UNIVERSITÄT PASSAU
Fakultät für Informatik und Mathematik

Modeling Self-Organizing Systems

Hermann de Meer, Richard Holzer, Patrick Wüchner

Chair for Computer Networks & Computer Communications
University of Passau
Germany

Computer Networks
& Communications
Prof. Hermann de Meer

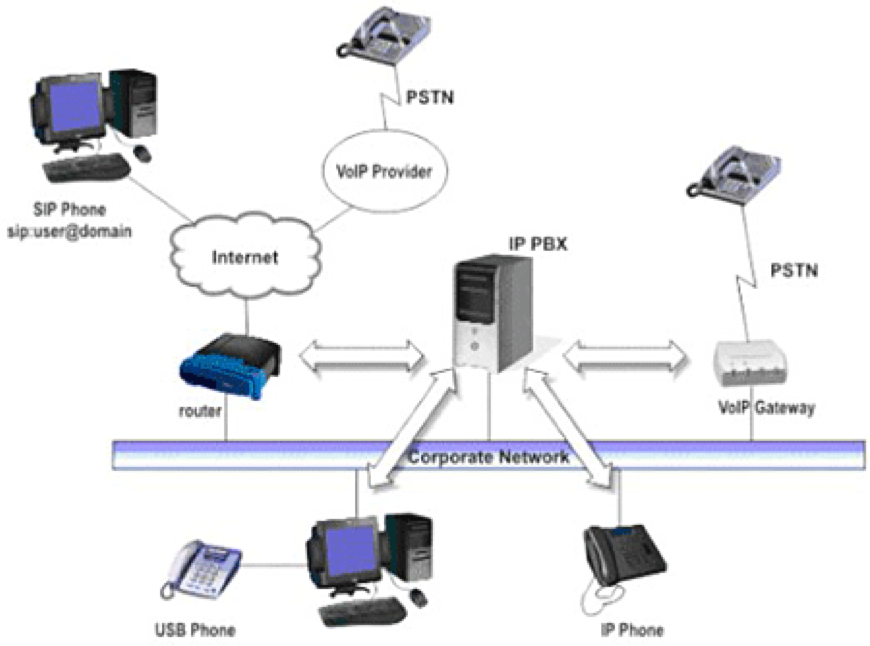
Overview

UNIVERSITÄT PASSAU
Fakultät für Informatik und Mathematik

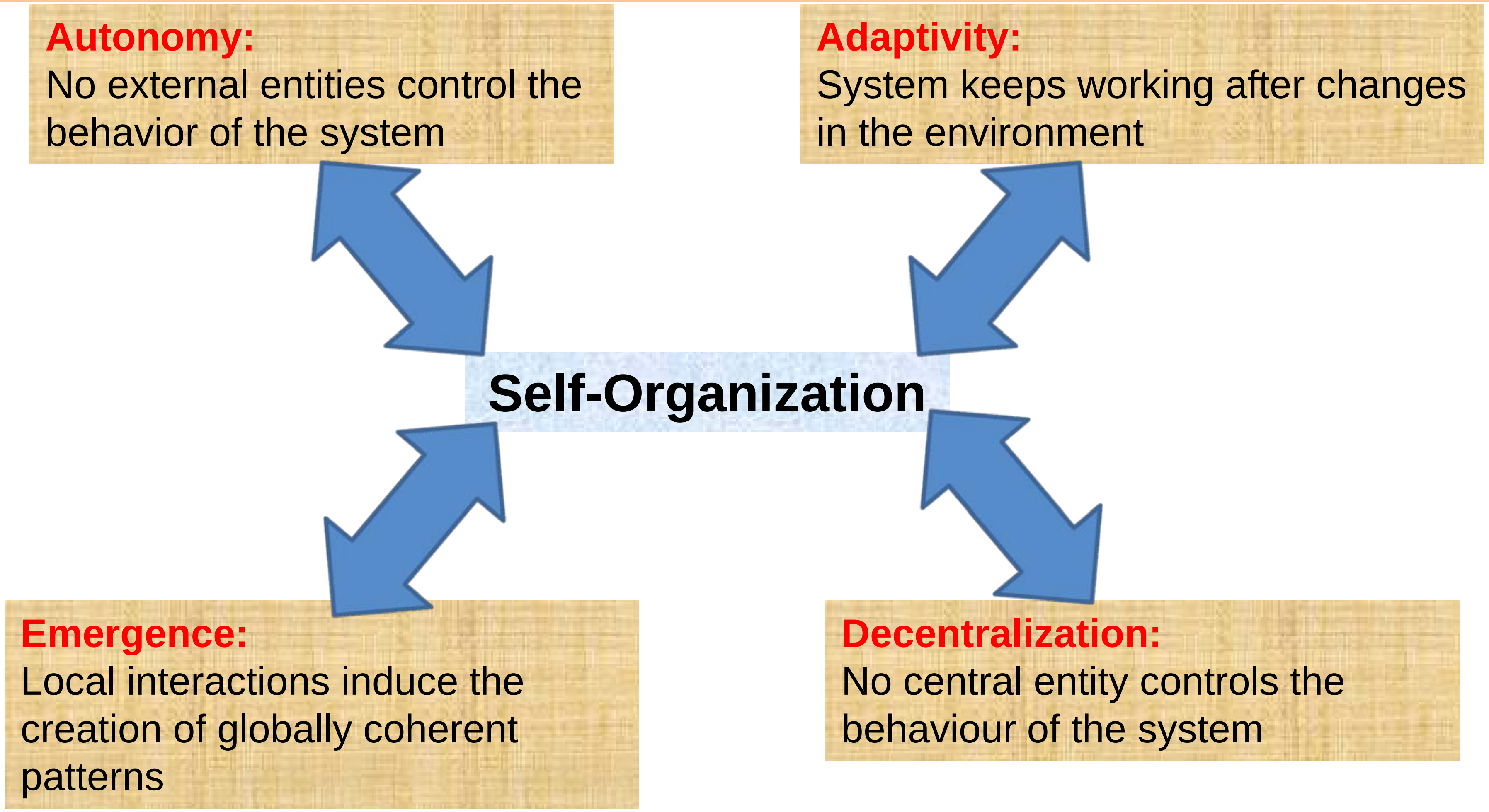
1. Systems
2. Discrete versus continuous
3. Macro level modeling
4. Micro level modeling
5. Deriving local interaction strategies
6. Quantitative measures
7. Example: Synchronization of sensor nodes
8. Example: Evacuation

What is a complex system?

- A **complex system** consists of a large number of interacting entities.
- **Local rules**, which describe the behavior of each entity in the system, lead to complex global states.
- A self-organizing system exhibits **emergence**: A global pattern in the collective behavior, which results from the local interactions.
- The emergent behavior does not result from the existence of a central controller.



Self-Organizing Systems



Autonomy:
No external entities control the behavior of the system

Adaptivity:
System keeps working after changes in the environment

Self-Organization

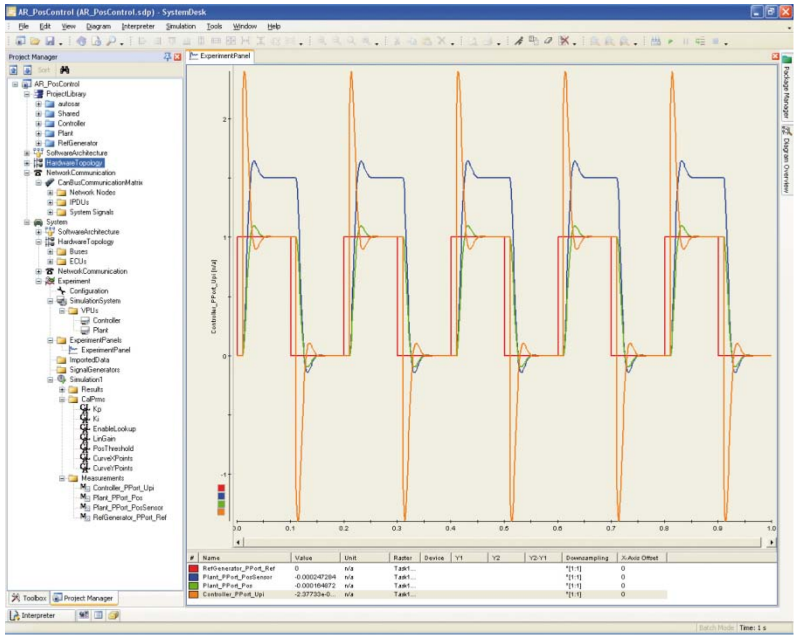
Emergence:
Local interactions induce the creation of globally coherent patterns

Decentralization:
No central entity controls the behaviour of the system

Difference between model and simulation

For a **simulation**,

- the **behavior** of each entity must be described,
- the algorithms of the behaviors of all entities in the system are **implemented** as good as possible (with respect to the relevant parameters/properties),
- system properties can be analyzed by different „**simulation runs**“,
- the **derived properties** correspond to the individual cases of the simulation runs.

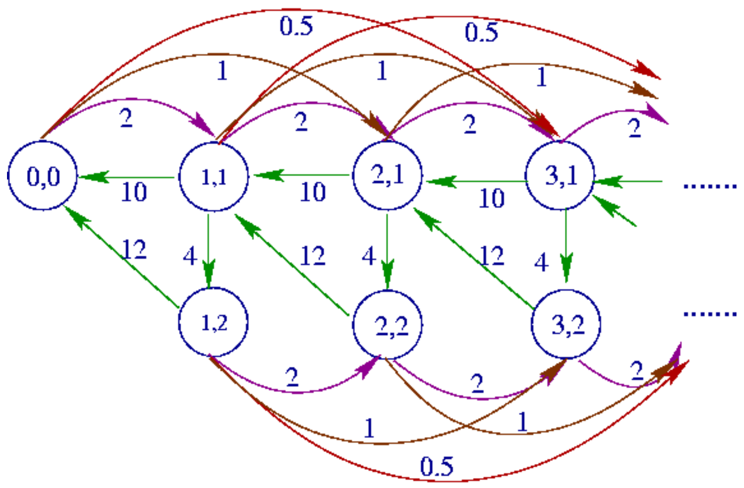

AR_PosControl (AR_PosControl.sdp) - SystemDesk
File Edit View Diagram Interpreter Simulation Tools Window Help
Project Manager
ExperimentPanel
Controller_PPort_Upi
Toolbox
Project Manager
Interpreter

#	Name	Value	Unit	Raster	Device	Y1	Y2	Y2-Y1	Downsampling	X-Axis Offset
🟥	RefGenerator_PPort_Ref	0	n/a	Task1...					*[1:1]	0
🟦	Plant_PPort_PosSensor	-0.000247284	n/a	Task1...					*[1:1]	0
🟩	Plant_PPort_Pos	-0.000164872	n/a	Task1...					*[1:1]	0
🟧	Controller_PPort_Upi	-2.37733e-0...	n/a	Task1...					*[1:1]	0

Difference between model and simulation

A formal **model** is a representation of the system, which

- **simplifies** the system on the relevant properties and parameters,
- is **mathematically formalized**,
- allows the mathematical derivation of **global properties** from the local rules.



Discrete or Continuous?

UNIVERSITÄT PASSAU
Fakultät für Informatik und Mathematik

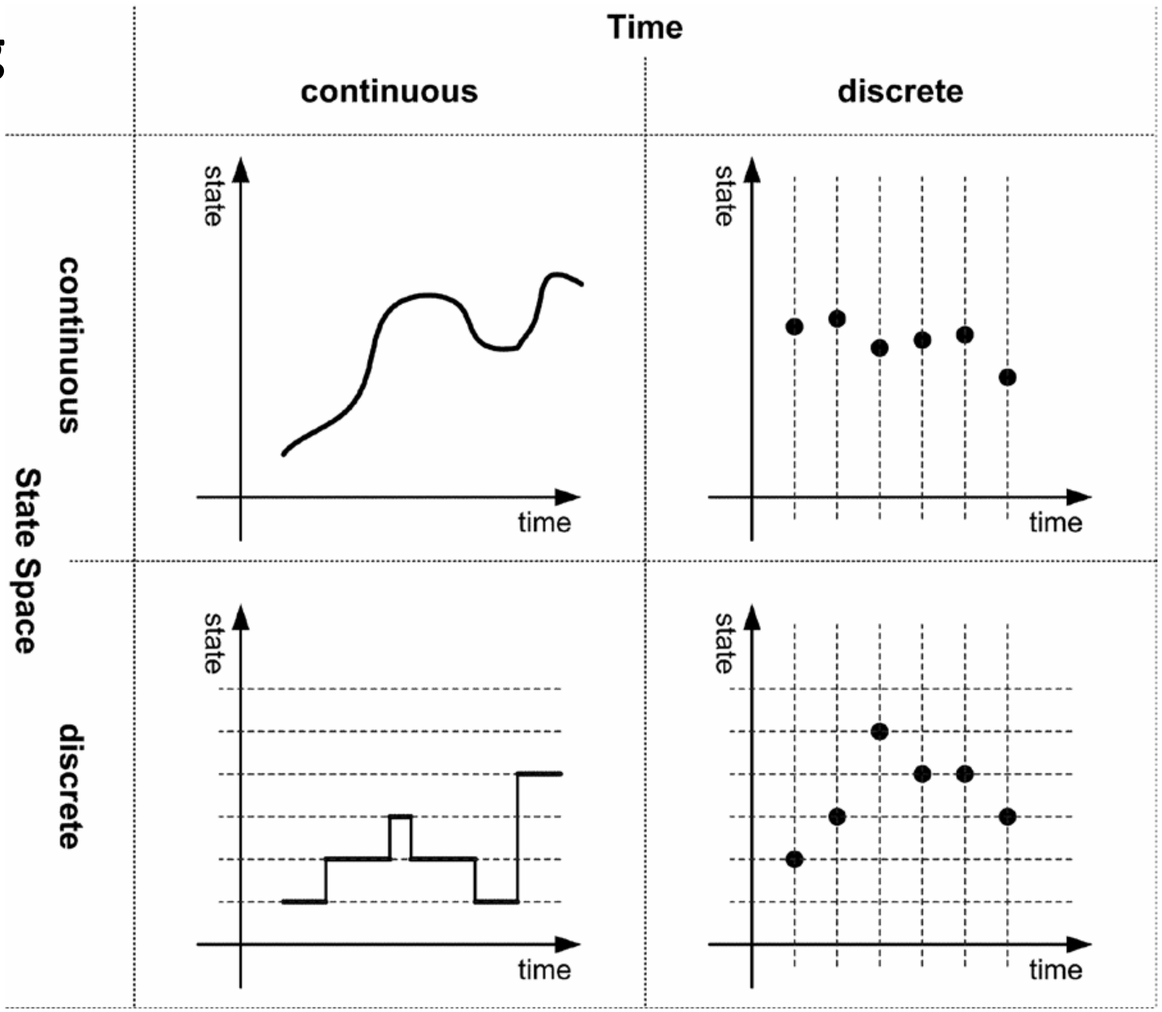
Macro level modeling

- Discrete/continuous time
- Discrete/continuous state space

	Discrete	Continuous
Time	Time increases in discrete steps. This could also be event based: Each step corresponds to one event.	Continuous change of the system can happen everytime
States	At each point of time there are only finite or countable many possibilities for the current state.	Uncountable state space.

Discrete or Continuous?

Macro level modeling



Discrete or Continuous?

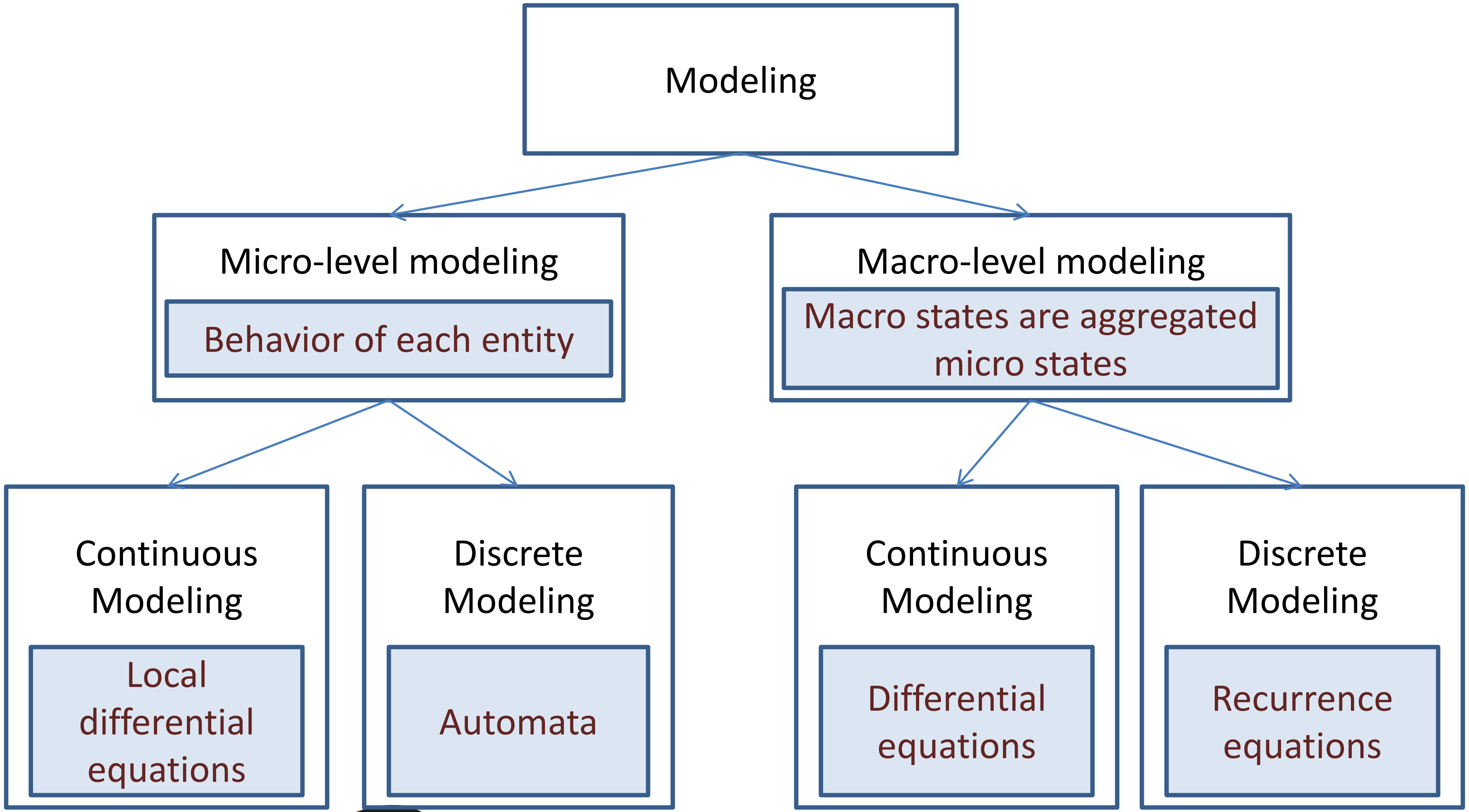
Micro level modeling

	Discrete	Continuous
Time	Time increases in discrete steps. This could also be event based: Each step corresponds to one event.	Continuous change of the system can happen everytime
Object set	Only a finite number of objects are modeled	Uncountable number of objects
States	Finite memory: Each object is in one of a finite number of states	Uncountable memory for each object
Interaction	Each object can interact with a finite number of other objects	Uncountable many interactions

Questions before the design of the model:
- Which system do I want to model?
- Which properties should be investigated?

⇒ Decision, whether discrete or continuous modeling fits better

Modeling approaches

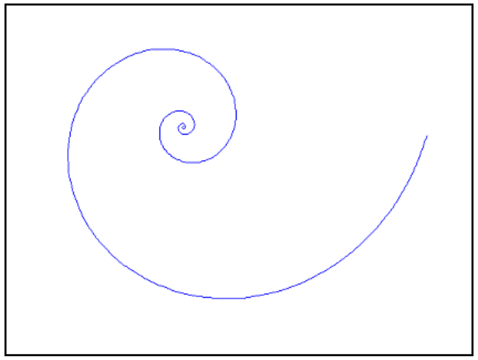


Macro Level Modeling

A **dynamical system** consists of

- a state space S,
  - The state space contains all possible states of the system.
  - If more than one value is needed to describe the state, then it can be of higher dimension, e.g., each state s = (s_1, …, s_n) 2 S is a tuple.
- a time variable t,
- an evolution law, which descibes the change of the state over time.

The orbit of the system is the path of the state over time.



Dynamical System

In a **continuous system**,

- the time variable is a real number,
- the evolution rule is a differential equation:

$$\frac{dx}{dt} = \dot{x} = X(x)$$

In a **discrete system**

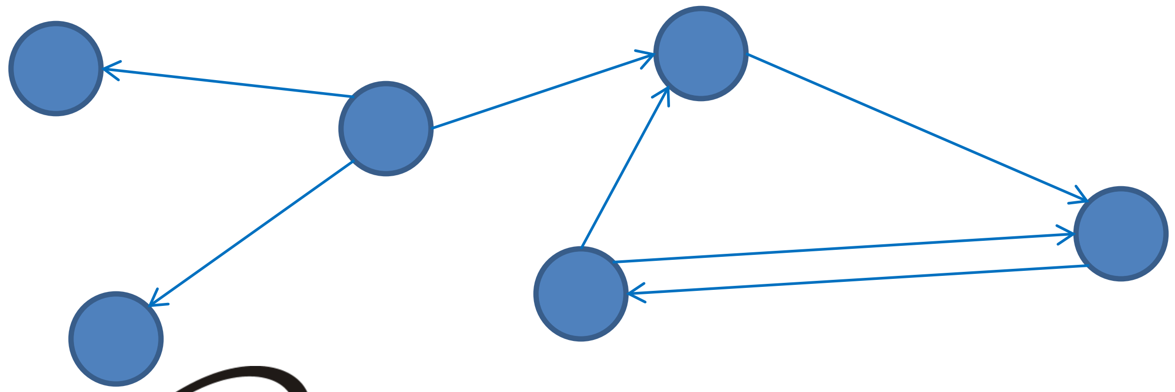
- the time variable is an integer,
- the evolution rule is a recurrence equation:

$$x_{t+1} = f(x_t)$$

Discrete Micro-Level Modeling

Model for **topology** in discrete micro-level model: **directed graph**

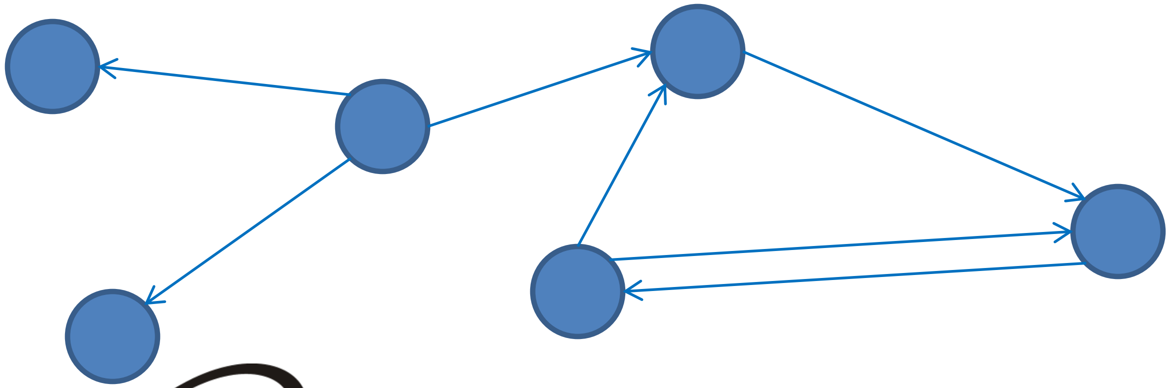
- Each node of the graph corresponds to one object of the system
- Each edge of the graph models interaction (e.g., exchange of data) between objects



Discrete Micro-Level Modeling

Model of **behaviour** of each node: **Stochastic automaton**

- The automaton receives **local input** from predecessor nodes.
- At each point in time, the automaton has an **internal state.**
- The automaton decides nondeterministically about the **change of state** and about the **local output** to the successor nodes.



A Generic Method to Derive Local Interaction Strategies

- To achieve a preferred global behavior of self-organizing systems, suitable **local interaction strategies** have to be found.
- A **general method** has been developed that allows to systematically derive local interaction strategies by specifying the preferred global behavior.

A Generic Method to Derive Local Interaction Strategies

Generic method:

- Model the system.
- Specify the **Laplace's Demon**: One entity of the system is chosen to be the LD and is equipped with global knowledge.
  - The LDs reactions can be specified for each opponent behaviour
- System **simulation** and recording of the LD's behaviour in each time step
  - Other entities deliver input from alphabet I
  - LD reacts with output from alphabet O
  - For each time step t a state from (I_{t-1}×O_t) is recorded

A Generic Method to Derive Local Interaction Strategies

- The state series is converted by the **Causal-State Splitting Reconstruction** (CSSR) algorithm
  - Result: Hidden Markov Model, which approximates the Input/Output states
- The hidden Markov Model can be used as a **local interaction strategy** for the entity by considering only those transitions that match the encountered input.

The proposed method was proven to be applicable in a game-theoretic setting, where strategies in the iterated prisoner's dilemma could be obtained.

From Micro-level to Macro-level: Quantitative Measures

To measure self-organizing properties of a system, we need to determine the quantity of information in the system.

Entropy of a random variable X:

$H(X) = - \sum P(X=w) \log_2 P(X=w)$

H(X) is the average number of bits for an optimal encoding of the information of X.

With this concept we can measure for each point of time

- the information in the whole system
- the information on the internal edges
- the information on the input edges
- the information on the control edges
- the information on the output edges

Quantitative Measures

Quantitative measures can be obtained (partly based on information entropy):

Levels of

- emergence
  - How many globally coherent patterns are induced by local interactions?
- autonomy
  - How much control data from external entities are needed to keep the system running?
- target orientation
  - Is the high level goal, that the system designer had in his mind, reached by the system?
- adaptivity
  - Is the high level goal still reached after changes in the environment?
- resilience
  - Is the high level goal still reached after unexpected impacts on the system (e.g. break down of nodes, attacks by an intruder, …)?
- homogeneity
  - Do all nodes have the same behavior?
- global state awareness
  - How much information does a single node have about the global state?

Emergence

UNIVERSITÄT PASSAU
Fakultät für Informatik und Mathematik

To measure the **level of emergence**

ε 2 [0, 1]

of a system, we compute the dependencies between the values on the edges e 2 E.

At time t we compare the information contained in all edges with the sum of the information contained in each single edge:

$$\varepsilon_t = 1 – (H(\text{values on all edges}) / \sum_{e2E} H(\text{value on edge e}))$$

Level of emergence of the whole system S:

$\varepsilon(S) = \text{Avg}(t \mapsto \varepsilon_t)$, where Avg is the average value of the map

ε ¼ 1	high level of emergence	(many dependencies)
ε ¼ 0	low level of emergence	(few dependencies)

Autonomy

UNIVERSITÄT PASSAU
Fakultät für Informatik und Mathematik

To measure the **level of autonomy**

$\boldsymbol{\alpha}$ 2 [0, 1]

at time t, we compare the information contained in the **control edges** c 2 C with the information contained in **all edges** e 2 E during the whole run of the system:

$$\alpha_t = 1 – (\text{H(values on C) / H(values on E)})$$

Level of autonomy of the whole system S:

$$\alpha(S) = \text{Avg}(t \mapsto \alpha_t)$$

α ¼ 1	high level of autonomy	few control data
α ¼ 0	low level of autonomy	much control data

Target orientation

Before a new system is designed, we have a goal of the system in our mind:

The system should fulfill a given purpose.

In the model, the goal can be described by a valuation of configurations:

b : Conf ! [0, 1]   (Conf is the set of all global states)

Level of target orientation at time t:

$TO_t = \boldsymbol{E}(b(Conf_t))$,   where ***E*** is the mean value of the random variable

Level of target orientation of the whole system S:

$TO(S) = Avg(t \mapsto TO_t)$

TO(S) ¼ 1 means that the system runs through many good configurations

) high level of target orientation

TO(S) ¼ 0 means that the system runs through many bad configurations

) low level of target orientation

Resilience

There are different forms of resilience for computer networks:

- Resilience with respect to malfunctioned nodes
- Resilience with respect to attacks by an intruder, who is inside the network
- Resilience with respect to attacks by an intruder, who is outside the network
- Resilience with respect to natural disasters or other external influence, which might cause a breakdown of some nodes

How can we define Resilience in the model?

Resilience

UNIVERSITÄT PASSAU
Fakultät für Informatik und Mathematik

Idea: Replace the automatons of the malfunctioned nodes by new automatons.

If the behavior of a malfunctioned node v is not known in advance, use a parameter θ to define different possible behaviors for the new automaton.

The new goal can be described by a valuation of configurations in the modified system S_θ:

b_θ : Conf ! [0, 1]

Level of resilience at time t:

$\mathrm{Res}_t = \boldsymbol{E}(b_\theta(\mathrm{Conf}_t))$

Level of resilience of the whole system S:

$\mathrm{Res}(S) = \mathrm{Avg}(t \mapsto \mathrm{Res}_t)$

Res(S) ¼ 1 means, that the new system runs through many good configurations) high level of resilience

Res(S) ¼ 0 means, that the new system runs through many bad configurations) low level of resilience

Adaptivity

A system is adaptive, if it can fulfill its task despite of changes in the environment.

Control edges: C

Controling nodes of the environment: C- = { v 2 V | (v, w) 2 C for some w 2 V }

Change in the environment: Replace the automatons of C- by new automatons

If the new behavior of a node v 2 C- is not known in advance, use a parameter θ to define different possible behaviors for the new automaton.

The new goal can be described by a valuation of configurations in the modified system S_θ:

b_θ : Conf ! [0, 1]

Level of adaptivity at time t:

$Ad_t = \boldsymbol{E}(b_\theta(Conf_t))$

Level of adaptivity of the whole system S:

$Ad(S) = Avg(t \mapsto Ad_t)$

Ad(S) ¼ 1 means, that the new goal is reached despite the changes in the environment (the new system runs through many good configurations)

Ad(S) ¼ 0 means, that the goal is not reached after the changes (the new system runs through many bad configurations)

Homogeneity

How can we measure homogeneity?

Consider the local view of a node v:

At the current point of time t, the node v sees not the whole configuration $Conf_t$ but only the local configuration $Conf_{t,v}$, that is visible at v: The internal state of v and the values of the edges of v.

Level of homogeneity at time t:

$$Ho_t = 1 - \frac{\sum_{v,w \in V, v<w} |H(Conf_{t,v}) - H(Conf_{t,w})|}{\sum_{v,w \in V, v<w} \max(H(Conf_{t,v}), H(Conf_{t,w}))}$$

Level of homogeneity of the whole system S:

$Ho(S) = Avg(t \mapsto Ho_t)$

Ho(S) ¼ 1 means, that the system is homogeneous

Ho(S) ¼ 0 means, that the system not homogeneous at all

Global state awareness

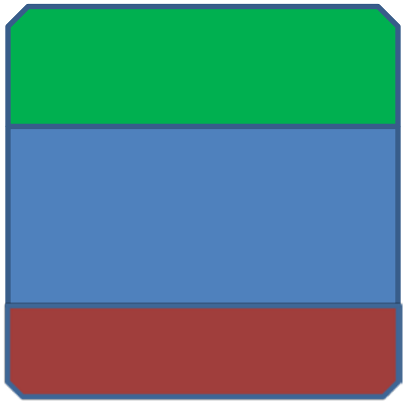
To measure the **level of global state awareness**

ω 2 [0, 1]

the initial states are partitioned according to the equivalence relation induced by a property of interest.

- Measurement of the information about the initial equivalence class inside of each node:

$$\omega_t = 1 - \frac{H(\text{equivalence class} \mid \text{local history})}{H(\text{equivalence class})}$$



Level of global state awareness of the whole system S:

$\omega(S) = \text{Avg}(t \mapsto \omega_t)$

ω ¼ 1 means high level of global state awareness

(each node knows much about initial equivalence class)

ω ¼ 0 means low level of global state awareness

(each node knows few about initial equivalence class)

Discussion

- Problem: **Scalability**
  - In large systems, it might be difficult to compute the quantitative measures.
  - **Simulation** results may be used to get approximations.
  - For entropy based measures, the probabilities can be approximated by the **relative frequencies** of the events in the simulation runs.
  - Mean values of random variables can be approximated by the **arithmetic mean** values of the outcome of the variables in the simulation runs.

Discussion

- How can the quantitative measures be used to provide a framework to study how global phenomena emerge in complex self-organizing systems from local interactions?
- How can the quantitative measures help to improve the design of new complex systems?

Example:
Slot synchronizing in wireless networks

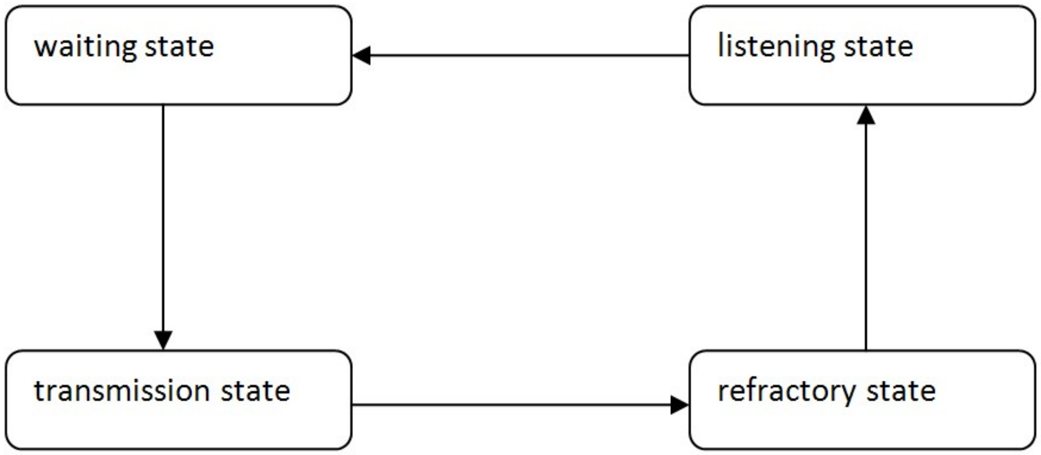
Wireless network:

- For communication, time is divided into slots.
- There is no central clock, which defines when a slot begins.
- The nodes try to synchronize the slots.

Example: Slot synchronizing in wireless networks

Slot synchronization algorithm of Tyrrell, Auer and Bettstetter:

- At each point of time, each node is in one of four different states:
- In the transmission state, the node transmits a pulse to it's neighbors to indicate the beginning of a slot.
- In the listening state, the node can receive and decode pulses from it's neighbors and it adjusts its phase function ϕ according to these pulses. The listening state ends, when the threshold $\phi_{max} = 1$ is reached.
- In the waiting state and in the refractory state, the node does nothing.
- The length of an uncoupled cycle is 2T with T>0.



Example:
Slot synchronizing in wireless networks

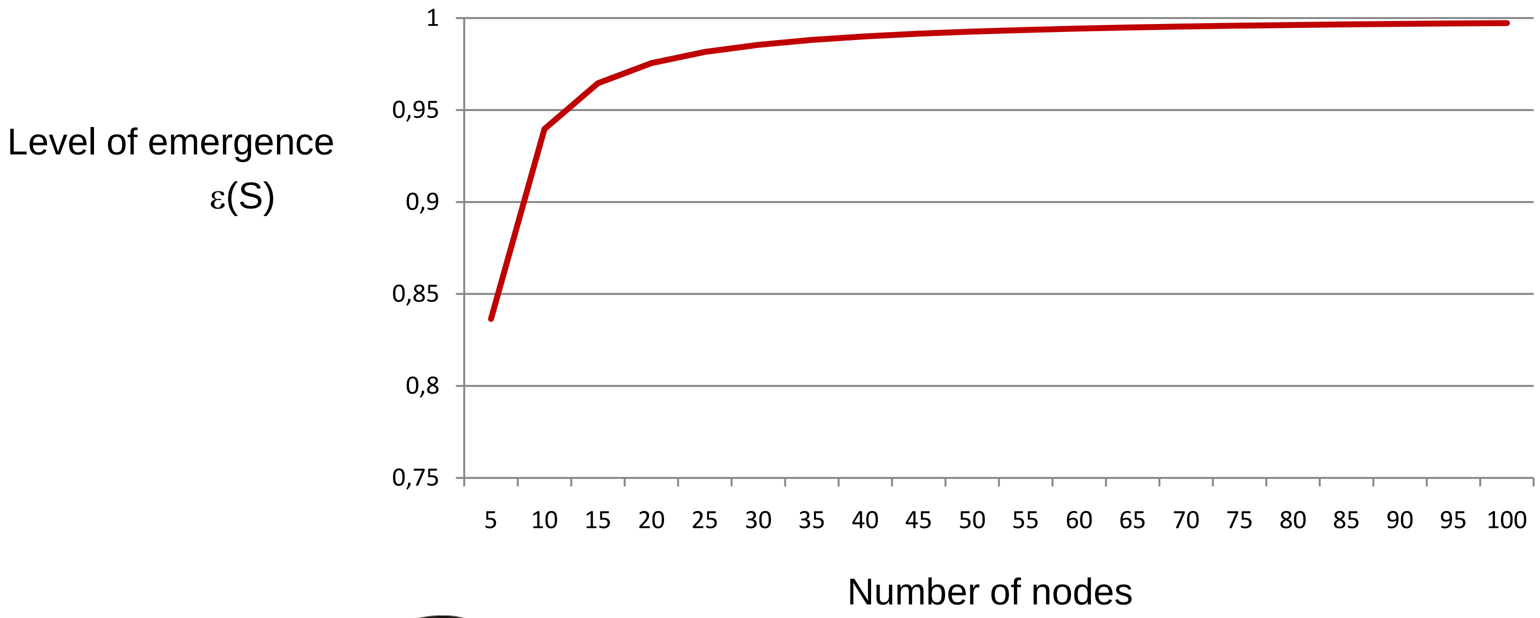
Simulation results show:

- Two groups of synchronizations are built.
- Inside each group we have good synchronization: Each object of the group fires a pulse at nearly the same time like the other objects of the group.
- The second group fires T time units after the first group.
- By using slots of length T we get a good slot synchronization

Example:
Slot synchronizing in wireless networks

The system is autonomous since it does not contain external node: $\alpha = 1$

The synchronization of the objects is emergence, since this is a global property of the system, which is induced by the local interactions.



Target orientation

Calculation of the level of **target orientation**:

The goal is to minimize the time differences between the beginning of the slots of the nodes.

$dist_c(v,w)$: Slot distance of nodes v, w 2 V in configuration c

$$b(c) = 1 - \frac{\sum_{v,w \in V} dist_c(v,w)}{|V^2| \cdot T/2}$$

After fixing the system parameters, we can calculate the level of target orientation.

$|V| = 30$, $T = 100$, $T_{dec} = 15$, $T_{Tx} = 45$, $T_{refr} = 35$, $T_{wait} = 40$, $\alpha = 1.2$, $\beta = 0.01$

Target orientation: TO(S) ¼ 0.997

Resilience

For the level of resilience, we first have to fix the behavior of the malfunctioned nodes.

Resilience with respect to the break down of a node:

If the graph is still strongly connected after the break down of a node, then the remaining nodes are able to synchronize: Res(S) ¼ TO(S) ¼ 1

Resilience with respect to an intruder at a node v_0 2 V periodically sending pulses to its neighbors.

$|V| = 30$, $T = 100$, $T_{dec} = 15$, $T_{Tx} = 45$, $T_{refr} = 35$, $T_{wait} = 40$, $\alpha = 1.2$, $\beta = 0.01$

Choose θ as the time between two pulses of the intruder.

θ	45	70	100	120	150
Res(S)	0.987	0.985	0.996	0.991	0.996

Adaptivity

Level of adaptivity:

There are no external nodes in the system.

$Ad_t = E(b_\theta(Conf_t)) = TO_t$

$Ad(S) = TO(S)$ ¼ 0.997

Homogeneity

Level of homogeneity:

For a regular graph, the system is homogeneous:

$$Ho_t = 1 - \frac{\sum\limits_{v,w \in V, v<w} |H(Conf_{t,v}) - H(Conf_{t,w})|}{\sum\limits_{v,w \in V, v<w} \max(H(Conf_{t,v}), H(Conf_{t,w}))} = 1$$

Ho(S) = 1

If the degree of the nodes differ, the system is not fully homogeneous, but it reaches a high level of homogeneity after building the synchronization groups: Ho(S) ¼ 1

Example: Evacuation scenario

- Scenario: Evacuation in a building
  - Each person wears a life belt:
    - Ambient Intelligence (AmI) device, which is able to communicate with other life belts to improve the evacuation.

Example: Evacuation scenario

- Topology
  - Each node in the graph represents one person wearing a life belt.
  - Each edge in the graph represents a communication channel
  - Since the persons move around, the topology changes during time
    -) Graph $G_t = (V, E_t)$ depends on time t
  - In a simple scenario, external nodes are not needed. The system will self-organize.
    - In a more complex scenario, external nodes can be introduced to model changes in the environment (e.g. break down of a part of the building, etc.)

Example: Evacuation scenario

- Behaviour
  - The behaviour of each node is represented by an automaton.
  - Internal state contains
    - Current position
    - Some relevant information about exits received by other nodes in the past.
  - Local rules for state change and communication with neighbours:
    - **Problem:** How can we find **good** rules to maximize the evacuation?
    - **Idea:** Define different rule sets (or a fixed rule set containing some global rule parameters) and use quantitative measures for comparison.

Example: Evacuation scenario

UNIVERSITÄT PASSAU
Fakultät für Informatik und Mathematik

For the target orientation we need a valuation of configurations:

b : Conf ! [0, 1]

- In the evacuation scenario the good configurations are those where many people have already escaped:
  - b(c) = #escaped/N   N = number of persons

- Consider a run of the system starting at time t = 0 ending at t = T.
- $TO_t = \boldsymbol{E}(b(Conf_t))$ is a nondecreasing function
- Goal: Try to maximize TO_T
- Different rule sets can be compared.
- Rules with different parameters can be compared.

Example: Evacuation scenario

- Next to the target goal orientation also other measures might be useful to **compare different strategies** and/or **different rule parameters**.
- Example: **Level of global state awareness**

Example: Evacuation scenario

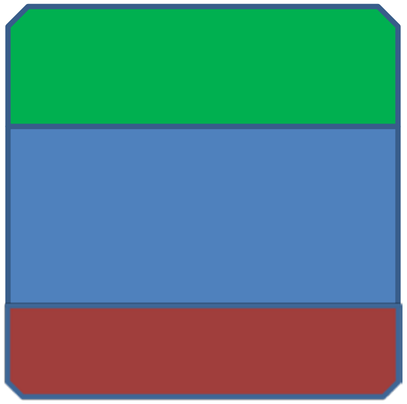
To measure the **level of global state awareness**

$\omega \in [0, 1]$

The initial states are partitioned according to the equivalence relation induced by a property of interest.

- Measurement of the information about the initial equivalence class inside of each node:

$$\omega_t = 1 - \frac{H(\text{equivalence class} \mid \text{local history})}{H(\text{equivalence class})}$$



- Equivalence class 1:
  All configurations where exit 1 has highest throughput
  - high capacity, only small crowd near the exit
- Equivalence class 2:
  All configurations where exit 2 has highest throughput
- Equivalence class 3:
  All configurations where exit 3 has highest throughput

Example: Evacuation scenario

UNIVERSITÄT PASSAU
Fakultät für Informatik und Mathematik

- **Idea:** Each person tries to learn in which equivalence class the system is and uses the information about his own position to decide which exit fits best for him.
- The more information the people get about the equivalence class the better they can decide about the best exit.
- Communication between the nodes is used for **learning**:
  - "I found an exit at position (x, y)"
  - "At my current position (x, y) are k other persons in the neighbourhood."
  - …

Example: Evacuation scenario

- Calculate the measure of **global state awareness** $\omega(S)$ for different strategies.
- The strategy with highest global state awareness yields the **maximal information** such that each person has good chances that his decision is really best.
- **Problem**:
  - Before the level of global state awareness can be applied, a point of time T for the **initialization** must be fixed.
  - The equivalence classes always belong to this point of time T.
  - Starting the system at time T, the people learn in each step something about the equivalence class at time T.
  - The best exit at time t > T might differ from the best exit at time T.
  - But this method is still useful since the information spread is much faster than the movement of the persons.

Conclusion

The mathematical modeling can be used for a wide variety of systems:

- Technical systems
- Biological systems
- Physical systems

and many more.

The models can help to analyse the behavior of complex systems.

Quantitative measures provide a link from the micro level to the macro level:

- They describe **global properties of the system**
- They can be used for the **analysis** of real world systems.
- They can be used for **design**, **engineering** and **optimization** of new systems.

UNIVERSITÄT PASSAU
Fakultät für Informatik und Mathematik

Thank you for your attention

Hermann de Meer

Chair for Computer Networks & Computer Communications
University of Passau
Germany

demeer@fim.uni-passau.de

References

- R. Holzer and H. de Meer. ***Quantitative Modeling of Self-Organizing Properties***. In Proc. of 4th International Workshop on Self-Organizing Systems (IWSOS 2009). Zurich, Switzerland, December 9-11, 2009. Springer, LNCS.
- R. Holzer and P. Wüchner and H. de Meer. **Modeling of Self-Organizing Systems: An Overview**. In Workshop über Selbstorganisierende, adaptive, kontextsensitive, verteilte Systeme (SAKS 2010), Electronic Communications of the EASST, Vol. 27, 2010
- Christopher Auer, Patrick Wüchner, and Hermann de Meer. ***The Degree of Global-State Awareness in Self-Organizing Systems***. In Proc. of 4th International Workshop on Self-Organizing Systems (IWSOS 2009). Zurich, Switzerland, December 9-11, 2009. Springer, LNCS.
- Richard Holzer and Hermann de Meer and Christian Bettstetter. **On autonomy and emergence in self-organizing systems**. IWSOS 2008 - 3rd International Workshop on Self-Organizing Systems, Vienna, Austria, December 10-12, 2008, Springer Verlag 2008
- Christopher Auer, Patrick Wüchner, Hermann de Meer. **Target-Oriented Self-Structuring in Classifying Cellular Automata.** In Proc. of the 15th Int'l Workshop on Cellular Automata and Discrete Complex Systems (Automata 2009). São José dos Campos, Brazil, October 10-12, 2009. Extended version submitted to the Journal of Cellular Automata (JCA).
- Christopher Auer, Patrick Wüchner, and Hermann de Meer. ***A Method to Derive Local Interaction Strategies for Improving Cooperation in Self-Organizing Systems.*** In Proc. of 3rd Int'l Workshop on Self-Organizing Systems (IWSOS 2008). Vienna, Austria, December 10-12, 2008. Springer, LNCS 5343, pp. 170-181. *(received best-paper award)*
- Kamil Kloch, Jan W. Kantelhardt, Paul Lukowicz, Patrick Wüchner, Hermann de Meer. **Ad-hoc information spread between mobile devices: A case study in analytical modeling of controlled self-organization in IT systems.** 23rd Int'l Conf. on Architecture of Computing Systems (ARCS 2010), Leibniz Universität Hannover, Germany, February 22-25, 2010.