Methods for Approximations of Quantitative Measures in Self-Organizing Systems

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 micro-level modeling a multigraph can be used to describe the connections between objects. The behavior of the objects in the system can be described by (stochastic) automatons. In such a model, quantitative measures can be defined for the analysis of the systems or for the design of new systems. Due to the high complexity, it is usually impossible to calculate the exact values of the measures, so approximation methods are needed. In this paper we investigate some approximation methods to be able to calculate quantitative measures in a micro-level model of a complex system. To analyze the practical usability of the concepts, the methods are applied to a slot synchronization algorithm in wireless sensor networks.

Key words: Self-Organziation, Mathematical modeling, Systems, Quantitative measures, Approximation

1 Introduction

Self-organizing systems provide mechanisms to manage themselves as much as possible to reduce administrative requirements for users and operators. Since such systems are usually very complex, mathematical models are used for the analysis of the systems and for the design of new systems. In micro-level modeling the behavior of each entity of the system and the communication between the entities are described. Macro-level modeling uses the technique of aggregation to derive a model for the system variables of interest. Each macro-state can be seen as an equivalence class of micro-states. Quantitative measures [1] [2] can be used as a link from micro-level modeling to macro-level modeling. These measures are defined in a micro-level model and they measure a global property of the system. The dynamic change of the values of the measures during the time yields a macro-level model. These measures allow the analysis of the existing systems with respect to self-organizing properties like autonomy or emergence

[0] This research is partially supported by the SOCIONICAL project (IP, FP7 Call 3, ICT-2007-3-231288), by the ResumeNet project (STREP, FP7 Call 2, ICT-2007-2-224619) and by the Network of Excellence EuroNF (IST, FP7, ICT-2007-1-216366).

and to optimize some system parameters with respect to a given goal. Also for the design of new systems the measures can help to compare different rule sets or different system parameters for optimization. Further discussions about the importance of quantitative measures can be found in [1], [2]. A practical example, where a quantitative measure is used for an optimization of system parameters in an intrusion detection system can be found in [3].

Unfortunately, many systems are too complex to be able to calculate the exact values of the quantitative measures, because the measures usually consider the global state space, which grows exponentially with the number of entities.

This paper investigates some approximation methods for the calculation of quantitative measures. Section 2 gives an overview of the related work and Section 3 recalls the micro-level model of [2] for complex systems. In Section 4 some approximation methods for quantitative measures are explained. In Section 5 we investigate, how the approximation methods can be used for the calculation of the quantitative measures for emergence, target orientation and resilience. In Section 6, we apply the approximation methods to the model of slot synchronization in wireless networks, which was defined in [1], [2]. Section 7 contains some discussions and advanced conclusions of the methods described in this paper.

2 Related work

In the last years, much research has been done in the field of self-organizing systems. The main properties of self-organizing systems [4] [5] are self-maintenance, adaptivity, autonomy, emergence, decentralization, and optimization. A non-technical overview of self-organization can be found in [5]. Other definitions and properties of self-organizing systems can be found in information theory [6], thermodynamics [7], cybernetics [8], [9], [10] and synergetics [11]. [12] gives a description of the design of self-organizing systems. [13] gives a systematic overview on micro-level and macro-level modeling formalisms suitable for modeling self-organizing systems. A survey about practical applications of self-organization can be found in [14]. For modeling continuous self-organizing systems and a comparison between discrete and continuous modeling see [15]. Quantitative measures of autonomy, emergence, adaptivity, target orientation, homogeneity, resilience and global-state awareness, can be found in [1], [2], [13], [16] and [3]. Other approaches to quantitative emergence are given in [16] and [17]. In [16] emergence is defined as an entropy difference by considering the change of order within the system. [17] analyses and improves this approach by introducing a multivariate entropy measure for continuous variables, and by using different divergence measures for the comparison of the corresponding density functions. The Parzen window approach, which is used in this paper, was also used in [17] as an estimation method for the density functions.

In this paper we use the definition of emergence of [1], which measures dependencies between communications in the system. The methods of [1], [2] for discrete micro-level modeling with stochastic automatons are also used in this paper. The major drawback of the measures defined in [1], [2] was, that the

state space in practical applications is usually too large to be able to calculate the measures analytically. This paper contributes to this problem: We analyze different approximation methods with respect to usability for the approximation of quantitative measures in complex systems.

3 Discrete micro-level model

For modeling discrete systems, we use the methods of [2], which are based on the ideas of [5]: The topology of a system is represented by a multigraph G, where V is the set of vertices and K is the set of directed edges. The behavior of each entity $v \in V$ is described by a stochastic automaton a_v, which has a set of internal states S_v and uses an alphabet A of symbols to communicate with the automatons of other nodes by sending a symbol through an edge $k \in K$ to a successor node. Since the automaton does not need to be deterministic, probability distributions are used to describe the outputs and the state transition function of each stochastic automaton. The influence of the environment on the system can be modeled by special vertices (external nodes) in the multigraph [2]. With these concepts micro-level models can be built for a wide variety of complex systems of the real world, e.g. systems that appear in biology, physics, computer science or any other field. 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.

When 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 called *configuration*. It represents a snapshot of the system. For a configuration c and a set $T \subseteq K$ of edges the assignment of the edges in T in the configuration is also denoted by $c|_T$.

To analyze the behavior of a system, we initialize it at time $t_0 = 0$ by choosing a start configuration $c_0 \in \Gamma$, where Γ is the set of all possible start configurations and $P_\Gamma(c)$ is the probability, that $c \in \Gamma$ is used for the initialization of the system. Then the automatons produce a sequence $c_0 \to c_1 \to c_2 \to \dots$ of configurations during the run of the system. Since the automata and the initialization are not deterministic, the sequence $c_0 \to c_1 \to c_2 \to \dots$ 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.

For measuring the information in a system we use the statistical entropy: For a discrete random variable X taking values from a set W the *entropy* $H(X)$ *of* X is defined by [18]

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

The entropy measures, how many bits are needed to encode the outcome of the random variable in an optimal way. For example, the entropy $H(Conf_t)$ measures the amount of information contained in the configuration of the system at time t. The entropy $H(Conf_t|_K)$ measures the amount of information, which is communicated between the entities at time t.

4 Methods for approximations

For the analysis of global properties of the system, quantitative measures can be used. Unfortunately, entropies of global random variables like $H(Conf_t)$ are difficult to calculate analytically, because complex systems have a huge global state space: The set of all configurations grows exponentially with the number of the entities in the system. Therefore we need methods to approximate the needed values. We can use simulation runs to be able to approximate probabilities and entropies. Let R be the number of simulation runs. Each simulation run leads to a time series, which is the configuration sequence $c_0 \to c_1 \to c_2 \to \ldots$ produced by the simulation run. Then the probability of a value $a \in A$ on a single edge $k \in K$ at time $t \geq 0$ can be approximated by the relative frequency $rel_{t,k,a}$, which is the number of time series, which contains the value a on the edge k at time t, divided by R:

$$P(Conf_t|_{\{k\}} = a) \approx rel_{t,k,a} := \frac{1}{R} \cdot r_{t,k,a}$$
$$r_{t,k,a} := |\{s \ : \ s \text{ is a time series with } Conf_t|_{\{k\}} = a\}|$$

Unfortunately, this method is still too complex for probabilities of global valuations like $P(Conf_t = c)$ or $P(Conf_t|_K = \underline{a})$ for $\underline{a} \in A^K$. The range of values for these random variables is too large, so the relative frequencies received from simulation runs are too inaccurate for the approximation of probabilities. For such global probabilities, we investigate three different approximation methods:

1. Classification
   The probability space is divided into different classes. The relative frequency for each class is calculated by considering the time series. The probability for a single element of the class can be approximated by the relative frequency of the class divided by the size of the class. For $P(Conf_t|_K = \underline{a})$ the classification can be done by choosing a subset $K_0 \subseteq K$ and build $|A|^{|K_0|}$ equivalence classes $[b] := \{a \in A^K \ : \ a|_{K_0} = b\}$ for $b \in A^{K_0}$, where the size of each class is $|[b]| = |A|^{|K|-|K_0|}$. Now the relative frequencies $rel_{t,K_0,[b]}$ are calculated for each class $[b]$ and the probability $P(Conf_t|_K = \underline{a})$ is approximated by $P(Conf_t|_K = \underline{a}) \approx \frac{1}{|[b]|} \cdot rel_{t,K_0,[b]}$ for $b = \underline{a}|_{K_0}$. Analogously this concept can be used for the approximation of $P(Conf_t = c)$: In this case we do not only consider the valuations $c|_K$ of edges but also the internal states of the nodes. The classification can be done by choosing a subset $V_0 \subseteq V$ of nodes and a subset $K_0 \subseteq K$ of edges. Then each class is characterized by a valuation $b : K_0 \to A$ of K_0 and the states $g \in \prod_{v \in V_0} S_v$ of the nodes

$v \in V_0$. The number of equivalence classes is $A^{K_0} \cdot \prod_{v \in V_0} |S_v|$. For example, if we have $A = \{0, 1\}$ and a single edge $K_0 = \{k\}$ and $V_0 = \emptyset$, then we have two equivalence classes of configurations: In the first class are all configurations c with $c_K(k) = 0$, and in the second class all configurations have the value 1 on k. After calculating the approximations of the probabilities, we get an approximation of the entropies. For the entropy $H(\mathrm{Conf}_t\,|_K)$ of the edge valuations we have

$$\begin{aligned} H(\mathrm{Conf}_t\,|_K) &= -\sum_{\underline{a} \in A^K} P(\mathrm{Conf}_t\,|_K = \underline{a}) \log_2 P(\mathrm{Conf}_t\,|_K = \underline{a}) \\ &= -\sum_{[b]\mathrm{class}} \sum_{\underline{a} \in [b]} P(\mathrm{Conf}_t\,|_K = \underline{a}) \log_2 P(\mathrm{Conf}_t\,|_K = \underline{a}) \\ &\approx -\sum_{[b]\mathrm{class}} \sum_{\underline{a} \in [b]} \frac{1}{|[b]|} \cdot rel_{t,K_0,[b]} \log_2 \left(\frac{1}{|[b]|} \cdot rel_{t,K_0,[b]} \right) \\ &= -\sum_{[b]\mathrm{class}} rel_{t,K_0,[b]} \log_2 \left(\frac{1}{|[b]|} \cdot rel_{t,K_0,[b]} \right) \end{aligned}$$

A similar formula can be obtained for the approximation of the entropy $H(Conf_t)$, where the classes are defined not only by edge valuations but also by some internal states of some nodes. Note that this approximation method for $H(\mathrm{Conf}_t\,|_K)$ and $H(\mathrm{Conf}_t)$ is also efficient for a large number of classes (low size of each class $[b]$), since only the summands with $rel_{t,K_0,[b]} \neq 0$ have to be taken into account.

2. Parzen window approach
For a random variable X with a sample set $W = \{w_1, \ldots, w_R\} \subseteq \mathbb{R}^{dim}$ (observations of X) we can use the kernel density estimator based on a Gaussian kernel [17], [19]

$$p(a) = \frac{1}{R} \sum_{j=1}^{R} \frac{1}{(2\pi h^2)^{dim/2}} \exp(-\frac{1}{2} \frac{dist(a, w_j)^2}{h^2})$$

with

$$\begin{aligned} dim = &\text{ dimension of the random variable } X \\ a \in \mathbb{R}^{dim} & \\ p(a): &\text{ approximation of the density of } X \text{ at } a \\ R: &\text{ number of samples for } X \\ dist(a, w_j): &\text{ Euclidean distance between } a \text{ and } w_j \\ h: &\text{ user-defined parameter [20] for changing variance and bias} \end{aligned}$$

By integrating over the density function p, we can calculate probabilities for the random variable X. But in our case, we consider discrete systems,

so the random variables (e.g. $Conf_t$) are discrete. If we assume that the random variable only yields integer values for each component (i.e. $X \in \mathbb{Z}^{dim}$), then we can use $P(X = c) = P(dist_\infty(X, c) \leq \frac{1}{2})$ for the approximation, where $dist_\infty(a, b) = \max\{|a_i - b_i| \ : \ i = 1, 2, \ldots, dim\}$ of vectors $a, b \in \mathbb{R}^{dim}$ is the distance induced by the maximum norm on $\mathbb{R}^{dim}$. Since the set $\{a \mid dist_\infty(a, c) \leq \frac{1}{2}\}$ is a hypercube of size 1, the value $P(dist_\infty(X, c) \leq \frac{1}{2})$ can be approximated directly with the density function p, i.e. $P(X = c) \approx p(c)$. This approximation can then be used to get approximations of the entropies $H(\mathrm{Conf}_t)$ and $H(\mathrm{Conf}_t\,|_K)$.

3. Restriction of the set of initial configurations
When we have a system, in which large parts are deterministic, then a restriction of the set of the initial configurations reduces the complexity. Let $\Gamma_0 \subseteq \Gamma$ be a set of initial configurations. Then the time series are received from simulation runs starting in Γ_0. If all automatons are deterministic, two simulation runs with the same initial configuration $c_0 \in \Gamma_0$ would lead to the same time series, so for each initial configuration c_0 at most one simulation run is needed. If some automata are stochastic, the same initial configuration might lead to different time series. The entropy $H(\mathrm{Conf}_t\,|_K)$ (and analogously $H(\mathrm{Conf}_t)$) can then be derived by using the relative frequency $rel_{t,\underline{a}}$ of a value $\underline{a} \in A^K$ at time t as an approximation for the probability $P(\mathrm{Conf}_t\,|_K = \underline{a})$:

$$\begin{aligned} H(\mathrm{Conf}_t\,|_K) &= -\sum_{\underline{a} \in A^K} P(\mathrm{Conf}_t\,|_K = \underline{a}) \log_2 P(\mathrm{Conf}_t\,|_K = \underline{a}) \\ &\approx -\sum_{\underline{a} \in A^K} rel_{t,\underline{a}} \log_2 rel_{t,\underline{a}} \end{aligned}$$

As for the method of classification, this sum can efficiently be calculated, since only the summands with $rel_{t,\underline{a}} \neq 0$ have to be taken into account.

5 Quantitative measures

In this section we investigate some quantitative measures for global properties, which have been proposed in [1], [2] and [13]. In the following, let $\mathcal{S}$ be a system and (Γ, P_Γ) be an initialization.

Emergence

The level of emergence measures global patterns in the system by considering the dependencies between the valuations of different edges. For a point of time $t \geq 0$ the *level of emergence* at time t is defined by [1]

$$\varepsilon_t(\mathcal{S}, \Gamma) = 1 - \frac{H(\mathrm{Conf}_t\,|_K)}{\sum_{k \in K} H(\mathrm{Conf}_t\,|_{\{k\}})}.$$

The level of emergence is always a value in the interval [0, 1]. If at the current point of time $t \geq 0$ there are large dependencies between the values on

the single edges (which can be seen as patterns), the level of emergence is high: $\varepsilon_t(\mathcal{S}, \Gamma) \approx 1$. If the values of nearly all edges are independent, there will be no pattern, so the level of emergence is low: $\varepsilon_t(\mathcal{S}, \Gamma) \approx 0$. Therefore the map $t \mapsto \varepsilon_t(\mathcal{S}, \Gamma)$ measures the dependencies occurring during the whole run of the system.

To be able to approximate the values $H(\mathrm{Conf}_t|_{\{k\}})$, we use the relative frequencies of the values in the time series for the approximations of $P(\mathrm{Conf}_t|_{\{k\}} = a)$ for $a \in A$. We have the approximation (see Section 4)

$$P(\mathrm{Conf}_t|_{\{k\}} = a) \approx rel_{t,k,a} = \frac{r_{t,k,a}}{R},$$

where R is the number of all time series. For the approximation of $H(\mathrm{Conf}_t|_K)$ we can apply the methods of section 4. This leads to an approximation of the level of emergence. In Section 6 we will calculate the level of emergence for an example system, and in section 7 we will discuss the results.

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. To measure the target orientation, a valuation map $b : \mathrm{Conf} \to [0, 1]$ for the configurations can be used to describe 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. For a point of time $t \geq 0$ the *level of target orientation* [2] of $\mathcal{S}$ at time t is defined by $\mathrm{TO}_t(\mathcal{S}, \Gamma) = E(b(\mathrm{Conf}_t))$, where E is the mean value of the random variable. The level of target orientation measures the valuations $b(c)$ of the configurations during the run of a system.

Assume that we have R time series received from simulation runs. Then we get the approximation $\mathrm{TO}_t(\mathcal{S}, \Gamma) \approx \frac{1}{R} \sum_{j=1}^{R} b(\mathrm{Conf}_{t,j})$, where $\mathrm{Conf}_{t,j}$ is the configuration of the j-th simulation run at time t.

Resilience

To measure the resilience of a system, an 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 or it could be the behavior of an intruder. The system is resilient if despite the malfunctioned nodes the system still runs through many "good" configurations. 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. 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* [2] 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. Therefore the system is resilient

if despite the malfunctioned nodes the system still runs through many "good" configurations.

As for the level of target orientation, we get the approximation $\mathrm{Res}_t(\mathcal{S}, \Gamma) \approx \frac{1}{R} \sum_{j=1}^{R} b(\mathrm{Conf}_{t,j}^{\Theta})$, where $\mathrm{Conf}_{t,j}^{\Theta}$ is the configuration of the j-th simulation run in the changed system at time t.

6 Slot synchronization in wireless networks

In this section we apply the methods of the previous sections to a self-organized slot-synchronization algorithm in wireless networks [21]. The access to the wireless medium is organized in time slots. The distributed algorithm for slot-synchronization is based on the model of pulse-coupled oscillators by Mirollo and Strogatz [22].

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 [21] the pulse-coupled oscillator synchronization model is adapted to wireless systems, where also delays (e.g., transmission delay, decoding delay) are considered.

The discrete micro-level model is described in [1], so we omit the definition of the model here.

Now we compare the results of the quantitative measures calculated by the different approximation methods described in section 4. For each case, we used a complete graph G with 30 nodes with the parameters, which had also been used for the analysis in [21]. Starting from a random initialization, the synchronization usually takes about 500-800 time steps, so for our analysis we use a point in time t, which is greater than 800, such that the nodes have already been synchronized.

Concerning the level of target orientation of $\mathcal{S}$, the good configurations are those, where nearly all nodes work synchronously, so for the valuation b of the configurations we measure the slot distances[1] $dist_c(v, w)$ for $v, w \in V$ in each configuration c. The slot distance 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. The valuation is given by $b(c) = 1 - \frac{\sum_{v,w \in V} dist_c(v,w)}{|V^2| \cdot T/2}$, where T is the length of a slot. The mean value $\mathrm{TO}_t(\mathcal{S}, \Gamma) = E(b(\mathrm{Conf}_t))$ is approximated by the relative frequencies $\mathrm{TO}_t(\mathcal{S}, \Gamma) \approx \frac{1}{R} \sum_{j=1}^{R} b(\mathrm{Conf}_{t,j})$ as described in Section 5. For $R = 300$ simulation runs, the result is $\mathrm{TO}_t(\mathcal{S}, \Gamma) \approx 0.996$, so the system has a very high level of target orientation: After the groups of synchronizations are built, the

[1] see [2]

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 oriented.

Now we consider the level of 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: $b_\theta(c) = 1 - \frac{\sum\limits_{v,w \in V \setminus \{v_0\}} dist_c(v,w)}{|(V \setminus \{v_0\})^2| \cdot T/2}$. 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 method of Section 5: $\mathrm{Res}_t(\mathcal{S}, \Gamma) \approx \frac{1}{R} \sum\limits_{j=1}^{R} b(\mathrm{Conf}_{t,j}^{\Theta})$. For $\Theta = \{60, 80\}$ and $R = 300$ the approximated level of resilience is

$$\mathrm{Res}_t(\mathcal{S}, \Gamma) \approx \frac{1}{R} \sum_{j=1}^{R} b(\mathrm{Conf}_{t,j}^{\Theta}) \approx 0.988$$

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

Concerning the level of emergence we measure the dependencies between the communications in the system:

$$\varepsilon_t(\mathcal{S}, \Gamma) = 1 - \frac{H(\mathrm{Conf}_t\,|_K)}{\sum\limits_{k \in K} H(\mathrm{Conf}_t\,|_{\{k\}})}.$$

Each pulse is represented by the value $\mathrm{Conf}_t\,|_{\{k\}} = 1$ on the edge k, and the value 0 is used, if no pulse is sent. The values $H(\mathrm{Conf}_t\,|_{\{k\}})$ can be approximated by the relative frequencies of the values in the time series. For the complete graph with 30 nodes we have $|K| = 870$, so for the calculation of $H(\mathrm{Conf}_t\,|_K)$ there exist 2^{870} different edge valuations, which is much too large to be able to calculate the entropy analytically. Therefore we apply the three approximation methods mentioned in Section 4. Let us first consider the classification of nodes. Since all elements $\underline{a} \in [b]$ of an equivalence class get the same probability $P(Conf_t|_K = \underline{a}) \approx \frac{1}{|[b]|} \cdot rel_{t,K_0,[b]}$ for $b = a|_{K_0}$, this methods leads to an increase of the entropy: The relative frequency $rel_{t,K_0,[b]}$ is equally distributed under the elements of $[b]$, while in the time series the random variable $Conf_t|_K$ might hit some elements of $[b]$ more than one time, while other elements do not appear at all as values of $Conf_t|_K$. Therefore this approximation leads to a value $\frac{1}{|[b]|} \cdot rel_{t,K_0,[b]}$, which is higher than the exact value $P(Conf_t|_K = \underline{a})$, where the error depends on the size of the classes: The larger the classes $[b]$, the higher is the approximated value. Table 1 shows the approximations of the level of emergence at time $t = 1000$ in dependency of the parameter $|K_0|$ with $R = 300$ simulation runs. For small sets

$\lvert K_0 \rvert$	870	860	700
$\lvert [b] \rvert$	1	2^{10}	2^{170}
$\varepsilon_t(\mathcal{S}, \Gamma)$	0.988	0.980	0.773

Table 1. Approximation of the level of Emergence by classification

K_0, this method leads to a negative level of emergence, so this approximation method is not useful in this case.

A similar problem appears for the Parzen window approach discussed in section 4: In this case the probabilities are not equally distributed, but the entropy is increased anyhow, because the Gaussian kernel leads to an entropy, which is higher than the exact value of $H(\mathrm{Conf}_t \,|_K)$. But an increase in the parameter h leads to a decrease in the entropy $H(\mathrm{Conf}_t \,|_K)$, so the level of emergence $\varepsilon_t(\mathcal{S}, \Gamma)$ grows with increasing h. Therefore, the right parameter h has to be found to get a good approximation with the Parzen window approach. Table 2 shows the approximations of the level of emergence at time $t = 1000$ in dependency of the parameter h with $R = 300$ simulation runs. Note that a small change in the parameter h has a large impact on the result.

h	0.478	0.479	0.4795	0.48
$\varepsilon_t(\mathcal{S}, \Gamma)$	0.778	0.956	0.981	0.991

Table 2. Approximation of the level of Emergence by Parzen window

Concerning the methods of the restriction of initial configurations, we choose a set $|\Gamma_0|$ of initial configurations and use the time series of $R = |\Gamma_0|$ simulation runs. The entropy $H(\mathrm{Conf}_t \,|_K)$ grows with the size $|\Gamma_0|$, so the level of emergence $\varepsilon_t(\mathcal{S}, \Gamma)$ decreases with increasing $|\Gamma_0|$. Table 3 shows the results in dependency of the number $|\Gamma_0|$ of the used initial configurations.

$\lvert \Gamma_0 \rvert = R$	10	1000	2000
$\varepsilon_t(\mathcal{S}, \Gamma)$	0.99	0.984	0.982

Table 3. Approximation of the level of Emergence by restriction of initial configurations

7 Discussion of the results

Quantitative measures can be defined for the analysis of existing systems and for the design of new systems. Due to the high complexity, it is usually impossible to

calculate the exact values of the measures. The main result of this paper is, that we get approximation methods to be able to calculate quantitative measures in self-organizing systems. We investigated different methods for approximations:

- Mean values $E(X)$ of random variables can be approximated by calculating the arithmetic mean value of the samples received from simulation runs.
- Entropies $H(X)$ of random variables are based on probabilities $P(X = w)$ for the values of the random variable, so each method for the approximation of the probabilities leads to an approximation method for the entropy.
- Probabilities $P(X = w)$ of values for random variables can be approximated by relative frequencies. For random variables with a small range of values, this method usually gives good approximations of the probability, while for random variables with a large range of values, other methods are needed, since the relative frequency for all values w might be very low.
  - Classification
    By choosing a classification on the probability space, the problem is reduced to a smaller set of classes, but with a lower accuracy of the result. Since the probabilities are stretched through the elements of the classes, we get a higher entropy. The larger the classes, the more inaccurate is the result for the quantitative measure.
  - Parzen window approach
    An approximation of the density function is derived from the sample values of the random variable by using the Gaussian kernel. The main problem with this approach is to estimate the parameter h for the density function, which is usually not known in advance. A small change in the parameter may lead to a large change in the result of the quantitative measure.
  - Restriction of initial configurations
    By choosing a subset of initial configurations, the accuracy of the approximation of the probabilities with relative frequencies might be increased, especially if large parts of the system are deterministic. Consider for example the approximation of $\mathrm{Conf}_t\,|_K$ for some $t > 0$ by using the relative frequencies of valuations $\underline{a} : K \to A$ for the probability $P(\mathrm{Conf}_t\,|_K = \underline{a})$. If arbitrary initial configurations are used, then it might happen that the random variable $\mathrm{Conf}_t\,|_K$ takes R different values in the R simulation runs, which leads to two different values for the probability: $P(\mathrm{Conf}_t\,|_K = \underline{a}) = \frac{1}{R}$ if $\underline{a}$ is reached in a simulation run at time t and $P(\mathrm{Conf}_t\,|_K = \underline{a}) = 0$ otherwise. By restricting the set of initial configurations, a valuation $\underline{a}$ might be reached in more than one simulation run, so different relative frequencies can be distinguished, which might lead to a better estimation of the entropy than the binary information "$\underline{a}$ is reached" or "$\underline{a}$ is not reached". For systems, where nearly all automatons are nondeterministic, the advantage of this method is very limited, since the problem remains the same: The relative frequencies for all values w might be very low.

These results about the different analysis methods are also confirmed by the case study in Section 6:

- The approximation of the mean value for the target orientation leads to a very high value $\mathrm{TO}_t(\mathcal{S}, \Gamma) \approx 0.996$, which is very close to the exact value, because the slot distances are zero for almost every pair of nodes after the groups of synchronizations have been built.
- For the approximated value $\mathrm{Res}_t(\mathcal{S}, \Gamma) \approx 0.988$ for the resilience it may be difficult to estimate the exact value, but intuitively it is clear that a single malfunctioned node has only few influence on the whole network with 30 nodes, which form a complete graph, so synchronization between the normal nodes should still be possible, and we can assume that also in this case the exact value for the level of resilience is near 1, so the approximated value has a high accuracy.
- Concerning the emergence, the results are much worse: The values in table 1 indicate, that large classes lead to a decrease of the approximation of the level of emergence, while small classes are useless, because probabilities can not be well approximated by relative frequencies, if some classes are only reached few times in the simulation runs. Table 2 shows that an inaccurate estimation for the parameter h in the Parzen window approach leads to completely different results for the level of emergence. For the method of the restriction of initial configurations, we used $R = |\Gamma_0|$ in Section 6 because all automata are deterministic, so the time series is uniquely determined by the initial configuration.

Table 4 summarizes the advantages and the problems of these methods.

Method	**Advantage**	**Problems**
Classification	Reduction of state space	Large classes lead to inaccurate results
Parzen window approach	Efficient calculation	Parameter h needs to be known in advance
Restriction of initial configurations	Reduction of relevant states	For accurate results many initial configurations are needed

Table 4. Approximation methods for entropy based measures

All approximation methods discussed in this paper are based on time series: By using simulation runs, time series consisting of the configuration sequences can be obtained. Instead of calculating the measures analytically in the model, it is possible to get approximations of the measures directly from the set of time series. Since the model is not needed anymore, this can be generalized to arbitrary time series of configurations: For each set of configuration sequences, the quantitative measure, which were defined analytically in [1], [2] only for the

model, can be approximated by considering only the time series. This allows the usage of experimental data from the real world without the need of the model: By measuring the parameters of interest in the real world system, we get some time series, which can be used for the calculation of the quantitative measures. This fact helps to evaluate existing systems and to compare different systems or different system parameters with respect to self-organizing properties. The results of these evaluations can be used to improve the existing system.

The methods discussed in this paper can be easily applied for other quantitative measures, which have been defined in literature [1] [2] [3]:

- The level of homogeneity [2] is calculated from the entropies of local configurations of each node v, i.e. the information visible at v. Therefore each approximation method for the entropy discussed in Section 4 can be used for an approximation of the level of homogeneity.
- The level of autonomy [1] is calculated from the entropy of the successor configuration at the current point of time. Therefore each approximation method for the entropy discussed in Section 4 can be used for an approximation of the level of autonomy.
- The level of adaptivity [2] is calculated with respect to valuation maps $b_\theta : \mathrm{Conf}^\theta \to [0,1]$, which describe (like for the measure of resilience) which configurations are good and which configurations are bad. As in the case of resilience and target orientation, the mean values of the random variables can be approximated by calculating the arithmetic mean value of the samples received from the simulation run.
- The level of global-state awareness [3] is calculated from the entropy of a random variable describing an equivalence class of initial configurations. Therefore each approximation method for the entropy discussed in Section 4 can be used for an approximation of the level of global-state awareness.

8 Conclusion and future work

In this paper we investigated some approximation methods to be able to calculate quantitative measures in a micro-level model of a complex system. The approximation methods are based on time series which can be received from simulation runs of the system. To analyze the practical usability of the concepts, we applied the methods to a slot synchronization algorithm in wireless sensor networks. The main goal of our contribution is to analyze different approximation methods for different quantitative measures with respect to usefulness for practical applications. The case study investigated in Section 6 shows that the measures, which are not based on entropy but on the mean value of random variables, can be well approximated by the methods discussed in this paper. This fact holds especially for the resilience and target orientation. In the case study, a practical application could for example be the optimization of the system parameters α, β in the synchronization algorithm for adjusting the phase function ϕ. The level of target orientation (and analogously for resilience) can be calculated for different values for these parameters to find the optimal values.

Another practical application can be found in [3]: The quantitative measure for global state awareness is used to optimize the system parameters of an intrusion detection system.

For measures, which are based on entropies of global properties of the system, the large range of values of the random variables might lead to inaccurate results. These problems are not limited to the investigated example, but appear for many other applications. One possible solution for the method of classification is the choice of "good" classes, where the probability distribution inside each class is nearly uniform, which would lead to more accurate results for the entropy. The characterization of such classes and methods for finding them is left for future work.

References

1. R. Holzer, H. de Meer, and C. Bettstetter, "On autonomy and emergence in self-organizing systems," in *IWSOS 2008*, Springer, 2008.
2. R. Holzer and H. de Meer, "Quantitative Modeling of Self-Organizing Properties," in *IWSOS 2009* (T. Spyropoulos and K. A. Hummel, eds.), vol. 5918 of *LNCS*, pp. 149–161, Springer, December 2009.
3. C. Auer, P. Wuechner, and H. de Meer, "The degree of global-state awareness in self-organizing systems," in *IWSOS 2009*, Springer, 2009.
4. 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.
5. 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.
6. C. R. Shalizi, *Causal Architecture, Complexity and Self-Organization in Time Series and Cellular Automata*. PhD thesis, University of Wisconsin-Madison, 2001.
7. G. Nicolis and I. Prigogine, *Self-Organization in Non-Equilibrium Systems: From Dissipative Structures to Order Through Fluctuations*. Wiley, 1977.
8. H. von Foerster, *Self-Organizing Systems*, ch. On Self-Organizing Systems and their Environments, pp. 31–50. Pergamon, 1960.
9. W. R. Ashby, *Principles of Self-organization*, ch. Principles of the Self-organizing System, pp. 255–278. Pergamon, 1962.
10. F. Heylighen and C. Joslyn, "Cybernetics and second order cybernetics," *Encyclopedia of Physical Science & Technology*, vol. 4, pp. 155–170, 2001.
11. H. Haken, *Self-organizing Systems: An Interdisciplinary Approach*, ch. Synergetics and the Problem of Selforganization, pp. 9–13. Campus Verlag, 1981.
12. C. Gershenson, *Design and Control of Self-organizing Systems*. PhD thesis, Vrije Universiteit Brussel, Brussels, Belgium, May 2007.
13. R. Holzer, P. Wuechner, and H. De Meer, "Modeling of self-organizing systems: An overview," *Electronic Communications of the EASST*, vol. 27, pp. 1–12, 2010.
14. 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.

15. R. Holzer and H. de Meer, "On modeling of self-organizing systems," in *Autonomics 2008*, 2008.
16. 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.
17. D. Fisch, M. Jnicke, B. Sick, and C. Müller-Schloer, "Quantitative emergence a refined approach based on divergence measures," in *Fourth IEEE International Conference on Self-Adaptive and Self-Organizing Systems*, (Budapest), 2010.
18. T. M. Cover and J. A. Thomas, *Elements of Information Theory*. Wiley, 2nd ed., 2006.
19. C. M. Bishop, *Pattern Recognition and Machine Learning*. New York: Springer, 2006.
20. C. M. Bishop, "Novelty detection and neural network validation," in *IEEE Proc. Vision, Image Signal Processing*, vol. 141, pp. 217–222, 1994.
21. 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.
22. R. Mirollo and S. Strogatz, "Synchronization of pulse-coupled biological oscillators," *SIAM Journal of Applied Mathematics*, vol. 50, pp. 1645–1662, 1990.