantification of ΔH_{view} (2)

- If we compare the states at $t = 0$ and $t = 1$ on the same level of abstraction, we see a *decrease of order*, due to a *higher degree* of *uniform distribution*!



S at $t = 0$



S at $t = 1$

Quantification of ΔH_{view} (3)

Final step: subtract ΔH_{view} from the emergence M calculated before:

- $M = H_x^0 - H_x^1 - \Delta H_{view}$
- $M = 2{,}24643934467 - 1{,}57095059445 - 0{,}76096404744$
- $M = -0{,}08547529722$

- And `Ta-da‘!

- Now, we get a negative emergence value M which indicates an *increase* in terms of *entropy* from time $t = 0$ to $t = 1$, given the change in the level of abstraction.

Emergence and order

- Emergence $M = \Delta H_{emergence}$ is a measure of the (self-organised) increase of order from start to end of a process.
- Is there an absolute indicator of order?
- We can define order as the difference between the entropy at maximum disorder (H_{max}) and at a certain system state (H):

  $M = \Delta H = H_{max} - H = \Delta H_{emergence} - \Delta H_{view}$

  with $\Delta H_{view} = 0$:

  $$\text{Emergence } M = \Delta H_{emergence} = H_{max} - H$$

- H_{max} is the system or attribute entropy for the case of a uniform probability distribution.

Emergence definition

Term definition

- Emergence is the increase of order due to self-organised processes between the elements of a system.
- Higher order is measured in terms of a lower description complexity, i.e. in terms of lower information content.
- The more structure a system displays, the less explicit information is necessary to describe it: Kolmogoroff complexity.
- Any entropy decrease due to a change to a higher abstraction level (ΔH_{view}) must be subtracted.
- Emergence $M = \Delta H_{emergence} = H_{max} - H - \Delta H_{view}$
- Relative Emergence m (for $\Delta H_{view} = 0$)

$$m = \frac{H_{max} - H}{H_{max}}$$

Information theory:

- Redundancy $R = H_{max} - H$
- Relative redundancy: $r = \frac{H_{max} - H}{H_{max}}$
- Why is $M = R$ (or $\Delta H_{emergence} = R$)?
- Answer: Different notions of utility
  - Information theory: A channel should be used only for the transmission of relevant (= new) information.
  - Max. information content if $H = H_{max} \rightarrow R = 0$
  - Newness or unpredictability of messages increase their value (information content).
  - The physical steps (light pulses, current levels, voltage levels) should be used economically only to transport relevant information.
  - Living (self-organising) systems are ”better“ if they display higher order, lower newness, higher predictability.

Redundancy conclusions:

- Communication engineer:
  Redundant information should not be transmitted over a channel.
  → Redundancy is ”bad“.

- Nature, living systems:
  The existence of predictable information about a system means that no (or little) new information has to be sent via the channel.
  → High predictability (= emergence = redundancy) is ”good“.

How to utilise emergence information?

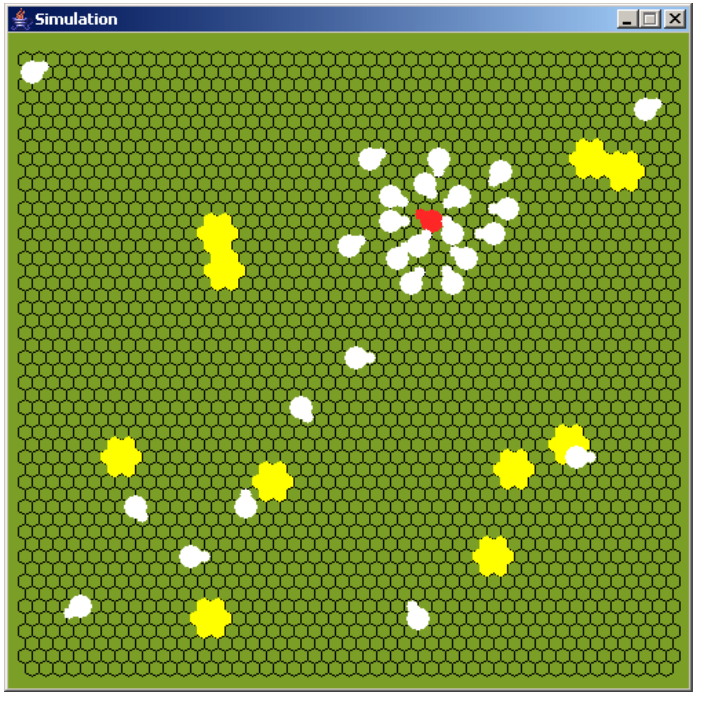
- Emergence can be calculated for a given system for different attributes.
- It can be used as an early indicator of (emergent) ordering processes.
- System emergence (the total of all attribute emergence values) is not selective enough.
- More interesting: Emergence fingerprint for all relevant attributes.
- Open questions:
  - Which attributes are relevant?
  - What is positive (wanted) and negative (unwanted) emergence?
  - How can we identify results of self-organised processes?

Example: cannibalistic behaviour of chicken

ORGANIC
COMPUTING

Problem

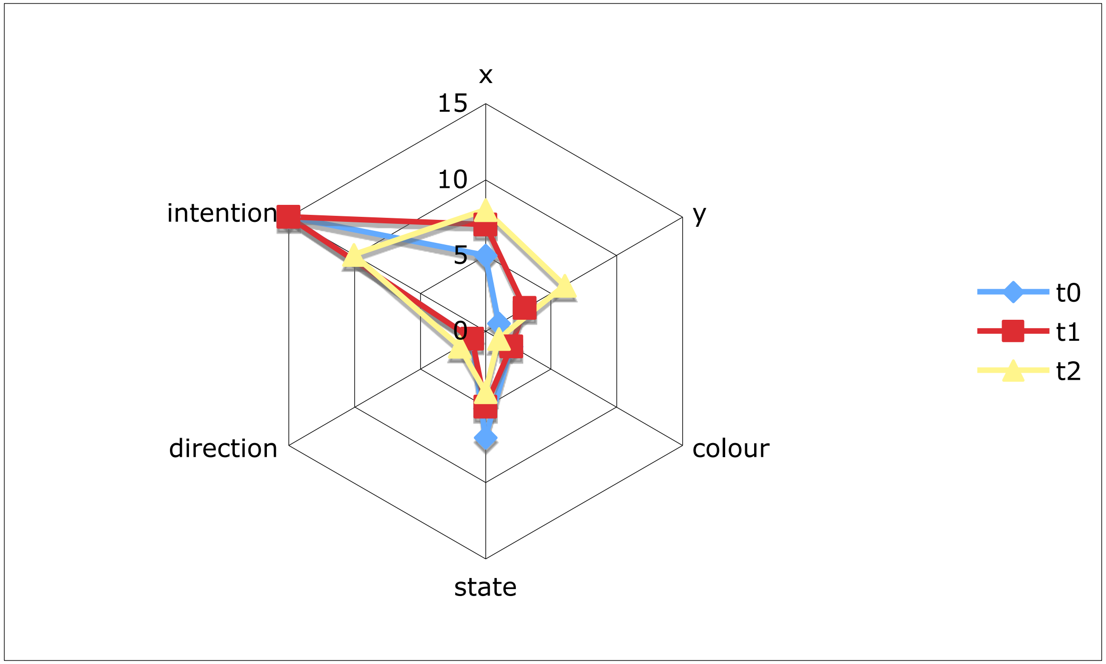
- TiHo Hannover
- Chicken stock in large farms
- Thousands of chicken in one shed
- Injured chicken:
  - Other chicken start to hurt injured chicken.
  - Do not stop until chicken is dead.
  - Even a small scratch causes dead!
  - Bad for chicken (→ dead) and owner (→ cost)
- What to do?
  - Noise disturbs chicken, they let up from injured chicken.
  - But: noise is bad (stress level)



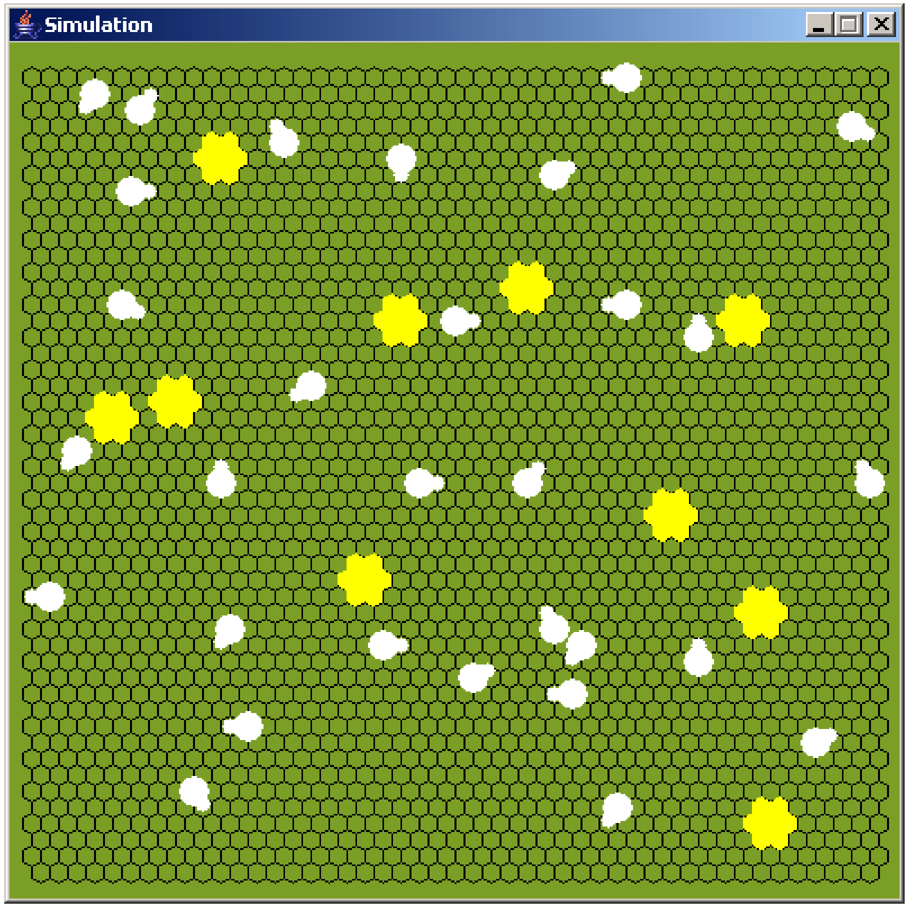
Yellow: food source
White: chicken with heading
Red: injured chicken

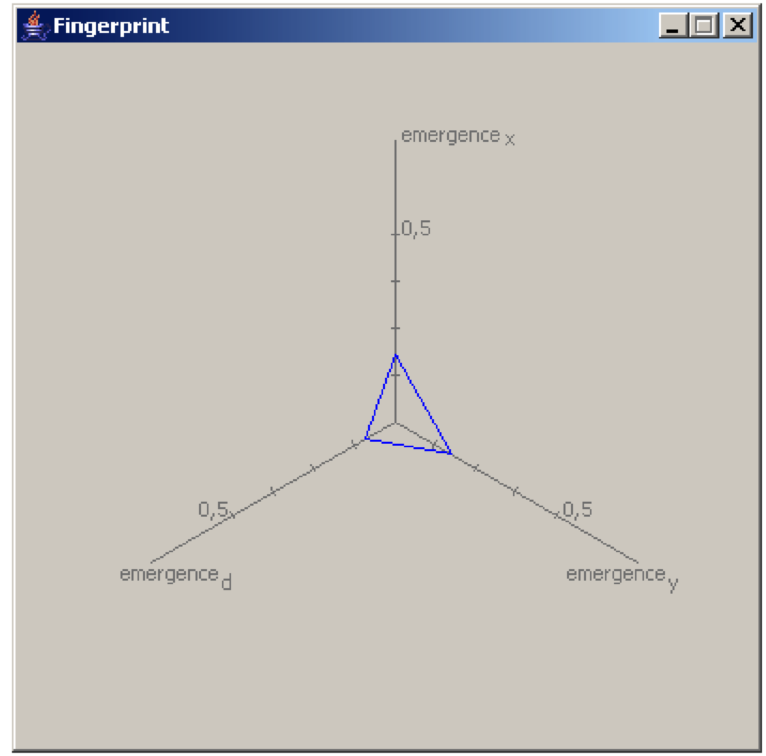
Emergence fingerprint

- Emergence fingerprint = visualisation of all (relevant) attribute emergence values of a system.
  - Visualisation as *n*-dimensional Kiviat graph
- Example
  - x-position, y-position, colour, state, direction, intention



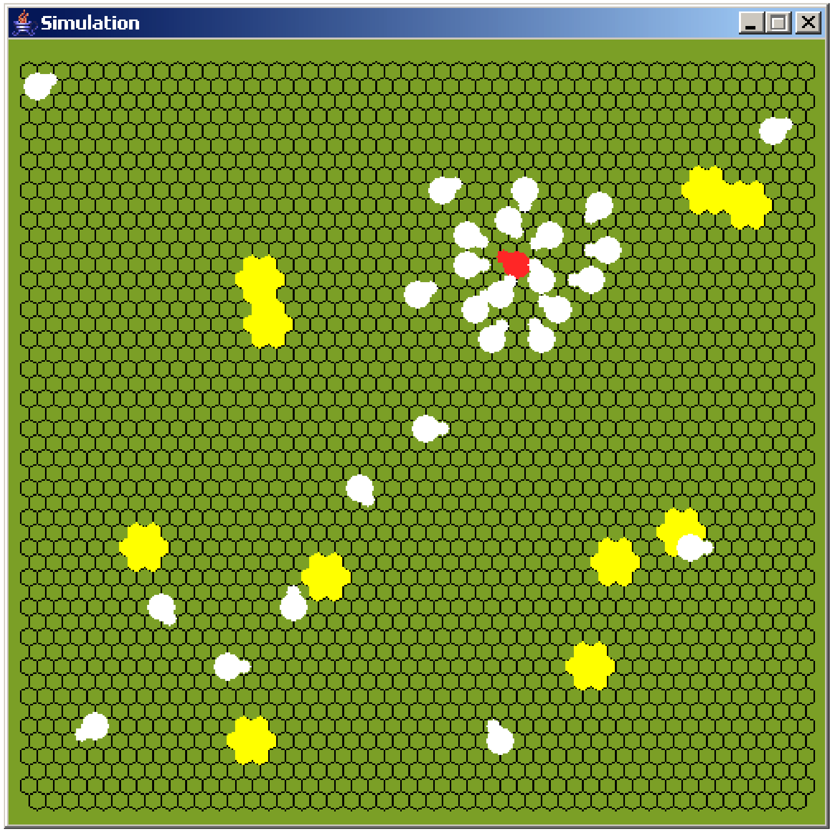
Emergence fingerprint (2)

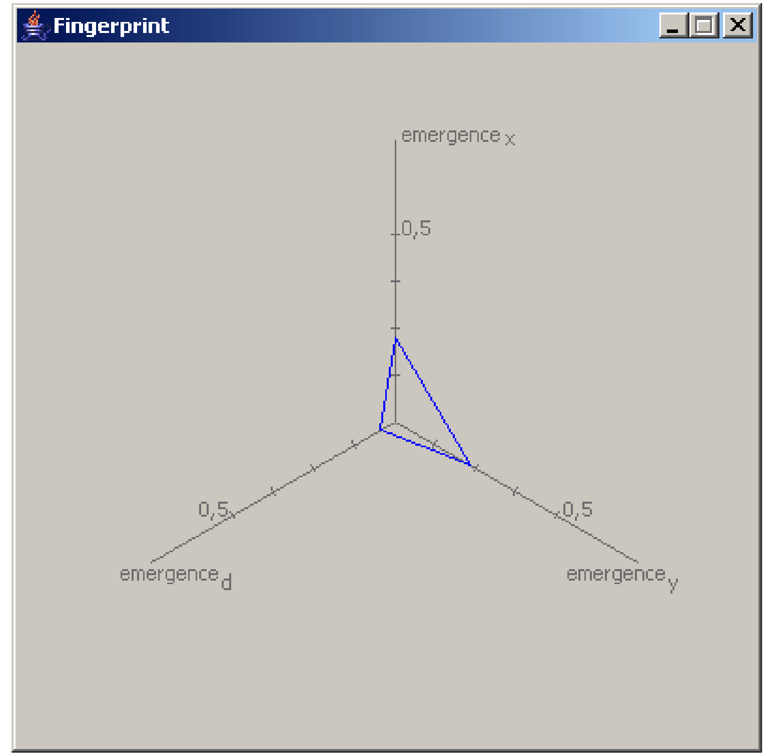




Pattern 1: $M_x = 0.181$, $M_y = 0.177$, $M_{direction} = 0.091$

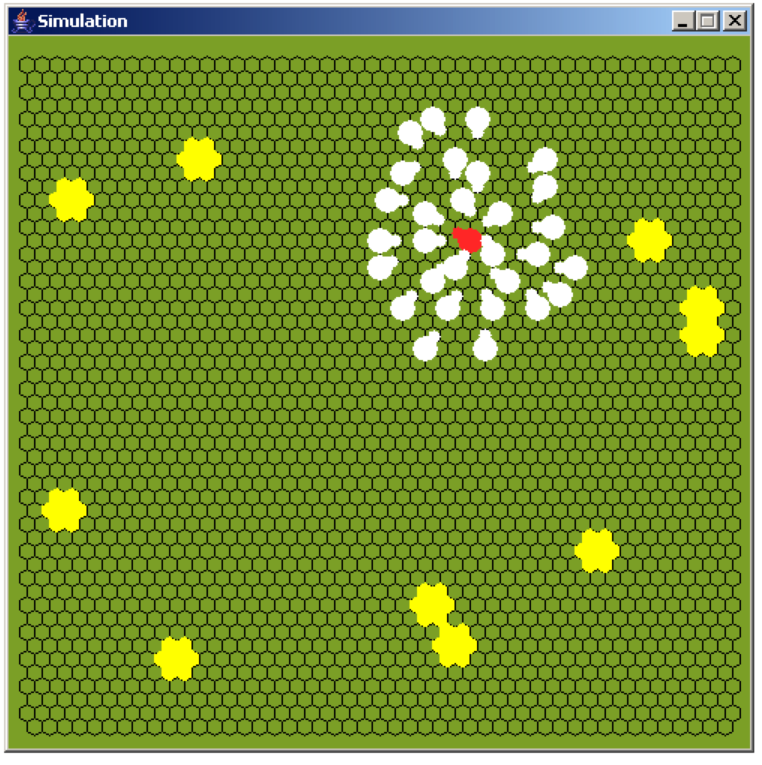
Emergence fingerprint (3)

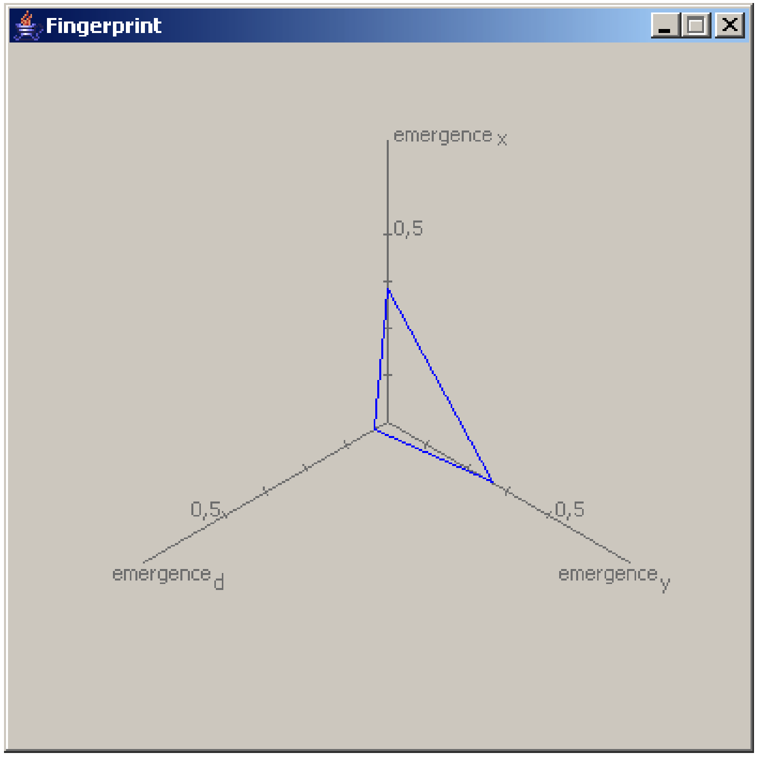




Pattern 2: $M_x = 0.226$, $M_y = 0.237$, $M_{direction} = 0.046$

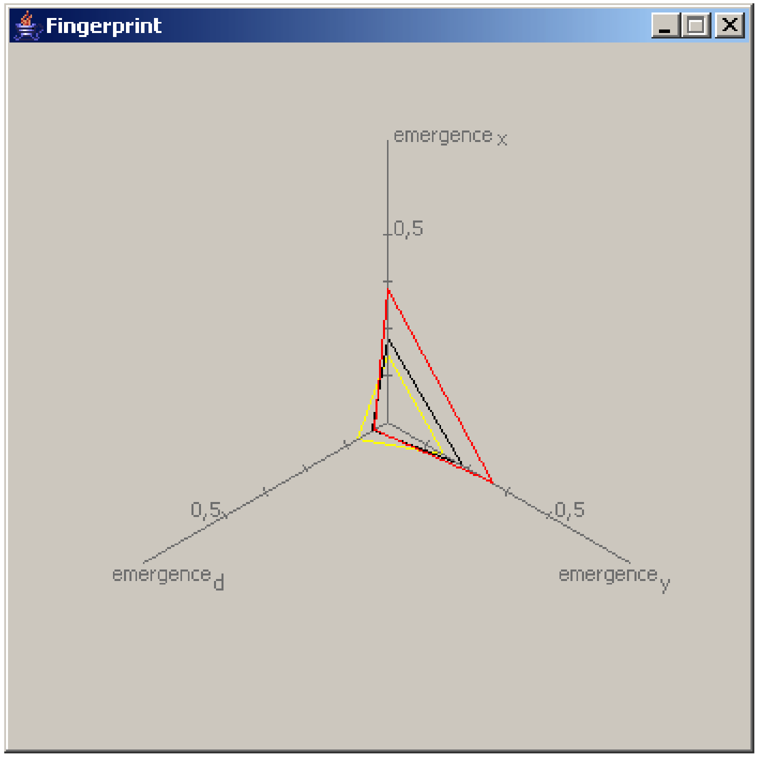
Emergence fingerprint (4)

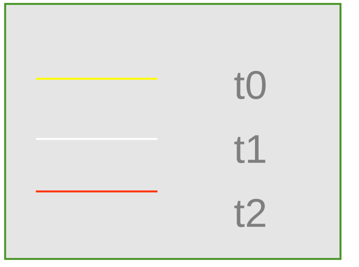




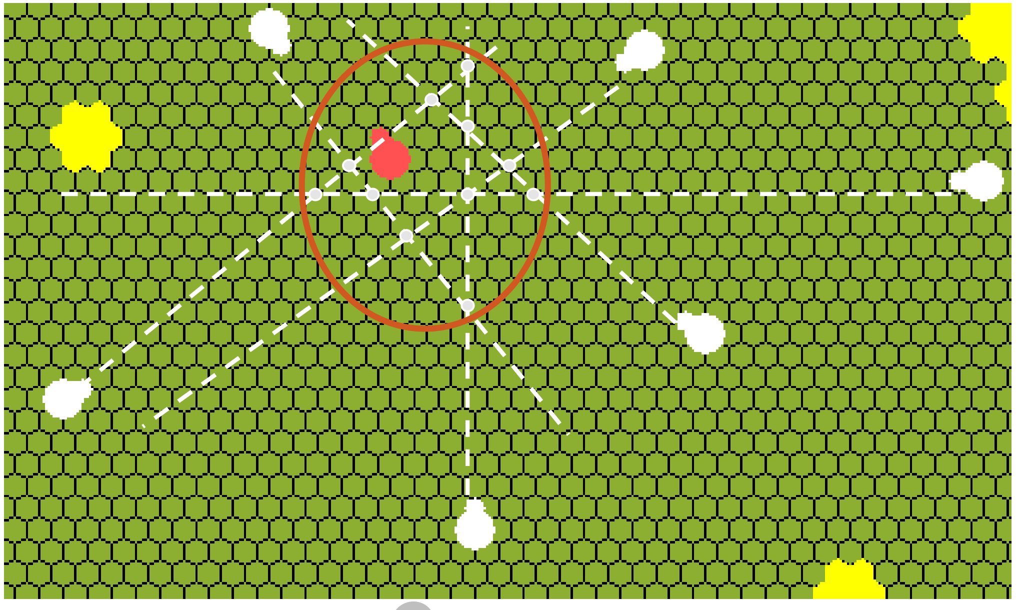
Pattern 3: $M_x = 0.359$, $M_y = 0.328$, $M_{direction} = 0.041$

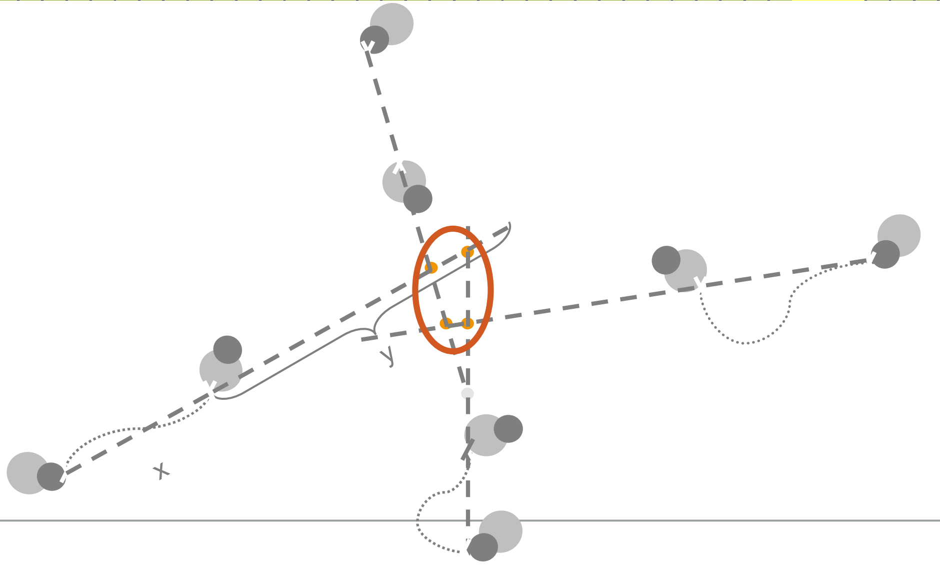
Emergence fingerprint (5)





Emergence fingerprint (6)





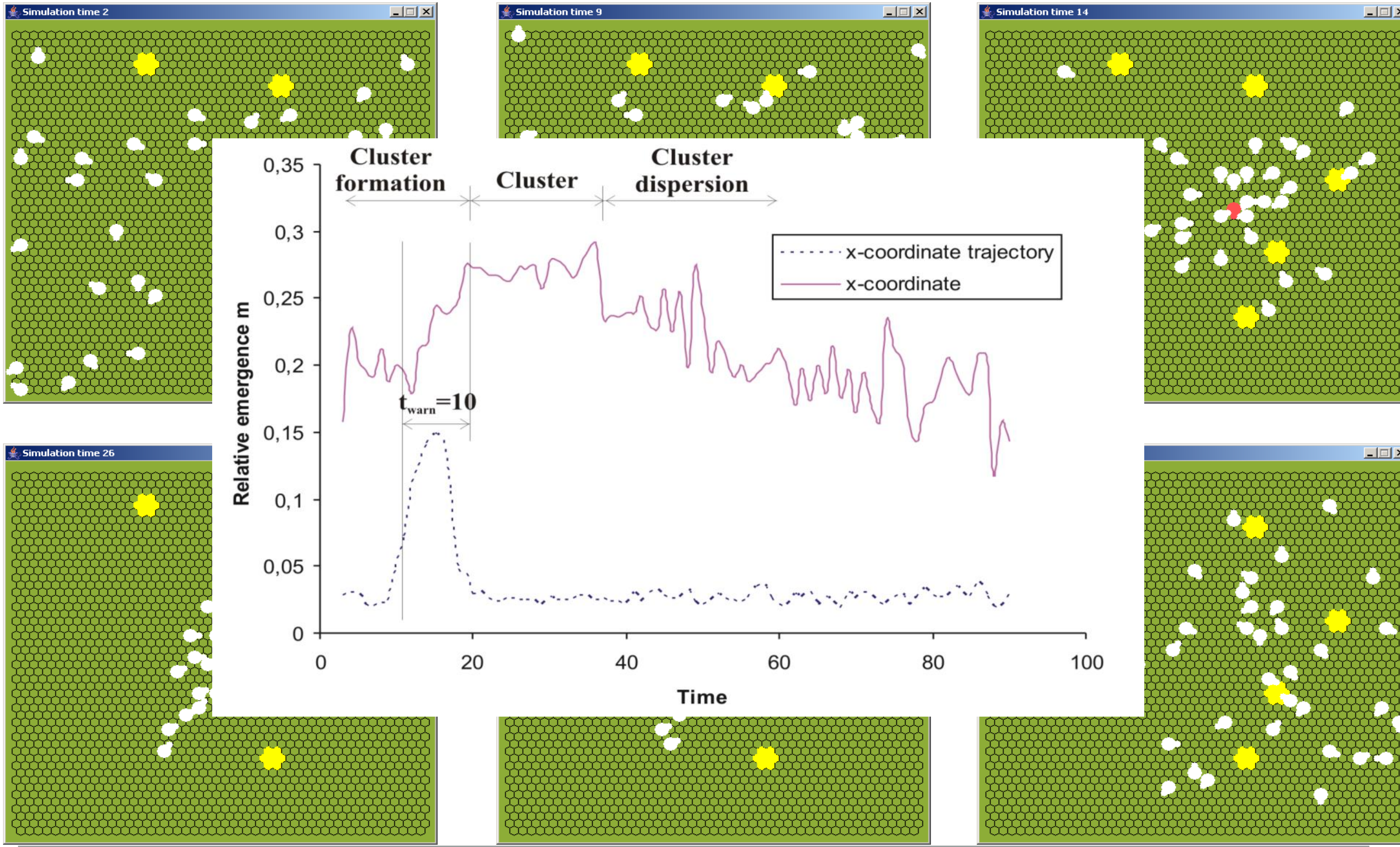
$$v = \frac{x}{\Delta t}$$

$$y = v \times \tau$$

Δt Observation period

τ Prediction period

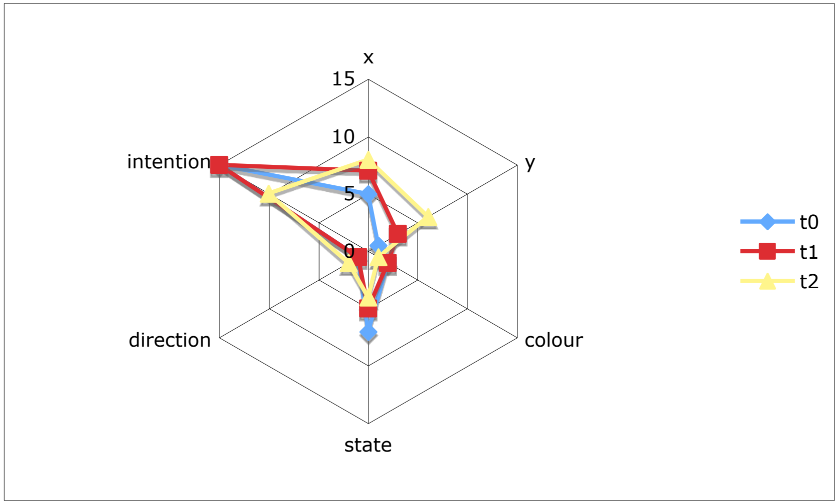
Cluster formation



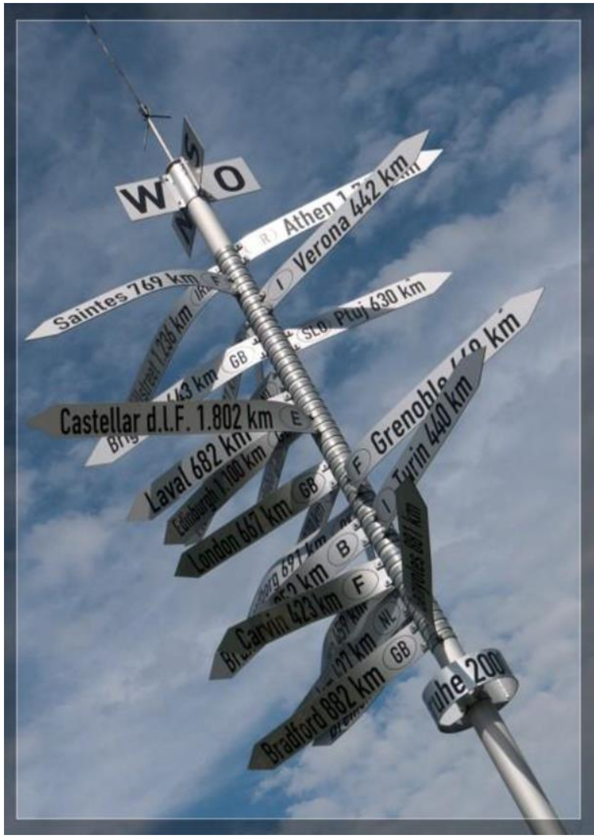
Summary: quantification of emergence

Process

1. Quantify entropy for each attribute.
2. Calculate emergence (M) for each attribute:
   $M = \Delta H = H_{Start} - H_{end} - \Delta H_{view}$
3. ΔH_{view} is the (possible) change of abstraction when observing H_{Start} and H_{end}, e.g. converting *float* to *int* values.
4. a) Determine the system emergence as sum over all attributes.
   b) Illustrate as “fingerprint”.
5. Is this due to self-organisation?



- A first example: water temples in Bali
- A second example: ants
- Emergence
- Term definition
- Quantification of emergence
- A refined approach to emergence quantification
- Conclusion and further readings



ORGANIC COMPUTING

Until now:

- Regarded emergence as the difference between an entropy at the beginning of some process and at the end.
- Discrete entropy difference (DED):

$$DED[x] = H_{start}[x] - H_{end}[x]$$

- A process is called emergent if DED[x] > 0 and the process is self-organised.
- What if we do not know if self-organisation is in place?
- Entropy values are computed for different attributes – which leads to a so-called emergence fingerprint – and this fingerprint serves as basis for certain decisions, e.g., concerning interactions with the system S.

Approach:

- We want to measure the amount of information we gain when we know that a categorical variable x has value i'.
- In a probabilistic framework: probability $p(x = i')$.
- Another unrelated, categorical attribute y and a value j': $p(y = j')$.
- This information measure has to be additive: If we knew the values of both attributes, the two information values are added.
- Hence: use $-\ln p(x = i')$ and $-\ln p(y = j')$

(which are always non-negative)

- If we observe both values, the amount of information for this observation of statistically independent variables gets:

$$-\ln \left(p(x = i', y = j')\right) = -\ln p(x = i') - \ln p(y = j')$$

From probabilities to entropy

- We are not interested in specific values of an attribute.
- Instead: We are interested in expected values.
- Hence: determine the expectation of the information with respect to the corresponding distribution.
- This is exactly the entropy, i.e., for a variable x with a corresponding distribution $p(x)$ we get:

$$H[x] = -\sum_x p(x) \ln p(x)$$

- Then: sum up over all possible values of x again.
- Entropy describes the expected amount of information which we gain when we observe x.

Limitations of the previous approach

Measure may be unsatisfying in some applications due to:

1. There are many attributes with continuous values in practical applications.
2. Many applications are multi-variate, i.e., based on several (categorical and continuous) attributes.

- The former problem (1) is solved by categorisation of continuous attributes.
  → Could be problematic as entropy measurements depend on size and position of the chosen “bins”.
- The latter (2) is solved by analysing the fingerprints.
  → If this analysis is conducted automatically, the different entropy values must be combined at some time.

Approach:

- Multivariate entropy measure for continuous variables.
- Combine all attributes into a vector x.
  Then: continuous entropy (also known as differential entropy) is:

$$H[x] = -\int p(x) \ln p(x) \, dx$$

- where p is the joint density of x.
- p combines all attributes, i.e. several continuous random variables.
- For simplicity: assume that we only have continuous variables.
  →Hybrid (categorical/continuous) approaches are possible.
- Please note: a continuous entropy (in contrast to a categorical one) may have negative values.

Density

- Approach relies on estimating the density of a continuous variable.
- Neglect (by now) the functional form of the density function
(e.g. to assume that it is Gaussian).
- Then: a non-parametric density estimation approach can be used.
- Assume: given a set X of N observations of x (i.e., samples): $x_0, \dots, x_{N-1}$.
- Goal: estimate $p(x')$ for arbitrary x' (not necessarily $x \in X$).
- Idea: count all samples in a certain environment around x' and divide this number by the size of the environment.

Density (2)

Alternative (smoother):

- Use Parzen window approach, i.e. a kernel density estimator based on a Gaussian kernel:

$$p(x') \approx \frac{1}{N} \sum_{x_n \in X} \frac{1}{(2\pi h^2)^{\frac{D}{2}}} \exp(-\frac{1}{2} \frac{||x' - x_n||^2}{h^2})$$

- where D is the dimensionality of x and h is a user-defined parameter.
- h depends on the data set X – there are a number of heuristics to estimate h (e.g. h is set to the average distance of the ten nearest neighbours from each sample, averaged over the entire data set).

Evaluation of the integral

- Continuous entropy model contains integral.
  → How to evaluate this?
- Remember: data set X contains samples x_n distributed according to p (i.e., $x_n \sim p$).
- Hence: Entropy can be approximated

$$\hat{H}[x] \approx -\frac{1}{N} \sum_{x_n \in X} \ln p(x_n)$$

- where the $p(x_n)$ are estimated using the Parzen approach.
- Note: this discrete approximation of the entropy does not sum up over discrete points in the input space situated on a regular grid.
- Hence: take their non-uniform distribution into account by a correcting factor $\frac{1}{P(x_n)}$.
  → Corresponds to the concept of importance sampling.

From entropy to emergence

- The static approach defines emergence using a difference of entropy values.
  → Emergence is considered as a change of order within a system.
- Here, we define emergence as an unexpected or unpredictable change of the distribution underlying the observed samples.
- Then: use divergence measure to compare two density functions,
  i.e. p(x) at t_0 and $q(x)$ at t_1.
- Possible measure is Kullback-Leibler (KL) divergence $KL(p||q)$.
- Also known as relative entropy.
- Compares two probability density functions.

$$KL(p||q) = -\int p(x) \ln \frac{q(x)}{p(x)} dx$$

Limitations of Kulback-Leibler

- KL divergence is not a true metric since it is not symmetric.
- However:
  - $KL(p||q) \geq 0$ and
  - $KL(p||q) = 0$ only if $q(x) = p(x)$.

- We measure the expected amount of information contained in a new distribution with respect to the original distribution of samples and not with respect to the new distribution:

$$KL(p||q) = -\int p(x) \ln q(x) \, dx + \int p(x) \ln p(x) \, dx$$

- There are concepts for symmetric variants (neglected here).

Application of the measures

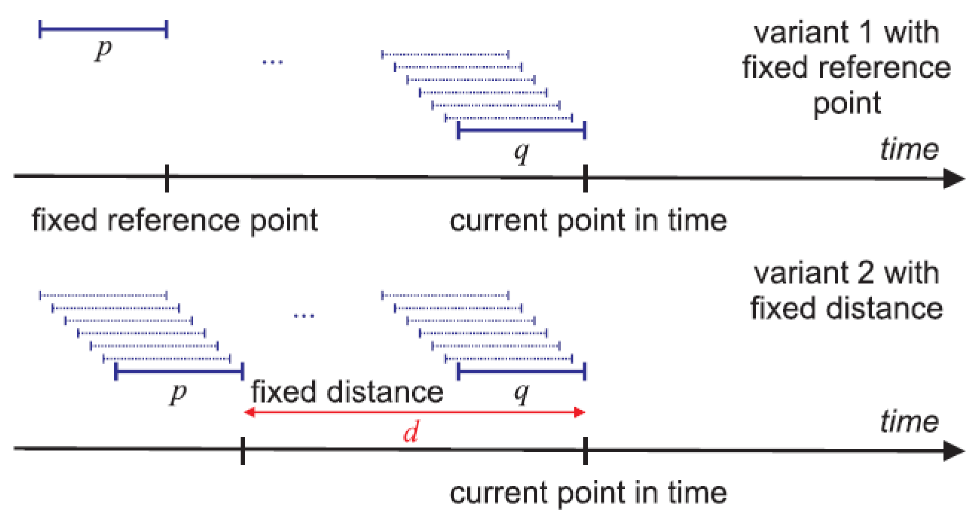
Measures are applicable to emergence quantification:

- Abstraction from the technical system.
- Consider only distributions of samples in the attribute space.
- Assumption: observation of a number of processes “generating” samples.
- Goal: comparison of the distributions underlying the observed samples.
- Concept: estimation of the distributions at two different points in time, an earlier one (p) and a later one (q).
- Instead of assuming that we get a set of observations at each (discrete) point in time: one single observation at each point in time (these points are considered as equidistant in time).

Application of the measures (2)

Sliding window:

- Estimate p and q in sliding data windows.
- Windows have fixed length, must be:
  - long enough to estimate p and q with sufficient reliability.
  - short enough to allow for the assumption that the observed processes are nearly time-invariant in these windows.
- Distinguish:
  - First (earlier) time interval is fixed at a certain point in time, whereas the second interval moves along the time axis with the current point in time. → Online application.
  - Both windows move along the time axis in a fixed temporal distance. → Distance d is important parameter of the measure-ment technique.



Application of the measures (3)

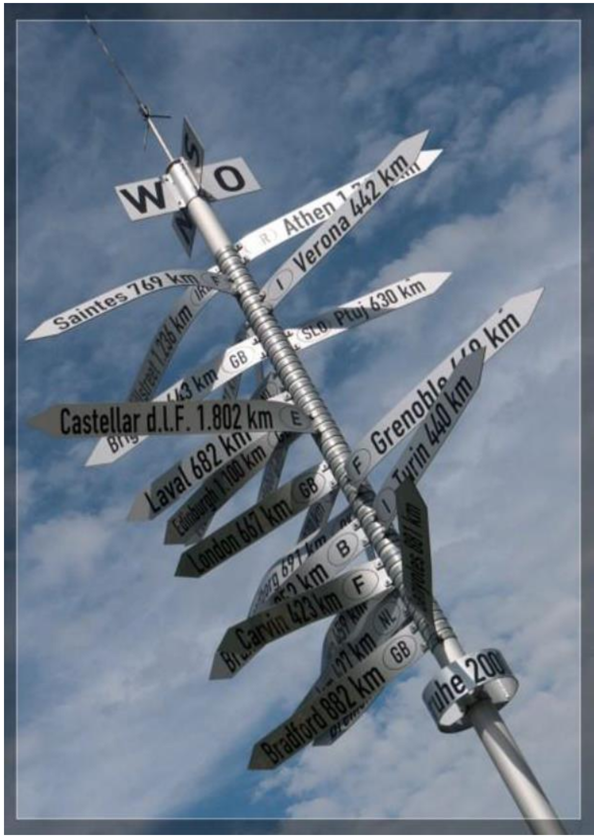
- Estimation of densities p and q: non-parametric or model-based approaches.
  → Depending on application.
- Hybrid approaches are possible as well.
- If both densities are estimated in a non-parametric approach:
  → Either using the sampling points in the first set of observations ($x_n \sim p$) or those given in the second ($x_n \sim q$).
- Suggestion: Evaluate both intervals and average measures.
  → Get more robust estimates.
- Comparison leads to 'degree' of emergence. In addition:
  - Detection of processes that disappear (i.e., components become obsolete).
  - Detection of newly emerging processes (i.e., new components are required).
    → Novelty detection.
  - Detection of components that change their characteristics (i.e., components change their parameters such as centre or mixing coefficients).
    → Concept drift.

Emergence detection

- Based on probabilistic (or information-theoretic) considerations.
- May be used to determine ‘degree‘ of emergence.
- Contrast to previous approach:
  - Applicable in cases with continuous attributes,
  - Applicable if several attributes have to be combined,
  - Applicable if application allows for model-based density estimates.
- Measures can assess emergence gradually.
- Can further be used to detect novel situations or phenomena such as concept drift.
- In organic systems:
  - Monitor the overall distribution by combining measures for different components.
  - Supervise components individually.

Agenda

- A first example: water temples in Bali
- A second example: ants
- Emergence
- Term definition
- Quantification of emergence
- A refined approach to emergence quantification
- Conclusion and further readings



From self-organised order to emergence

- Nature as inspiration: Complexity is handled by self-organised order.
- Order is observer- and goal-dependent!
- Self-organised order consists of purposeful self-organisation processes and additional emergent phenomena.
- Same ingredients in organic systems → same processes expected!
- Consequence: We have to measure and master emergence.
- Approach:
  - Observe behaviour of system
  - Measure order (i.e. based on entropy)
  - Compare measures at different points.

Conclusion (2)

This chapter:

- Demonstrated how self-organised order appears in natural, technical and social systems.
- Highlighted the control of complexity by self-organisation and emergence.
- Defined the term ‘emergence’ and its relation to self-organisation.
- Explained how emergence is quantification for systems with discrete attributes.
- Refined this quantification concept to be applicable to continuous attributes and their combinations.

By now, students should be able to:

- Explain the relation between self-organisation and emergence.
- Briefly summarise the term emergence.
- Give examples for emergent phenomena, e.g. in nature.
- Quantify emergence in technical systems based on discrete attributes.
- Outline how emergence detection is done for systems with continuous attributes.

Further readings

ORGANIC
COMPUTING

- Steven Johnson: „Emergence – The connected lives of ants, brains, cities, and software“, Scribner publishers, New York, 2001.
- Nelson Fernandez, Carlos Maldonado, Carlos Gershenson: „Information Measures of Complexity, Emergence, Self-organisation, Homeostasis, and Autopoiesis“, online available at: http://arxiv.org/pdf/1304.1842v1.
- Moez Mnif and Christian Müller-Schloer: “Quantitative Emergence”, in: “Organic Computing - A Paradigm Shift for Complex Systems, pages 39 - 52, 2011, Birkhäuser Verlag, Basel, CH. DOI: 10.1007/978-3-0348-0130-0_2
- Dominik Fisch, Martin Jänicke, Bernhard Sick and Christian Müller-Schloer, "Quantitative Emergence - A Refined Approach Based on Divergence Measures," 2010 Fourth IEEE International Conference on Self-Adaptive and Self-Organizing Systems, Budapest, 2010, pp. 94-103. DOI: 10.1109/SASO.2010.31
- Deborah Johnson: “Ants At Work: How An Insect Society Is Organised“. Free Press 2011, New York (USA) and London (UK), ISBN: 978-1451665703.

Questions …?