Quantitative Modeling of Self-Organizing Properties

Richard Holzer and Hermann de Meer

Faculty of Informatics and Mathematics, University of Passau, Innstrasse 43, 94032 Passau, Germany
{holzer,demeer}@fim.uni-passau.de,

Abstract. For analyzing properties of complex systems, a mathematical model for these systems is useful. In this paper we give quantitative definitions of adaptivity, target orientation, homogeneity and resilience with respect to faulty nodes or attacks by intruders. The modeling of the system is done by using a multigraph to describe the connections between objects and stochastic automatons for the behavior of the objects. The quantitative definitions of the properties can help for the analysis of existing systems and for the design of new systems. To show the practical usability of the concepts, the definitions are applied to a slot synchronization algorithm in wireless sensor networks.

Key words: Self-Organziation, Mathematical modeling, Systems, Adaptivity, Target orientation, Homogeneity, Resilience

1   Introduction

A goal for networking systems is to reduce administrative requirements for users and operators. A technical system should be able to manage itself as much as possible without requiring human effort. Failures and malfunctioned modules should be detected automatically and auto-corrected if possible. This leads to the concept of self-organization, which is a topic that have become more and more important in the last few years. Much research has been done for the design and analysis of autonomous and self-organizing systems.

To assist these technological advances, a mathematical model is useful for a better understanding of complex systems and to improve the design of new systems. The main goal of this paper is to contribute to this issue. We give formal definitions of adaptivity, target orientation, homogeneity and resilience. We also apply these concepts to a practical example, namely the self-organized slot-synchronization in wireless networks.

In this paper, Section 2 gives an overview of the related work and Section 3 gives a mathematical model for complex systems. Sections 4-7 define quantitatively the concepts adaptivity, target orientation, homogeneity and resilience.

In Section 8, we apply our definitions to model slot synchronization in wireless networks. Section 9 contains some discussions, applications and advanced conclusions of the formal model described in this paper.

2   Related work

In the last few years, much research has been done in the topic of self-organizing systems. Unfortunately, there is no generally accepted meaning of self-organization. According to [1] and [2] some typical features of self-organization are autonomy, emergence, self-maintenance, adaptivity, decentralization, and optimization.

A non-technical overview of self-organization can be found in [2]. Other definitions and properties of self-organizing systems can be found in thermodynamics [3], information theory [4], cybernetics [5], [6], [7] and synergetics [8]. An extensive description about the design of self-organizing systems is in [9]. A good overview about modeling complex systems can be found in [10]. [11] gives a survey about practical applications of self-organization. For modeling continuous self-organizing systems and a comparison between discrete and continuous modeling see [12]. Quantitative definitions of autonomy, emergence and global-state awareness, which use the information theoretical concept of entropy, can be found in [13], [14] and [15]. The methods of [13] for modeling a system with stochastic automatons are also used in Section 3 of this paper. While [13] gives quantitative definitions only for the concepts autonomy and emergence, this paper contains the definitions of adaptivity, target orientation, homogeneity and resilience.

3   Discrete Systems

For modeling discrete systems, we use the methods of [13], which are based on the ideas of [2]: A multigraph describes the connections between objects and stochastic automatons describe the behavior of the objects. These concepts allow the modeling of a wide variety of complex systems of the real world, e.g. systems that appear in biology, physics, computer science or any other field.

In the real world, not all properties are known in all detail (e.g. it would be very difficult to describe a deterministic behavior of an animal), but there are many things, that can better be described by probabilities. Therefore, a stochastic behavior is more adequate than a deterministic one. In this section, we use directed multigraphs to describe the communication channels for the interaction between objects: Each node in the multigraph corresponds to an object and each edge of the multigraph is used to model the interaction (e.g. transfer of data) between the objects. For modeling the external influence of the environment we use special vertices (external nodes) in the multigraph, where the edges from these vertices represent the channels for the input into the system, and the edges to these vertices represent the output of the system. We distinguish between user data (data from the environment that is processed by the system)

and control data (data from the environment to change the behavior of the system). The behavior of the objects is modeled by finite stochastic automatons.

Definition 1 *A* discrete system[1] $\mathcal{S} = (G, E, C, A, a)$ *consists of*

- *a finite directed multigraph $G = (V, K, \tau)$, where V is the set of vertices, K is the set of directed edges (loops are also allowed), and $\tau : K \to V^2$ assigns to each edge $k \in K$ the corresponding vertices $\tau(k)$, where the starting vertex is also denoted by $k-$ and the ending vertex is denoted by $k+$. Therefore we have $\tau(k) = (k-, k+)$ for each $k \in K$. For a vertex $v \in V$ the set of edges ending in v is denoted by $v- := \{k \in K \mid k+ = v\}$ and the set of edges starting at v is denoted by $v+ := \{k \in K \mid k- = v\}$. Analogously for $T \subseteq V$ the sets $T-$ and $T+$ are defined by $T- := \bigcup_{v \in T} v- = \{k \in K \mid k+ \in T\}$ and $T+ := \bigcup_{v \in T} v+ = \{k \in K \mid k- \in T\}$.*
- *a subset of vertices $E \subseteq V$, which elements are called* external nodes*. The other vertices are called* internal nodes*. The* input edges *are denoted by $I = E+ = \{k \in K \mid k- \in E\}$. The* output edges *are denoted by $O = E- = \{k \in K \mid k+ \in E\}$. All other edges are called* internal edges*.*
- *a subset of the input edges $C \subseteq I$. The elements of C are called* control edges*. The other input edges $U := I \setminus C$ are called* user edges*.*
- *a finite set A, which is used as alphabet for communication between the nodes.*
- *a family $a = (a_v)_{v \in V}$ of stochastic automatons $a_v = (A^{v-}, A^{v+}, S_v, P_v, d_v)$, where*
  - *$A^{v-} = \{(x_k)_{k \in v-} \mid x_k \in A, k \in v-\}$ are the* local input values*,*
  - *$A^{v+} = \{(x_k)_{k \in v+} \mid x_k \in A, k \in v+\}$ are the* local output values*,*
  - *S_v is the set of states,*
  - *$P_v : S_v \times A^{v-} \times S_v \times A^{v+} \to [0,1]$ is a function, such that $P(q, x, \cdot, \cdot) : S_v \times A^{v+} \to [0,1]$ is a probability distribution on $S_v \times A^{v+}$ for each $q \in S_v$ and $x \in A^{v-}$. The value $P(q, x, q', y)$ is the probability, that the automaton moves from state $q \in S_v$ into the new state $q' \in S_v$ and gives the local output $y \in A^{v+}$ when it receives the local input $x \in A^{v-}$.*
  - *$d_v : S_v \to \mathbb{R}^+$ is a map, where $d_v(q)$ describes the delay between two pulses of the clock when the automaton is in state $q \in S_v$ (there is no global clock, but each automaton has its own clock).*

These definitions allow us to model complex systems of the real world: Assume that we would like to analyze a system, e.g. a computer network. Then each node of the network corresponds to a vertex of the multigraph. If one node of the network is able to communicate with another node, then we draw an edge between the vertices in the graph. The behavior of each node is modeled by a stochastic automaton, which describes, how the internal state changes for each input, which it gets from the other nodes. Note that the clock pulses of the automatons are deterministic: For each node $v \in V$ and each state $q \in S_v$ the value $d_v(q)$ is the amount of time, the automaton stays in the state q. We could also

---

[1] see also [13]

use nondeterminism for the clock pulses (e.g. with a probability distribution on a discrete set of values for the duration), but this is usually not needed, because this can be modeled by using the nondeterminism for the state transition into different states $q \in S_v$ with different values $d_v(q)$.

If we consider the global view on the system at a point of time, then we see a current local state inside each automaton and a current value on each edge, which is transmitted from one node to another node. Such a global view is a snapshot of the system. It is formally defined in the following definition:

Definition 2 *Let $\mathcal{S}$ be a system. A* configuration $c = (c_V, c_K, D)$ *consists of*

- *a tuple $c_V \in \prod_{v \in V} S_v$ of states, which defines the current states of the automatons,*
- *a map $c_K : K \to A$, which defines the current symbols on the edges,*
- *a map $D : V \to \mathbb{R}^+$, where $D(v) \leq d_v(c_V(v))$ describes the length of the time interval between the current point of time and the next pulse of the clock in the automaton a_v.*

The set of all configuration is denoted as Conf*. For a configuration $c = (c_V, c_K, D)$ and a set $T \subseteq K$ of edges the assignment $c_K|_T : T \to A$ of the edges in T is also denoted by $c|_T$. The restriction of c to the external nodes is defined by $c|_{ext} = (c_V|_E, c_K|_{E+}, D|_E)$. The restriction of c to the internal nodes is defined by $c|_{int} = (c_V|_{V \backslash E}, c_K|_{(V \backslash E)+}, D|_{V \backslash E})$. An* initialization *of $\mathcal{S}$ is a pair (Γ, P_Γ), where Γ is a set of configurations and $P_\Gamma : \Gamma \to [0,1]$ is a probability distribution on Γ, which describes, with which probability the system starts in a certain configuration $c \in \Gamma$. For a configuration $c = (c_V, c_K, D)$ the* duration of c *is defined by $d_c = \min\{D(v) \mid v \in V\}$. Let $N_c = \{v \in V \mid D(v) = d_c\}$. Then the elements of N_c are the nodes with the soonest clock pulse after the current point of time. A configuration $c' = (c'_V, c'_K, D')$ is a* successor configuration *of $c = (c_V, c_K, D)$ with probability $p \in [0,1]$ (notation: $P(c \to c') = p$) if*

- $c'_V(v) = c_V(v)$ *for $v \in V \backslash N_c$,*
- $c'_K(k) = c_K(k)$ *for $k \in (V \backslash N_c)+$,*
- $D'(v) = D(v) - d_c$ *for $v \in V \backslash N_c$,*
- $D'(v) = d_v(c'_V(v))$ *for $v \in N_c$,*
- $p = \prod_{v \in N_c} P_v(c_V(v), (c_K(k))_{k \in v-}, c'_V(v), (c'_K(k))_{k \in v+})$.

A (finite or infinite) tuple $s = (c_0, c_1, c_2, \ldots)$ of configurations is called configuration sequence *if for each $j \geq 0$ we have $P(c_j \to c_{j+1}) > 0$. For a configuration c let $succ(c)$ be a random variable with the probability distribution $P(succ(c) = c') = P(c \to c')$ for each successor configuration c' of c. This concept of successor can be extended in a canonical way to arbitrary sequences $(c_0, c_1, c_2, \ldots, c_j)$ of configurations to get the probability $P(c \to^* c')$, that c' is reached from the configuration c, where the steps are considered as independent. The duration of a sequence $s = (c_0, c_1, c_2, \ldots, c_j)$ of configurations $c_i \in$ Conf is $d_s = \sum_{i=0}^{j} d_{c_i}$. For*

a given duration $t \geq 0$ let $P(c \rightarrow^t c')$ be the probability, that c' is active t time units after the time of c, i.e. we consider the probabilities of the sequences $s = (c_0, c_1, c_2, \ldots, c_j)$ of configurations with $c_0 = c$ and $c_j = c'$ with $d_s - d_{c'} \leq t < d_s$. For $t \geq 0$ let $P(\Gamma \rightarrow^t c)$ be the probability, that c is active at time t. Define

$$\Gamma_t = \{c \mid P(\Gamma \rightarrow^t c) > 0\},$$

i.e. Γ_t is the set of all configurations c that may be active at time t, where we assume that the initialization of the system is at time $t_0 = 0$. Let Conf_t be the random variable taking values in Γ_t with the probability distribution $P(\mathrm{Conf}_t = c) = P(\Gamma \rightarrow^t c)$ for $c \in \Gamma_t$.

To analyze the behavior of a system, we initialize it at time $t_0 = 0$ by choosing a start configuration $c_0 \in \Gamma$ and then the automatons produce a sequence $c_0 \rightarrow c_1 \rightarrow c_2 \rightarrow \ldots$ of configurations during the run of the system. When we do a snapshot of the system at time $t \geq 0$, we see a current configuration $c \in \Gamma_t$. Since the automaton and the initialization are not deterministic, the sequence $c_0 \rightarrow c_1 \rightarrow c_2 \rightarrow \ldots$ is not uniquely determined by the system, but it depends on random events. So for each time $t \geq 0$, we have a random variable Conf_t, which describes, with which probability $P(\mathrm{Conf}_t = c)$ the system is in a given configuration c at time t.

A single node v in the system has not the global view, it only sees its local input and output values and its internal state. This concept is given in the following definition.

Definition 3 *Let $\mathcal{S}$ be a system, $v \in V$ and $c = (c_V, c_K, D)$ be a configuration. The* local configuration *of c in v is defined by*

$$Loc_v(c) = (c_V(v), (c_K(k))_{k \in v - \cup v+}, D(v))$$

Let $Conf_{t,v} = Loc_v(Conf_t)$ be the random variable for the local configuration of v at time $t \in \mathbb{R}_0^+$.

For measuring the information in a system we use the statistical entropy:

Definition 4 *For a discrete random variable X taking values from a set W the entropy $H(X)$ of X is defined by [16]*

$$H(X) = - \sum_{w \in W} P(X = w) \log_2 P(X = w).$$

For discrete random variables X, Y the conditional entropy *$H(X|Y)$ is defined by*

$$
\begin{aligned}
H(X|Y) = &H(X,Y) - H(Y) \\
= &- \sum_{w,w' \in W} P(X = w, Y = w') \log_2 P(X = w, Y = w') \\
&+ \sum_{w \in W} P(Y = w) \log_2 P(Y = w)
\end{aligned}
$$

The entropy measures, how many bits are needed to encode the outcome of the random variable in an optimal way. In Section 6 we use this concept to define quantitatively the homogeneity of a system. Another concept, that we need for the quantitative definitions in Sections 4-7, is the average value of a function:

Definition 5 *Let $f : \mathbb{R}_0^+ \to \mathbb{R}$ be a real function, which is integrable on every finite interval. For points of time $s > r \geq 0$ the* average value of f in the interval $[r, s]$ *is defined by* $\text{Avg}_{[r,s]}(f) = \frac{1}{s-r} \int_r^s f(t)dt$. *The* average value of f *is defined by* $\text{Avg}(f) = \liminf_{t \to \infty} \text{Avg}_{[0,t]}(f)$.

4   Target orientation

Before a new system is designed, we have the goal of the system in our mind: The system should fulfill a given purpose. The behavior of each node is defined in such a way, that this goal is reached, so the design of a system needs a *target orientation*, which is specified in the following definition.

Definition 6 *Let $\mathcal{S}$ be a system and (Γ, P_Γ) be an initialization. Let $b : \text{Conf} \to [0, 1]$ be a valuation map for the configurations. For a point of time $t \geq 0$ the* level of target orientation *of $\mathcal{S}$ at time t is defined by* $\text{TO}_t(\mathcal{S}, \Gamma) = E(b(\text{Conf}_t))$, *where E is the mean value of the random variable. The* level of target orientation *of the system $\mathcal{S}$ is defined by* $\text{TO}(\mathcal{S}, \Gamma) = \text{Avg}(t \mapsto \text{TO}_t(\mathcal{S}, \Gamma))$. *The system $\mathcal{S}$ is called target oriented with respect to b if* $\text{TO}(\mathcal{S}, \Gamma) = 1$.

For the target orientation, the valuation map b describes which configurations are "good": A high value $b(c) \approx 1$ means that the configuration c is a part of our goal which we had in mind during the design of the system. The level of target orientation measures the valuations $b(c)$ of the configurations during the whole run of a system: A high level of target orientation ($\text{TO}(\mathcal{S}, \Gamma) \approx 1$) means, that the mean valuation of the configurations during a run of the system often is nearly 1.

In [17], a practical application of target orientation is given in the theory of cellular automatons. In this example, the target goal is the classification of the initial state. This is the reason, why [17] uses a different definition of target orientation: A system is target oriented, if the nodes are able to classify the initial state with respect to a given equivalence relation, which describes the classes of interest.

5   Resilience

For computer networks, there are different forms of resilience:

- resilience with respect to malfunctioned nodes
- resilience with respect to attacks by an intruder, which is inside the network

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

The following definition can be used to measure these different forms of resilience:

Definition 7 *Let $\mathcal{S}$ be a system and (Γ, P_Γ) be an initialization. Let Θ be a set and $p_\Theta : \Theta \to [0,1]$ be a probability distribution. Let $Z = (Z_{\theta,v})_{\theta \in \Theta, v \in V}$ be a family of stochastic automatons*

$$Z_{\theta,v} = (A^{v-}, A^{v+}, ZS_{\theta,v}, ZP_{\theta,v}, Zd_{\theta,v})$$

For $\theta \in \Theta$ let $\mathcal{S}^\theta$ be the system $\mathcal{S}$ after replacing a_v by $Z_{\theta,v}$ for all $v \in V$. Let $(\Gamma^{\mathcal{S}^\theta}, P_{\Gamma^{\mathcal{S}^\theta}})$ be an initialization of $\mathcal{S}^\theta$. Let Conf^θ be the set of the configurations of $\mathcal{S}^\theta$. Let $b = (b_\theta)_{\theta \in \Theta}$ be a family of valuation maps $b_\theta : \mathrm{Conf}^\theta \to [0,1]$ for the configurations. For a point of time $t \geq 0$ let Conf_t^Θ be the random variable, which applies the random variable Conf_t in the system $\mathcal{S}^\theta$ after choosing $\theta \in \Theta$ randomly according to the probability p_Θ. The level of resilience of $\mathcal{S}$ at time t is defined by $\mathrm{Res}_t(\mathcal{S}, \Gamma) = E(b(\mathrm{Conf}_t^\Theta))$, where E is the mean value of the random variable. The level of resilience of the system $\mathcal{S}$ is defined by $\mathrm{Res}(\mathcal{S}, \Gamma) = \mathrm{Avg}(t \mapsto \mathrm{Res}_t(\mathcal{S}, \Gamma))$. The system $\mathcal{S}$ is called resilient with respect to b if $\mathrm{Res}(\mathcal{S}, \Gamma) = 1$.

In this definition the automaton $Z_{\theta,v}$ can be used to describe the malfunctioned behavior of a node v. In a computer network, this behavior could be caused by hardware failure, it could be the behavior of an intruder inside the network ($v \in V \setminus E$) or outside of the network ($v \in E$) or it does not send data to its successor nodes due to a breakdown. The system is resilient if despite the malfunctioned nodes the system still runs through many "good" configurations.

If there are only few malfunctional nodes, then we can use $Z_v = a_v$ for the other nodes. If the behavior of a malfunctional node v depends on the original behavior a_v, then the automaton Z_v can be a modification of the original automaton a_v to describe the malfunctional behavior of v.

6 Homogeneity

The following definition describes how homogenous a system is.

Definition 8 *Let $\mathcal{S}$ be a system and (Γ, P_Γ) be an initialization. Let $<$ be a strict linear order on V. For a point of time $t \geq 0$ the level of homogeneity of $\mathcal{S}$ at time t is defined by $\mathrm{Ho}_t(\mathcal{S}, \Gamma) = 1 - \dfrac{\sum\limits_{v,w \in V, v<w} |H(\mathrm{Conf}_{t,v}) - H(\mathrm{Conf}_{t,w})|}{\sum\limits_{v,w \in V, v<w} \max(H(\mathrm{Conf}_{t,v}), H(\mathrm{Conf}_{t,w}))}$. The level of homogeneity of the system $\mathcal{S}$ is defined by $\mathrm{Ho}(\mathcal{S}, \Gamma) = \mathrm{Avg}(t \mapsto \mathrm{Ho}_t(\mathcal{S}, \Gamma))$. The system $\mathcal{S}$ is called homogeneous, if $\mathrm{Ho}(\mathcal{S}, \Gamma) = 1$.*

The level of homogeneity measures the similarity of the local configurations of different nodes: A high level of homogeneity means, that the local configurations have a similar entropy, while a low level indicates, that the entropies of the local configurations in different nodes differ very much.

7   Adaptivity

The following definition measures the adaptivity of a system.

Definition 9 *Let $\mathcal{S}$ be a system and (Γ, P_Γ) be an initialization. Let (Θ, p_Θ) be a set with a probability distribution $p_\Theta : \Theta \to [0,1]$. Let $Z = (Z_{\theta,v})_{\theta \in \Theta, v \in C-}$ be a family of stochastic automatons*

$$Z_{\theta,v} = (A^{v-}, A^{v+}, ZS_{\theta,v}, ZP_{\theta,v}, Zd_{\theta,v})$$

For $\theta \in \Theta$ let $\mathcal{S}^\theta$ be the system $\mathcal{S}$ after replacing a_v by $Z_{\theta,v}$ for all $v \in C-$. Let $\Gamma^{\mathcal{S}^\theta}$ be an initialization of $\mathcal{S}^\theta$. Let Conf^θ be the set of the configurations of $\mathcal{S}^\theta$. Let $b = (b_\theta)_{\theta \in \Theta}$ where $b_\theta : \mathrm{Conf}^\theta|_{int} \to [0,1]$ is a valuation map for the configurations of the internal nodes. For a point of time $t \geq 0$ let Conf_t^Θ be the random variable, which applies the random variable Conf_t in the system $\mathcal{S}^\theta$ after choosing $\theta \in \Theta$ randomly according to the probability p_Θ. The level of adaptivity of $\mathcal{S}$ at time t is defined by $\mathrm{Ad}_t(\mathcal{S}, \Gamma) = E(b(\mathrm{Conf}_t^\Theta|_{int}))$, where E is the mean value of the random variable. The level of adaptivity of the system $\mathcal{S}$ is defined by $\mathrm{Ad}(\mathcal{S}, \Gamma) = \mathrm{Avg}(t \mapsto \mathrm{Ad}_t(\mathcal{S}, \Gamma))$. The system $\mathcal{S}$ is called adaptive with respect to b if $\mathrm{Ad}(\mathcal{S}, \Gamma) = 1$.

The level of adaptivity measures the influence of the change of control data: A high value of $\mathrm{Ad}(\mathcal{S}, \Gamma)$ means that the mean valuation of the configurations during each run of the system with the new control data is nearly 1, so many "good" configurations are reached.

If the system has no external nodes ($E = \emptyset$), then no automaton in replaced. In this case, the concept of target orientation can be seen as a special case of the concept of adaptivity: By choosing a one element set $\Theta = \{\theta\}$ we get $\mathrm{TO}(\mathcal{S}, \Gamma) = \mathrm{Ad}(\mathcal{S}, \Gamma)$. For $E = \emptyset$ with $|\Theta| > 1$ the level $\mathrm{Ad}(\mathcal{S}, \Gamma)$ is the weighted mean level of target orientation $\mathrm{TO}(\mathcal{S}, \Gamma)$ with respect to $\theta \in \Theta$:

$$\mathrm{Ad}(\mathcal{S}, \Gamma) = \sum_{\theta \in \Theta} p_\Theta(\theta) \cdot \mathrm{TO}_\theta(\mathcal{S}, \Gamma)$$

where $\mathrm{TO}_\theta(\mathcal{S}, \Gamma)$ is the level of target orientation with respect to b_θ.

8   Slot synchronization in wireless networks

In this section we apply the definitions of the previous sections to the process of self-organized slot-synchronization in wireless networks [18]. We consider a set of nodes, each node being able to communicate with some of the other nodes. The

access on the shared medium is organized in time slots. Since there is no central clock, which defines when a slot begins, the nodes perform a slot synchronization in a completely distributed manner. An algorithm for this purpose is proposed by Tyrrell, Auer, and Bettstetter in [18]. It is based on the model of pulse-coupled oscillators by Mirollo and Strogatz [19].

In the latter synchronization model, the clock is described by a phase function ϕ which starts at time instant 0 and increases over time until it reaches a threshold value $\phi_{th} = 1$. The node then sends a "firing pulse" to its neighbors for synchronization. Each time a node receives such a pulse from a neighbor, it adjusts its own phase function by adding $\Delta\phi := (\alpha - 1)\phi + \beta$ to ϕ, where $\alpha > 1$ and $\beta > 0$ are constants.

In [18] the pulse-coupled oscillator synchronization model is adapted to wireless systems, where also delays (e.g., transmission delay, decoding delay) are considered. The duration of an uncoupled period is now $2T$ with $T > 0$. This period is divided into four states (see Figure 1). Let $\gamma \in [0, 2T]$ be a time instant. Then the node is in a

- waiting state, if $\gamma \in [0, T_{wait}) =: I_{wait}$,
- transmission state, if $\gamma \in [T_{wait}, T_{wait} + T_{Tx}) =: I_{Tx}$,
- refractory state, if $\gamma \in [T_{wait} + T_{Tx}, T_{wait} + T_{Tx} + T_{refr}) =: I_{refr}$,
- listening state, if $\gamma \in [T_{wait} + T_{Tx} + T_{refr}, 2T) =: I_{Rx}$,

where the constants are defined as follows:

- T_{Tx}: Delay for transmitting a value,
- T_{wait}: Waiting delay after the phase function reached the threshold. The transmission of the firing pulse begins after this delay. The waiting delay is calculated by $T_{wait} = T - (T_{Tx} + T_{dec})$, where T_{dec} is the delay for decoding the received value.
- T_{refr}: Refractory delay after the transmission of the firing pulse to avoid an unstable behavior.
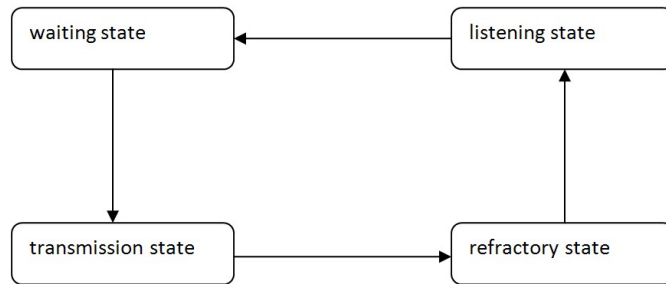

Figure 1: State diagram

Let $T_{Rx} = 2T - T_{wait} - T_{Tx} - T_{refr}$ be the duration of an uncoupled listening state. We assume that each of these durations $T_{wait}, T_{Tx}, T_{refr}, T_{Rx}$ is less than T. The listening state is the only state, in which firing pulses from the neighbors

can be received and decoded, and the phase function is changed only during the listening state.

This network can be modeled as a discrete system $\mathcal{S} = (G, E, C, A, a)$, where the internal state of an automaton consists of the current value of the phase function $\phi \in [0, 1]$, the current position $\gamma \in [0, 2T]$ in the cycle in Figure 1 and some information about the decoding delays. In [13] the complete system and the algorithm is modeled in full details, so we omit the details here. Simulation results in [18] show that during the run of the system, groups of synchronizations are built, i.e. inside each group we have good synchronization (each node of the group fires at nearly the same time like the other nodes of the group), and if we wait long enough, then there are only two groups left firing T time units apart from each other. If we use T as the size of the slots, where the beginning of each slot is at $\gamma = 0$ or $\gamma = T$, then both groups have the same slot allocation, so the slots of all nodes are synchronized.

We now apply our definitions of the previous sections.

Concerning the level of target orientation of $\mathcal{S}$, the good configurations are those, where nearly all nodes work synchronously. Since the beginning of a slot is at $\gamma = 0$ or $\gamma = T$, we define the slot distance of two nodes $v, w \in V$ in a configuration $c \in \mathrm{Conf}$ by $dist_c(v, w) = dist(\gamma_v + T\mathbb{Z}, \gamma_w + T\mathbb{Z})$, where γ_v is the current value of γ at node v and $dist(X, Y) = \inf\{|x - y| \; : \; x \in X, y \in Y\}$ is the usual distance between sets of numbers. The slot distance $dist_c(v, w) \in [0, \frac{T}{2}]$ is the amount of time elapsed between the beginning of the slot of one node and the beginning of the slot of the other node. For a good configuration the slot distances should be low: $b(c) = 1 - \frac{\sum\limits_{v,w \in V} dist_c(v,w)}{|V^2| \cdot T/2}$. In this example, the level of target orientation does not depend on the initialization (Γ, P_Γ), so the mean value $E(b_\theta(\mathrm{Conf}_t))$ can be calculated by using an arbitrary start configuration as a deterministic initialization. We have calculated the level of resilience for a complete graph G with the parameters, which had also been used for the analysis in [18]:

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

We have calculated $\mathrm{TO}_t(\mathcal{S}, \Gamma) = E(b(\mathrm{Conf}_t))$ for $t \leq 10^6$ steps. Then we can approximate the level of target orientation by $\mathrm{TO}(\mathcal{S}, \Gamma) \approx \mathrm{Avg}_{[10^5, 10^6]}(t \mapsto \mathrm{TO}_t(\mathcal{S}, \Gamma))$. The result is $\mathrm{TO}(\mathcal{S}, \Gamma) \approx 0,997$, so the system has a very high level of target orientation. This can also be derived from the results of [18]: After the groups of synchronizations are built, the slot distances are zero for almost every pair of nodes, so $\mathrm{TO}_t(\mathcal{S}, \Gamma) \approx 1$ and therefore the system is target orientated: $\mathrm{TO}(\mathcal{S}, \Gamma) \approx 1$.

Now we consider the level of resilience. Assume that a set $\theta \subseteq V$ of nodes have a breakdown in the wireless network. Then the parameter set Θ consists of all subsets of V, i.e. $\Theta = \mathfrak{P}(V)$ and $p_\Theta(\theta) \in [0,1]$ is the probability that the nodes of θ brake down and the other nodes are still working. After the breakdown we have a system $\mathcal{S}^\theta$, where the automaton $Z_{\theta,v}$ does not send any pulses for $v \in \theta$ and $Z_{\theta,v} = a_v$ works like before for $v \in V \setminus \theta$. The valuation map b_θ describes the good configurations, i.e. all working nodes should still be able to synchronize: $b_\theta(c) = 1 - \frac{\sum\limits_{v,w \in V \setminus \theta} dist_c(v,w)}{|(V \setminus \theta)^2| \cdot T/2}$. If the working nodes still form a strongly connected graph, they are able to synchronize. If G is the complete graph, where each node is connected to each other node, then the system is resilient: $\mathrm{Res}(\mathcal{S}, \Gamma) \approx 1$. If G is not the complete graph then the working nodes might not be strongly connected anymore, so the synchronization only take place inside each connection component. In this case we have $\mathrm{Res}(\mathcal{S}, \Gamma) < 1$, where the exact value depends on the graph G and on the probabilities p_Θ: If $p_\Theta(\theta)$ is small for large sets θ, then with high probability, the working nodes still form a connected graph, so $\mathrm{Res}(\mathcal{S}, \Gamma)$ is near 1.

Instead of a breakdown of nodes, we also can model the resilience with respect to an intruder at a node $v_0 \in V$, who wants to disturb the communication. In this case, the parameter set Θ can be used to describe the behavior of the intruder. Here we use Θ as a discrete subset of $\mathbb{R}^+$, where $\theta \in \Theta$ is the duration between two consecutive pulses, that the intruder sends periodically to the neighbors. The system $\mathcal{S}^\theta$ is the system $\mathcal{S}$ after replacing the automaton a_{v_0} by Z_{v_0} and leave all other automatons as they are: $Z_v = a_v$ for $v \neq v_0$. The good configurations are those, where all other nodes are synchronized: $b_\theta(c) = 1 - \frac{\sum\limits_{v,w \in V \setminus \{v_0\}} dist_c(v,w)}{|(V \setminus \{v_0\})^2| \cdot T/2}$. If the graph without the node v_0 is not connected anymore, then the two connection components are not able to synchronize anymore, so we get $\mathrm{Res}(\mathcal{S}, \Gamma) < 1$. For the complete graph with the parameters, which have already been used above for the target orientation, we calculated the level of resilience (with the approximation $\mathrm{Res}(\mathcal{S}, \Gamma) \approx \mathrm{Avg}_{[10^5, 10^6]}(t \mapsto \mathrm{Res}_t(\mathcal{S}, \Gamma))$). The results for different sets Θ with the uniform distribution $p_\Theta : \Theta \to [0,1]$ are given in Table 1.

Θ	$\{45\}$	$\{70\}$	$\{100\}$	$\{120\}$	$\{150\}$	$\{\,45, 70, 100, 120, 150\,\}$
$\mathrm{Res}(\mathcal{S}, \Gamma)$	0.987	0.985	0.996	0.991	0.996	0.991

Table 1. Level of Resilience

Therefore the system in this model has a high level of resilience with respect to an intruder, which periodically sends pulses.

Now let us consider the level of homogeneity. If the in-degree $|v - |$ of all nodes are the same (e.g. in a complete graph), then the local configuration $\mathrm{Conf}_{t,v}$ has the same entropy for all nodes v since all automatons are the identical and the initialization is uniform. Therefore the have $\mathrm{Ho}_t(\mathcal{S}, \Gamma) =$

$1 - \frac{\sum\limits_{v,w \in V, v < w} |H(\mathrm{Conf}_{t,v}) - H(\mathrm{Conf}_{t,w})|}{\sum\limits_{v,w \in V, v < w} \max(H(\mathrm{Conf}_{t,v}), H(\mathrm{Conf}_{t,w}))} = 1$ and the system is homogeneous. If the
in-degrees of the nodes differ, then the entropies $H(\mathrm{Conf}_{t,v})$ and $H(\mathrm{Conf}_{t,w})$
need not be the same, but after the two groups of synchronization have been
built, we have $H(\mathrm{Conf}_{t,v}) \approx H(\mathrm{Conf}_{t,w})$, because the only difference of these
entropies is the information which predecessor $q \in v-$ belongs to which group
during the transmission state. So in this case the system is nearly homogeneous.

For the level of adaptivity, we note that the system has no external nodes.
For the valuation map, which we already used for the level of target orientation,
we get $\mathrm{Ad}(\mathcal{S}, \Gamma) = \mathrm{TO}(\mathcal{S}, \Gamma) = 1$.

9   Discussion of the results

The main result of this paper is, that we get a formalism to analyze complex
systems with respect to self-organizing properties. For the adaptivity, target
orientation, homogeneity and resilience the quantitative definitions given in the
previous sections can be useful for the analysis of complex systems.

When we have a real world system and we would like to analyze this system
with respect to self-organizing properties, we first create the model with the
definitions of Section 3. The connections between the objects are described by the
multigraph and the behavior of the objects are described stochastic automatons.
In this model we can apply the definitions of sections 4-7 to calculate

- the level of target orientation,
- the level of adaptivity,
- the level of resilience,
- the level of homogeneity.

The level of target orientation shows, how good the system satisfies the goal,
for which the system is designed for. The adaptivity and the resilience are two
of the main properties of self-organizing systems, so together with the level of
emergence and the level of autonomy of [13] they are also indicators, how self-
organizing the system is. Finally, the level of homogeneity shows how different
or equal the nodes behave.

One major problem with the definitions of these levels is the complexity: For
a very complex system, it might be difficult to compute the exact values for the
levels. But the examples in this paper show, that even if the system is too large
to compute the exact levels, the definitions can still be useful to get an approx-
imation of these levels. For example, if we consider the level of homogeneity for
the slot synchronization in Section 8, then we can derive $\mathrm{Ho}(\mathcal{S}, \Gamma) \approx 1$ from
Definition 8 without computing the exact value.

10   Conclusion

In this paper we described how self-organzing properties of complex systems can
be measured quantitatively. While [13] and [14] have already given definitions

for the emergence and autonomy, the novel contribution of this paper are the quantitative definitions for some other properties of self-organizing systems:

– adaptivity,
– target orientation,
– homogeneity,
– resilience.

These definitions may help for the analysis of existing systems and for the design of new systems with self-organizing properties.

References

1. H. De Meer and C. Koppen, "Characterization of self-organization," in *Peer-to-Peer Systems and Applications* (R. Steinmetz and K. Wehrle, eds.), vol. 3485 of *Lecture Notes in Computer Science*, pp. 227–246, Springer-Verlag, 2005.
2. F. P. Heylighen, "The science of self-organization and adaptivity," in *Knowledge Management, Organizational Intelligence and Learning, and Complexity* (L. D. Kiel, ed.), The Encyclopedia of Life Support Systems, EOLSS Publishers, 2003.
3. G. Nicolis and I. Prigogine, *Self-Organization in Non-Equilibrium Systems: From Dissipative Structures to Order Through Fluctuations.* Wiley, 1977.
4. C. R. Shalizi, *Causal Architecture, Complexity and Self-Organization in Time Series and Cellular Automata.* PhD thesis, University of Wisconsin-Madison, 2001.
5. H. von Foerster, *Self-Organizing Systems*, ch. On Self-Organizing Systems and their Environments, pp. 31–50. Pergamon, 1960.
6. W. R. Ashby, *Principles of Self-organization*, ch. Principles of the Self-organizing System, pp. 255–278. Pergamon, 1962.
7. F. Heylighen and C. Joslyn, "Cybernetics and second order cybernetics," *Encyclopedia of Physical Science & Technology*, vol. 4, pp. 155–170, 2001.
8. H. Haken, *Self-organizing Systems: An Interdisciplinary Approach*, ch. Synergetics and the Problem of Selforganization, pp. 9–13. Campus Verlag, 1981.
9. C. Gershenson, *Design and Control of Self-organizing Systems.* PhD thesis, Vrije Universiteit Brussel, Brussels, Belgium, May 2007.
10. N. Boccara, *Modeling Complex Systems.* Springer-Verlag, 2004.
11. G. Di Marzo Serugendo, N. Foukia, S. Hassas, A. Karageorgos, S. K. Mostfaoui, O. F. Rana, M. Ulieru, P. Valckenaers, and C. Van Aart, "Self-organisation: Paradigms and applications," in *Proc. 11th International Conference on Analysis and Optimization of Systems – Discrete Event Systems*, vol. 2977 of *Lecture Notes in Computer Science*, pp. 1–19, Springer-Verlag, 2004.
12. R. Holzer and H. de Meer, "On modeling of self-organizing systems," in *Autonomics 2008*, 2008.
13. R. Holzer, H. de Meer, and C. Bettstetter, "On autonomy and emergence in self-organizing systems," in *IWSOS 2008*, Springer, 2008.
14. M. Mnif and C. Mueller-Schloer, "The quantitative emergence," in *Proc. of the 2006 IEEE Mountain Workshop on Adaptive and Learning Systems (SMCals 2006)*, pp. 78–84, IEEE, 2006.
15. C. Auer, P. Wuechner, and H. de Meer, "The degree of global-state awareness in self-organizing systems," in *IWSOS 2009*, Springer, 2009.
16. T. M. Cover and J. A. Thomas, *Elements of Information Theory.* Wiley, 2nd ed., 2006.

17. C. Auer, P. Wuechner, and H. de Meer, "Target-oriented self-structuring in classi-
fying cellular automata," in *Automata 2009*, 2009.
18. A. Tyrrell, G. Auer, and C. Bettstetter, "Biologically inspired synchronization
for wireless networks," in *Advances in Biologically Inspired Information Systems:
Models, Methods, and Tools* (F. Dressler and I. Carreras, eds.), vol. 69 of *Studies
in Computational Intelligence*, pp. 47–62, Springer, 2007.
19. R. Mirollo and S. Strogatz, "Synchronization of pulse-coupled biological oscilla-
tors," *SIAM Journal of Applied Mathematics*, vol. 50, pp. 1645–1662, 1990.