Measuring Self-organisation at Runtime
A Quantification Method based on Divergence Measures

Sven Tomforde[1], Jan Kantert[2] and Bernhard Sick[1]
[1]*Intelligent Embedded Systems, University of Kassel, Wilhelmshöher Allee 73, 34121 Kassel, Germany*
[2]*Institute of Systems Engineering, Leibniz Universiät Hannover, Appelstr. 4, 30169 Hannover, Germany*

Keywords: Self-organisation, Quantification of Self-organisation, Organic Computing, Adaptivity, Probabilistic Models, Communication Patterns.

Abstract: The term “self-organisation” typically refers to the ability of large-scale systems consisting of numerous autonomous agents to establish and maintain their structure as a result of local interaction processes. The motivation to develop systems based on the principle of self-organisation is to counter complexity and to improve desired characteristics, such as robustness and context-adaptivity. In order to come up with a fair comparison between different possible solutions, a prerequisite is that the degree of self-organisation is quantifiable. Even though there are some attempts in literature that try to approach such a measure, there is none that is real-world applicable, covers the entire runtime process of a system, and considers agents as blackboxes (i.e. does not require internals about status or strategies). With this paper, we introduce a concept for such a metric that is based on external observations, neglects the internal behaviour and strategies of autonomous entities, and provides a continuous measure that allows for an easy comparibility.

1 INTRODUCTION

In 1991, Marc Weiser formulated his vision of “ubiquitous computing” (Weiser, 1991) which predicted that individual devices such as personal computers will be replaced by “intelligent things” and these things support humans in an imperceptible manner. A fundamental prerequisite for such a pervasive system serving humans in their daily lives is the ability of distributed technical devices to self-organise. Since a centralised or even human controlled process is not applicable due to the sheer amount of decisions to be taken, the interdependencies among distributed elements, and the resulting complexity. As a consequence, initiatives such as Proactive (Tennenhouse, 2000), Autonomic (Kephart and Chess, 2003), and Organic Computing (Tomforde et al., 2011) or Complex Adaptive Systems (Kernbach et al., 2011) emerged that investigate large-scale self-adaptation and self-organisation processes.

Although there is no commonly agreed definition of the term, the basic concept related to self-organisation typically means that the structure of the overall system is dynamic (i.e., time-variant), and the adaptations causing these dynamic changes are done by the entities forming the system. More precisely, the system consists of autonomous entities and these entities decide with which other entity they interact (e.g., to solve a task or to exchange information). In the context of this paper, we will consider the structure of a system being identical with the relations among distributed entities (i.e., subsystems or *agents* in the terminology of multi-agent systems). We further assume that such a relation has a functional meaning, i.e., it defines an interaction that supports the functionality of the overall system.

The underlying hypothesis of all the aforementioned initiatives is that self-organisation is beneficial in comparison to traditional, centralised solutions; and that this benefit can be expressed in terms of aspects such as higher robustness, higher efficiency, or reduced task complexity, for instance—see e.g. (Müller-Schloer et al., 2011; Wooldridge, 2009). However, some work suggests that neither the fully self-organised nor the fully centralised organisation blueprint will result in the most efficient (or even optimal) behaviour. One particular example from our own preliminary work can be found in establishing progressive signal systems for urban road traffic control, see (Tomforde et al., 2008). Here, nodes (i.e., intersection controllers) coordinate themselves to improve the overall traffic flow. As a consequence, the system

Tomforde S., Kantert J. and Sick B.
Measuring Self-organisation at Runtime - A Quantification Method based on Divergence Measures.
DOI: 10.5220/0006240400960106
In *Proceedings of the 9th International Conference on Agents and Artificial Intelligence (ICAART 2017)*, pages 96-106
ISBN: 978-989-758-219-6

structure in terms of coordination schemes emerges as a result of local interactions—self-organisation takes place. However, a fully self-organised solution is not beneficial in some cases—while, in turn, a fully centralised solution is not fast enough. As a consequence, hybrid approaches have been developed that apply *self-organisation to a certain degree* (Tomforde et al., 2010a).

A meaningful comparison between different solutions to the same problem area is twofold—if done correctly: it founds on desired metrics (e.g., robustness, utility) and accompanied cost (e.g., communication and computation overhead). However, to order these attempts according to the system organisation, a third metric has to be applied: a quantification of the degree of self-organisation. To estimate the degree of self-organisation, the runtime behaviour of a system has to be evaluated—without the need of accessing internal logic and status of autonomous entities: In increasingly open and interwoven system structures consisting of autonomous agents (Tomforde et al., 2014; Hähner et al., 2015), we have to deal with systems that we do not control and that we have not developed—we can just observe their behaviour from the outside. In this paper, we develop a concept for such a metric that is real-world applicable, covers the entire runtime process of a system, and considers agents as blackboxes (i.e., does not require internals about status or strategies), see (Wooldridge, 2009). It can be used to quantify how much self-organisation is observed—in contrast to classifying systems as being either purely self-organised or centralised. We can further use the metric to distinguish between such cases.

The remainder of this paper is organised as follows: Section 2 briefly summarises the state-of-the-art. Afterwards, Section 3 describes the underlying system model, the assumptions and preliminary work. Section 4 introduces the novel approach to quantify self-organisation in technical systems. Section 5 analysis the behaviour of the measurement in terms of an exemplary scenario. Finally, the paper closes with a summary and an outlook in Section 6.

2 STATE OF THE ART

The term "self-organisation" is increasingly used in literature, covering a variety of domains such as mathematics (Lendaris, 1964), thermodynamics (Nicolis and Prigogine, 1977), or information theory (Shalizi, 2001). In addition, non-technical considerations have been discussed, see e.g. (Heylighen, 1999) for an overview. Thereby, the term is typically used to describe effects where a certain structure emerges with a bottom-up perspective—meaning as a result of *autonomous* processes.

In systems engineering, especially in the context of initiatives such as Organic (OC) and Autonomic Computing (AC), methods to transfer classic designers' and administrators' decisions to the responsibility of the systems themselves are investigated. As a result, the relevance of a phenomenon of self-organisation is ubiquitously accepted—but a commonly agreed notion or definition of its characteristics is not existing. Instead, a variety of sometimes contradictory definitions and descriptions are observed.

Many natural and social systems served as inspiration to frame the understanding of the term self-organisation—ranging from work organisation in ant colonies, see (Dorigo and Birattari, 2010) for a technical imitation, to flow formations in pedestrian movements, see (Helbing, 2012). By external observation, humans recognise an increase of order (e.g., in terms of pattern forming)—a behaviour is produced that in some way can be called "organised" (i.e., generating some kind of structure). In addition, Polani refers to this observation as *self-organisation* if the "source of organisation is not explicitly identified outside of the system" (Polani, 2013).

Continuing the previous discussion, Polani defines self-organisation as a "phenomenon under which a dynamical system exhibits the tendency to create organisation out of itself, without being driven by an external system" (Polani, 2013). Unfortunately, this definition—the same holds for others—is pretty close to that of emergence. As a result, many attempts can be found where emergence and self-organisation are compared and distinguished from each other, e.g., in (Shalizi, 2001). However, if no clear notion of both terms is given, a separation is hardly possible.

In the context of OC, Mühl et al. proposed a formal classification of technical systems with the purpose to define a class for "self-organising technical systems" (Muehl et al., 2007). Their classification is founded on Zadeh's notion of "adaptivity" (Zadeh, 1963) and introduces a hierarchical structure: from *self-manageable* at the bottom layer to *self-managing* and to *self-organising*. They consider a system as self-organising "if it is (i) self-managing, i.e., the system adapts to its environment without outside control, (ii) structure-adaptive, i.e., the system establishes and maintains a certain kind of structure (e.g., spatial, temporal) regarding the system's primary functionality, and (iii) employs decentralised control, i.e., the system has no central point of failure" (Muehl et al., 2007). From the perspective of this paper, insisting on a complete absence of external control is not desirable since user influence is part of the overall concept, see (Tomforde et al., 2011) as one example. However, the classification gives a valuable guideline to what we need as basis for defining self-organisation in technical systems.

Besides these conceptual approaches to self-organisation, more formal methods have been proposed that try to come up with a quantification of self-organisation in technical systems. As a result of the heterogeneous origins for working on self-organised systems, a variety of attempts to measure and quantify it have been made. A first example has been introduced by Shalizi et al. (Shalizi and Shalizi, 2003; Shalizi et al., 2004). They presented a mathematical model following Shannon's entropy and defined self-organisation as a process that is characterised by an increase in the amount of information needed to forecast the upcoming system behaviour. The basic idea is that the increase of internal complexity, i.e., without external intervention, relates to the increase of self-organisation. The method works on the basis of observable attributes, such as location (i.e., coordinates) or sensor readings (e.g., temperature or speed).

Similarly, Heylighen et al. (Heylighen, 1999; Heylighen and Joslyn, 2001) presented a concept to use the statistical entropy as basis to determine a so-called "degree of self-organisation". Here, a system that is stuck in an attractor within the state space is defined to be self-organised, since the system cannot reach other states any more. Consequently, a decrease in statistical entropy can be observed which results in an increase of order—that in turn is used as measurement for self-organisation. The main message here is that self-organisation results in stable solutions—external inputs such as disturbances are needed to restart the organisation process again.

Closely related is the concept proposed by Parunack and Brueckner in the context of Multi-Agent Systems (Van Dyke Parunak and Brueckner, 2001). They use an entropy model in combination with the Kugler-Turvey model with the purpose to combine systems with decreasing and increasing order. In their work, a macro- and a micro-level instance of a system are coupled and the information entropy is used to determine the particular degree of self-organisation for each abstraction level. The main insight from this example is that self-organisation affects different abstraction layers.

Furthermore, Wright et al. (Wright et al., 2000; Wright et al., 2001) focussed on an attractor-oriented concept for considering self-organisation. They discussed an approach to measure self-organisation by using entropy as a function of an attractor's dimensionality within the underlying state space. Here, a self-organised system requires an attractor—while the organisation is considered according to Polani's notion (Polani, 2003). Following this approach, the authors define a system to be self-organised if the dynamics of organisation information grow during operation and in correspondence with these dynamics. As one conclusion of the work, we can state that self-organisation has a process perspective and is not just a static characteristic of the system.

Gershenson et al. considered self-organisation as a process opposed to emergence (Gershenson and Fernandez, 2012). Therefore, they measured emergence using Shannon's entropy formula again, and determined self-organisation as decrease of emergence over time. In general, Gershenson claims that in artificial self-organising systems (i.e., engineered systems) "structure and function should emerge from interactions between the elements" (Gershenson and Fernandez, 2012). This implies that the system's purpose is neither designed or programmed, nor controlled. In contrast, components should interact freely and mutually adapt towards a stable solution—also referred to as finding a "preferable" setting through emergence (Gershenson and Heylighen, 2003). In the context of this paper, we focus on purpose-oriented systems engineering—hence, letting the system find "something" is not an option.

OC came up with its own approach to—what they called—"controlled self-organisation", see (Schmeck et al., 2010). They define a corridor for being self-organised by using the maximum (no external intervention for building structural patterns) and minimum (extrinsically organised) boundaries. Following this concept, a quantification of self-organisation is done by considering the control mechanisms being responsible for structure adaptations. For an adaptive system S consisting of 1) m elements (with $m > 1$) that have a large degree of autonomy and 2) k control mechanisms ($k \geq 1$), the degree of self-organisation can be indicated as $(k : m)$.

Summarising the previous discussion of the term "self-organisation" in literature, we can initially state that there are highly heterogeneous notions and understandings of what the term comprises. Most of the discussed statements provide only a very basic approach to understanding when systems can be called self-organised. Also, there is typically either a non-technical perspective applied or the concept lacks a consideration of organisation in the sense of a technical system structure.

3 ASSUMPTIONS AND SYSTEM MODEL

In the following, we define self-organisation as a continuous process to establish, change, and maintain a system structure in terms of relationships between autonomous subsystems (or agents). This process is performed by the participating autonomous agents themselves, and it is utility-driven. This means that varying external (and/or internal) conditions require different system structures (and maybe even compositions in terms of participating subsystems), which can be expressed in relation to a certain system goal. This system goal, in turn, can be expressed by a given utility function U. If the subsystems act without external influences, i.e., their behaviour is controlled only by adapting U, we call these subsystems autonomous—which corresponds to the term *agent* where a computer system acts on behalf of a user (Wooldridge, 2009).

3.1 System Model

From a conceptual point of view, we assume a self-organising system S consisting of a potentially large set of autonomous subsystems $a_i \in A$. Each a_i is equipped with sensors and actuators. Internally, each a_i distinguishes between a productive part (*PS*, responsible for the basic purpose of the system) and a control mechanism (CM, responsible for controlling the behaviour of PS and deciding about relations to other subsystems). This corresponds to the separation of concerns between *System under Observation and Control* (SuOC) and *Observer/Controller* tandem in the terminology of OC, see (Tomforde et al., 2011). Figure 1 illustrates the basic system with its input and output relations. However, this is just used to highlight what we mean by referring to *autonomous* subsystems. In particular, the user guides the behaviour of a_i using U and does not intervene at decision level—actual decisions are taken by CM.

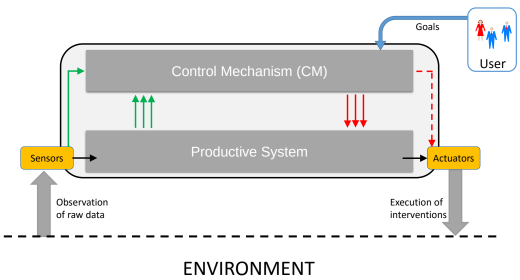

Figure 1: Schematic illustration of an autonomous subsystem.

We do not restrict the system composition, i.e., we allow open systems. However, to define the system boundaries within analysis, we require to deal with a set of agents that potentially participate in the overall system during a certain period of time. Furthermore, we do not require to have full access to each agent a_i: Each agent can belong to different authorities, can be controlled by other users, and can be designed and developed using arbitrary concepts. This includes that we are not aware of the strategies performed by CM, the goal provided by the corresponding user, and the applied techniques, for instance. However, we can observe the external effects: i) the actions that are performed by each a_i and ii) the messages that are sent and received.

Technically, a relationship between two agents represents a functional connection. This means that cooperation is required to solve a certain task: as input-output relations, as (mutual) exchange of information, or as negotiation between a group of a_i, to name just a few. In all these cases, interaction between agents takes place which we map onto the concept of organisation: a_i are connected functionally with each other and these connections are dynamic.

We assume that establishing and changing relations in technical systems requires communication. For simplicity reasons, we model communication as sending and receiving messages via a (shared) communication channel. Conceptually, indirect communication methods such as *stigmergy* (Beckers et al., 1994) can be mapped onto this communication scheme. However, we neglect these cases for the remainder of this paper. Most importantly, we require that the medium is shared and messages are routed using standard communication protocols and infrastructure, e.g., popular mechanisms such as the Transmission Control Protocol / Internet Protocol (IP) tandem, see (Tanenbaum, 2002). Correspondingly, a_i are uniquely identifiable (e.g., via their IP address). Based on this communication model, we assume that each message has an origin and a destination, and there are no fake messages (i.e., from an attacker with a modified origin field). In addition, we require that all messages are visible, and external sources (e.g., from users) can be identified (to neglect those messages as being not relevant for organisation).

3.2 Preliminary Work

In previous work (Kantert et al., 2015), we used a variant of the aforementioned system model. We proposed to measure the degree of organisation based on the structure of communication and especially of agreements in the system. To distinguish static organisation from dynamic self-organisation (i.e., as a process returning the system from a disturbed state into the target space in the sense of OC, see (Schmeck et al., 2010)), we record and quantify changes in the mentioned structure during and after disturbances occur in the system.

In order to estimate the degree of organisation, three different communication graphs $G_i(p) = (V, E_i)$ are built. Each graph is generated for a period p between t_1 and t_2. In all graphs, the set of vertices V represents the same set of agents in the system (i.e., each agent represents an autonomous subsystem). However, the edges E_i describe different kinds of communication processes in the system:

1. A graph G_R contains edges for all possible communication paths between nodes and serves as reference (i.e., it encodes the maximum of edges that are possible).
2. A graph G_C represents the observed communication, i.e., an edge is added if a communication between two nodes has been observed in the current period. The edges E_C in graph G_C define a subset of the edges in G_R, since sending a message requires a communication path.
3. In the third graph G_A, edges represent mutually stable relationships between two agents. In general, relationships are temporary. Such a relationship is the result of, e.g., a negotiation process.

Finally, the measure is defined as:

$$\Theta(G_R, G_Y) := \frac{|\{e_{i,j} : e_{i,j} \in E_1 \oplus e_{i,j} \in E_2\}|}{0.5 * (|V_1| + |V_2|)} \quad (1)$$

with G_Y as either G_C or G_A, E_1 defining the set of edges for the first graph, E_2 the set of edges for the second graph, V_1 and V_2 the corresponding sets of vertices. We applied the concept to a scenario from the smart camera domain, see (Rudolph et al., 2016). Here, we showed that reorganisation effects can be quantified using the developed metric. The graph calculation and comparisons are done in three phases: 1) at system startup, 2) during disturbances, and 3) after recovery from disturbances. This assumes disturbances to be seldom events that can be isolated and detected appropriately fast. However, the approach has some limitations—we will outline them and propose a modified technique in the next section.

4 AN APPROACH TO QUANTIFY SELF-ORGANISATION

In this section, we initially discuss the limitations of the previous approach, present a novel technique to measure self-organisation in technical systems, and discuss some design decisions related to this method. Finally, we outline which challenges have to be addressed to apply the developed concept to real world application.

4.1 Limitations of the Approach

The previously described approach (Section 3) is characterised by some limitations:

1. It needs full knowledge about the underlying *semantics* of the communication model. More precisely, we have to know if the message we observe is related to system organisation aspects or not.
2. The approach assumes minimal communication, i.e., agents do not share redundant information.
3. It assumes stable relationships between distributed entities in the overall system. More precisely, it builds a graph of nodes (reflecting entities) and adds an edge if a relation exists, and this relation is determined as result of an observed communication.

In technical systems considered to be blackboxes (i.e., without access to internals), the required knowledge about semantics may not be available (which is related to the first drawback), individual agents may broadcast data (which is related to the second drawback), or the communication may require a continuous process (which is related to the third drawback). This is accompanied by a possibility to detect mutually stable relationships for G_A. Furthermore, self-organisation may frequently result in changes of the structure. The graph representation is an additional limitation since it loses information: Either there is an edge or not (and these edges are undirected). Self-organisation processes may—in contrast to the assumed handshake model for G_A—require more sophisticated data exchange schemes, more complex decision processes incorporating more than just two partners, and come up with relations that involve more than just two interaction partners. In turn, they may also include directed relationships rather than undirected. These cases are not covered by the approach; the same holds for inherent dynamics (i.e., establishing and closing down relationships more than once in a cycle).

4.2 Modelling Self-organisation

In the following, we outline how these limitations are overcome. We again assume that self-organisation manifests itself by means of relations that are established, updated, and released by communication.

However, we do not require semantics or minimal data load. To detect or measure the self-organised behaviour of a technical system S, we observe the communication behaviour occurring among the autonomous agents a_i

We assume that a system S shows a "normal" communication behaviour. This means that there are no conspicuous communication patterns, and the communication patterns are characterised by low dynamics. As one example, we do not assume a minimalistic communication approach—but we assume that for a certain context the same (or at least a very similar) communication pattern will be used in all cases. Expressed in a probabilistic framework, we represent each subsystem a_i as a process that generates observable samples (i.e., messages). When measuring self-organisation (e.g., by means of sensors monitoring the communication channels), we have to use communication-specific pre-processing techniques to extract the values of attributes (*features*) from those samples (*observations*). These attribute values describe the current behaviour of the observed system. For a standard message, at least origin, destination, packet size, and time stamp will be available in the header information of the package sent over the channel. Based on these perceived attributes, we model the attribute space by a variable $\mathbf{x}$ in the following (or x in the case of an one-dimensional attribute space). In general, $\mathbf{x}$ may consist of categorical and continuous attributes, for instance. Please keep in mind that each message observed within a monitoring period is one sample in our model.

The basic model of self-organisation suggested in this paper is that the dynamics of the communication patterns reflect the dynamics of the self-organisation processes. If no self-organisation takes place (i.e., the structure of S is static) no communication for organisation purposes is required. There may be other communication (i.e., messages) within the system, but none related to structure adaptations. In response to disturbances, changing external or internal conditions, or modifications of the utility function (i.e., user-triggered or as a function of time), the system structure may have to be adapted as well. This is assumed to manifest in a change of the communication pattern: Subsystems may start to communicate with other systems, stop communicating with current partners, or may change the message frequencies, to name just a few implications. However, the basic point is that something is different, without the need of knowing what is different at a semantic level.

Based on this idea, we define self-organisation as an unexpected or unpredictable change of the distribution underlying the observed samples (i.e., the communication behaviour). Consequently, a divergence measure can be applied to compare two density functions. We will refer to a density function $p(x)$ representing an earlier point in time and to $q(x)$ as a density function representing the current observation cycle. A famous divergence measure is the Kullback-Leibler (KL) divergence $KL(p||q)$, see (Bishop, 2011). It is defined for continuous variables as follows:

$$KL(p||q) = -\int p(x) ln \frac{q(x)}{p(x)} dx \tag{2}$$

KL is sometimes referred to as *relative entropy*; however, there is also a discrete version of the measure, see (Shannon, 2001). KL is known to be applicable only in a restricted manner, since it is not symmetric. If needed we can provide a symmetric measure using:

$$KL_2(p,q) = \frac{1}{2}(KL(p||q) + KL(q||p)) \tag{3}$$

Although KL is still limited, it fulfils some important requirements: 1) if $p(x) = q(x)$ the measure $KL(p||q)$ is 0, and 2) $KL(p||q) \geq 0$. Changing the formulation of Equation 2 demonstrates the desired result:

$$KL(p||q) = -\int p(x)\ ln\, q(x) dx + -\int p(x)\ ln\, p(x) dx \tag{4}$$

This formula describes that we measure the expected amount of information contained in a new distribution with respect to a reference distribution of samples. Taking the symmetric concept as defined by Equation 3 into account, we can adapt Equation 4 as follows:

$$\begin{aligned} KL_2(p,q) &= \frac{1}{2}(KL(p||q) + KL(q||p)) \\ &= \frac{1}{2}(-\int p(x)\ ln\, p(x) dx - \int p(x)\ ln\, q(x) dx \\ &\quad + \int q(x)\ ln\, q(x) dx - \int q(x)\ ln\, p(x) dx) \end{aligned} \tag{5}$$

This formula (i.e., Equation 5) can be used as measure for quantifying self-organisation processes. Due to the basic approach to compare distributions of the underlying sample set (or more precisely: of the distribution of densities of observed samples within the input space during a certain observation period), the measure increases if the two distributions begin to change. Considering the basic assumption we made at the begin of this subsection, this exactly models what we expect to observe if self-organisation takes place.

The more subsystems a_i participate in the structure building process, the higher is the divergence to the previous distribution—and consequently, the higher is the measured self-organisation. Systems with hierarchical elements will be characterised by different communication pattern, resulting in a decreased degree of self-organisation since external-oriented (i.e., towards central components) messages are neglected. In case of fully centralised system structures, no self-organisation will be indicated: Messages towards/from external units (i.e., the centralised components) are neglected, and the communication patterns among CM do not change (i.e., only 'normal' behaviour in terms of "hello" messages, for instance).

The most important advantages of this approach are as follows: 1) Compared to, e.g., (Schmeck et al., 2010) it does not require internal information (such as the number of CM). 2) As an alternative to, e.g., (Muehl et al., 2007) it is continuously quantifiable. 3) In comparison to, e.g., (Gershenson and Fernandez, 2012) it is independent of emergence and what is understood to be emergent behaviour. 4) In contrast to, e.g., (Kantert et al., 2015) it is applied continuously and not just for disturbed periods. 5) In contrast to most of the concepts from Section 2, a general model of how the system works is not necessary—it is applicable with low effort. 6) In contrast to, e.g., (Shalizi et al., 2004) it takes only attributes that are relevant for the structure of technical systems into account. Finally, it can easily incorporate system boundaries by specifying communication addresses.

4.3 Observation Cycles

The process as outlined before requires an inherent comparability of two probability distributions. Transferred to the temporal behaviour of a self-organising system, this implies that the potential self-organising process manifests itself in the difference between a current and a referential distribution of attribute occurrences. Expressed in the probabilistic approach as outlined before, this means that we observe a number of processes that "generate" samples resulting in probability distributions. For a comparison, we have to define that the sample period is equal, i.e., we allow the same time period for the current observations as for a reference period. This can be done using a sliding window approach: A fixed time period d is used to observe samples for the current estimation process (i.e., between time t_0 and t_{-1}) and the same duration is used for a reference observation (i.e., the time period directly before the current observations are done: between t_{-1} and t_{-2}). Alternatively, the reference window might be fixed (i.e., static), e.g., at the begin of the observation (here, slow changes can be detected easier, but oscillating behaviour may be harder to detect). Figure 2 illustrates both approaches. However, it may be beneficial to use a hybrid approach that combines both concepts: estimating the change compared to the previous period and against a static distribution to be able to cover all aspects.

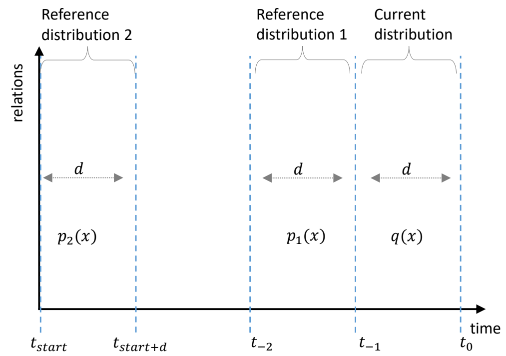

Figure 2: Two possibilities for choosing observation and reference window: sliding and static window approach.

The duration of these windows has to be chosen depending on the underlying application, i.e., the size must be long enough to reliably estimate the probability distributions and simultaneously short enough to be able to consider the observations as being (almost) time-invariant. However, there is no standard answer of how to configure the window size. A good estimate is the online self-adaptation cycle (i.e., the frequency at which CM observes the system status and takes decisions)—for instance, an organic system runs a feedback loop defined by the Observer/Controller tandem in cycles of fixed length to analyse the utility of the currently applied action and decides about necessary interventions accordingly (Tomforde et al., 2011). Taking a period of k cycles into account may serve as a first starting point.

4.4 Open Challenges

This approach to quantifying self-organisation describes the general processes and the basic idea of how self-organisation can be accessed. However, it still comes with some limitations and open questions that require research effort in the near future. From a probabilistic point of view, we proposed to use KL as divergence measure. This measure may take arbitrarily high (positive) values. It would certainly be more convenient to have a measure that is normalised, e.g., restricted to the unit interval $[0;1]$. There are several other divergence measures and we need to investigate if one fits better than KL.

As outlined before, the parameter d defines the length of the window used to observe samples. Although choosing the length of d will depend on the underlying application, the utilised communication infrastructure, and the resulting communication patterns, it may be beneficial to search for a heuristic to chose d. We already mentioned the adaptation cycle of CM as a possible reference point.

Another issue is as follows: If all subsystems come together in discrete cycles and decide about the system structure in a broadcast-based all-to-all approach, our method would come up with no meaningful results, since the distributions will be almost identical. However, we can detect such a peak behaviour in the observation stream (even together with the underlying period) and, consequently, identify a system as a possible candidate for such market-based self-organisation mechanisms.

5 EXAMPLE SCENARIO

To illustrate the presented technique, we outline a specific example application from the vehicular traffic control domain. However, the developed approach is not restricted to a single domain. For instance, self-adapting data communication networks are an obvious next step to analyse the behaviour of the measurement, e.g., in the context of the system presented in (Tomforde et al., 2009; Tomforde et al., 2010b). Another possible application scenario to analyse the behaviour in future work is a self-adapting activity recognition system that dynamically includes sensors from the environment and other devices as, e.g., outlined in (Jänicke et al., 2016). Here, the degree of self-organisation can be used to quantify to which extent the system made use of its ability to include new sensors or deactivate previously used sensors while trying to maximise the recognition rate.

5.1 Scenario

Due to the dynamics of traffic, adapting traffic control strategies to changing conditions is a promising application area for self-adaptive and self-organising systems (Prothmann et al., 2011). Besides changing durations of green phases, autonomous self-organisation comes into play if intersection controllers (i.e., a CM in the notion of Section 3.1) are responsible for establishing and maintaining progressive signal systems (PSS)—also called “green waves”. In (Tomforde et al., 2008), a distributed PSS mechanism (DPSS) for urban road networks is presented (Tomforde et al., 2008). The approach is a three step process and works as follows: 1) Initially, distributed CM determine partners that collaborate to form a PSS, 2) after establishing partnerships, the collaborating CM agree on a common cycle time, and 3) the partners select signal plans that respect the common cycle time, calculate offsets, and establish a coordinated signalisation. We analyse this example in the remainder to highlight the behaviour of the proposed measure.

For the first step of DPSS, each CM estimates (based on local sensor data) which is the most prominent stream running over the controlled intersection, i.e., the stream with the currently highest number of $\frac{vehicles}{hour}$. Afterwards, it sends a request for partnership to the upstream CM—which is the desired predecessor in a PSS. Either that CM accepts partnerships or it rejects. In case of rejection, the second best neighbour is asked. For the second step, an echo algorithm is used that starts at the first CM of a PSS. It estimates the locally desired values for the cycle time and sends this to the successor CM. All subsequent CM calculate the maximum of this received value and their desired cycle time and pass it to the next CM until the last intersection of the PSS is reached. This last CM propagates the chosen cycle time back to all CM in reverse direction. Finally, the first CM selects the most beneficial signal plan (i.e., defining green durations at the underlying intersection) that reflects the determined cycle time, calculates an offset (i.e., a relative start within the cycle) and passes this information to its successor. This is continued until the last intersection is reached. When all CM activate their chosen signal plan, the PSS is established. Afterwards, each node continuously monitors if the traffic behaviour still corresponds to the desired PSS—and starts an update or re-negotiation process if the preferences regarding traffic streams or desired cycle time change, or if a partner node is not available any more.

In this example, the organisation of the system manifests itself in terms of functional relations by means of forming a PSS, i.e., predecessor-successor relationships. The system is self-organised with respect to the definition in Section 3 since these relations are established and updated by the autonomous CM participating in the system.

5.2 Analysis of the Behaviour

In this traffic scenario, we assume each intersection controller (i.e., each CM) to continuously send “hello”-messages to verify the availability of neighbours—since a neighbour that is not reachable via communication cannot participate in a PSS. In addition, the process itself requires communication effort in all three phases. This process runs continuously: At system startup, initial coordination schemes may be generated, and they are continuously monitored and adapted throughout the system's operation. However, we observe periods of low communication effort (i.e., only "hello"-messages are exchanged) where no self-organisation takes place. In this case, the structure is static.

Traffic load in urban areas changes during the course of a day. As a consequence, the load on the streams varies. Consider commuter traffic as an illustrating example: During the morning rush hours, the highest traffic load moves towards the city centre, while the reverse direction is favoured in the afternoon. Consequently, the best possible PSS switches during lunch time from inwards to outwards. As a conclusion from this simple example, we can observe that the most prominent streams will change throughout the day. As a consequence, the CM will eventually activate the DPSS mechanism with all three phases, resulting in frequent message exchange. In addition, this process is not necessarily a disturbance as assumed in previous work. In contrast, it will most probably happen frequently. However, in case of disturbances (i.e., blocked roads and a resulting drop in traffic load for streams running over that road, or failures of neighboured CM) we will observe communication behaviour as well. The proposed measure is able to detect these phases of reorganisation and consequently quantifies the degree to which self-organisation takes place in this example system.

As a basis of comparison, other concepts for PSS can be compared to the DPSS approach using the proposed metric. Systems with hierarchical elements will show a different communication pattern, resulting in a decreased degree of self-organisation since external-oriented (i.e., towards central components) messages are neglected. In case of fully centralised system structures, no self-organisation will be indicated, since no messages among CM are exchanges (besides "hello" messages, for instance).

6 CONCLUSION

In this paper, we defined what self-organisation means in technical systems from our point of view and proposed a technique to measure and quantify it at runtime. Our basic system model assumes autonomous agents forming the overall system—which means that we do not have any insights about the status, the strategies, and the goals of an individual agent. Consequently, we propose to base a measure on the only information that is available for external observers: the communication in terms of messages. Compared to previous work, we do not make any assumptions regarding the purpose and semantics of communication. In contrast, we define a probabilistic model to estimate distributions of samples (i.e., messages) within a fixed time period. We then compare the current distribution against a reference distribution. The more these distributions differ, the more self-organisation takes place. We further discussed how the sampling period to derive these distributions has to be chosen and how a comparison can be calculated. For illustration purposes, we briefly discussed a scenario from urban traffic control, i.e., establishing progressive signal systems, and explained the expected behaviour of the proposed measure.

As part of this paper, we already discussed current challenges to be addressed in Section 4.4. Our future work will investigate how possible solutions can be developed. For instance, we proposed to make use of the Kullback-Leibler divergence to compare different distributions. However, there are various divergence measures known in literature and we have to investigate which performs best, i.e., quantifies the effect of self-organisation as close to human recognition as possible.

Furthermore, the concept is based on the observation of communication—and the possibility to neglect external messages (i.e., with origin or destination that are not part of S). The developed measure may also be used to relate it to the subsequent reorganisation process. If knowing about external influences and observing reorganisation afterwards, it may serve as an indicator that the system is less autonomous or that the user triggered a change of utility. Consequently, it may be extended towards estimating the autonomy of the system at runtime. Given a long enough observation period, we may also be able to learn what the most appropriate system structure is for a given context (and a given utility function).

Finally, we aim at analysing the behaviour of the developed measurement in simulations of self-organised systems. Initially, we consider traffic control as outlined before. Afterwards, we aim at increasing the scope towards data communication and activity recognition systems.

ACKNOWLEDGEMENTS

The authors would like to thank the German research foundation (Deutsche Forschungsgemeinschaft, DFG) for the financial support in the context of the "Organic Computing Techniques for Runtime Self-Adaptation of Multi-Modal Activity Recognition Systems" project (SI 674/12-1).

REFERENCES

Beckers, R., Holland, O. E., and Deneubourg, J. (1994). From local actions to global tasks: Stigmergy and collective robotics. *Artificial Life IV*, pages 181 – 189.

Bishop, C. M. (2011). *Pattern Recognition and Machine Learning*. Information Science and Statistics. Springer, 2nd edition. ISBN 978-0387310732.

Dorigo, M. and Birattari, M. (2010). Ant Colony Optimization. In Sammut, C. and Webb, G., editors, *Encyclopedia of Machine Learning*, pages 36–39. Springer US.

Gershenson, C. and Fernandez, N. (2012). Complexity and Information: Measuring Emergence, Self-organization, and Homeostasis at Multiple Scales. *Complexity*, 18(2):29–44.

Gershenson, C. and Heylighen, F. (2003). The Meaning of Self-Organisation in Computing. *IEEE Intelligent Systems*, pages 72 – 75.

Hähner, J., Brinkschulte, U., Lukowicz, P., Mostaghim, S., Sick, B., and Tomforde, S. (2015). Runtime Self-Integration as Key Challenge for Mastering Interwoven Systems. In *Proceedings of the 28th GI/ITG International Conference on Architecture of Computing Systems – ARCS Workshops, held 24 – 27 March 2015 in Porto, Portugal, Workshop on Self-Optimisation in Organic and Autonomic Computing Systems (SAOS15)*, pages 1 – 8. VDE Verlag.

Helbing, D. (2012). *Social Self-Organization: Agent-Based Simulations and Experiments to Study Emergent Social Behavior*. Springer. ISBN-13: 978-3642240034.

Heylighen, F. (1999). The Science Of Self-Organization And Adaptivity. In *Knowledge Management, Organizational Intelligence and Learning, and Complexity, in: The Encyclopedia of Life Support Systems, EOLSS*, pages 253–280. Publishers Co. Ltd.

Heylighen, F. and Joslyn, C. (2001). Cybernetics and Second-Order Cybernetics. In Meyers, R. A., editor, *Encyclopedia of Physical Science and Technology*. Acadmeic Press, New York, US, 3rd edition.

Jänicke, M., Tomforde, S., and Sick, B. (2016). Towards Self-Improving Activity Recognition Systems Based on Probabilistic, Generative Models. In *2016 IEEE International Conference on Autonomic Computing, ICAC 2016, Wuerzburg, Germany, July 17-22, 2016*, pages 285–291.

Kantert, J., Tomforde, S., and Müller-Schloer, C. (2015). Measuring Self-Organisation in Distributed Systems by External Observation. In *Proceedings of the 28th GI/ITG International Conference on Architecture of Computing Systems – ARCS Workshops*, pages 1 – 8. VDE. ISBN 978-3-8007-3657-7.

Kephart, J. and Chess, D. (2003). The Vision of Autonomic Computing. *IEEE Computer*, 36(1):41–50.

Kernbach, S., Schmickl, T., and Timmis, J. (2011). Collective adaptive systems: Challenges beyond evolvability. *ACM Computing Research Repository (CoRR)*. last access: 07/14/2014.

Lendaris, G. G. (1964). On the Definition of Self-organising Systems. *Proceedings of the IEEE*, 52(3):324–325.

Muehl, G., Werner, M., Jaeger, M., Herrmann, K., and Parzyjegla, H. (2007). On the Definitions of Self-Managing and Self-Organizing Systems. In *Communication in Distributed Systems (KiVS), 2007 ITG-GI Conference, Kommunikation in Verteilten Systemen, 26. Februar - 2. März 2007 in Bern, Schweiz*, pages 1–11.

Müller-Schloer, C., Schmeck, H., and Ungerer, T., editors (2011). *Organic Computing - A Paradigm Shift for Complex Systems*. Autonomic Systems. Birkhäuser Verlag, Basel, CH.

Nicolis, G. and Prigogine, I. (1977). *Self-Organisation in Nonequilibrium Systems: From Dissipative Structures to Order through Fluctuations*. Wiley, 1st edition.

Polani, D. (2003). Measuring Self-Organization via Observers. In Banzhaf, W., Ziegler, J., Christaller, T., Dittrich, P., and Kim, J., editors, *Advances in Artificial Life*, volume 2801 of *Lecture Notes in Computer Science*, pages 667–675. Springer Verlag, Berlin Heidelberg, Germany.

Polani, D. (2013). Foundations and Formalisations of Self-Organisation. In Prokopenko, M., editor, *Advances in Applied Self-Organizing Systems*, chapter 2, pages 23 – 43. Springer Verlag.

Prothmann, H., Tomforde, S., Branke, J., Hähner, J., Müller-Schloer, C., and Schmeck, H. (2011). Organic Traffic Control. In *Organic Computing – A Paradigm Shift for Complex Systems*, Autonomic Systems, pages 431 – 446. Birkhäuser Verlag, Basel, CH.

Rudolph, S., Kantert, J., Jänen, U., Tomforde, S., Hähner, J., and Müller-Schloer, C. (2016). Measuring Self-Organisation Processes in Smart Camera Networks. In Varbanescu, A. L., editor, *Proceedings of the 29th International Conference on Architecture of Computing Systems (ARCS 2016)*, chapter 14, pages 1–6. VDE Verlag GmbH, Berlin, Offenbach, DE, Nuremberg, Germany.

Schmeck, H., Müller-Schloer, C., Çakar, E., Mnif, M., and Richter, U. (2010). Adaptivity and Self-organisation in Organic Computing Systems. *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, 5(3):1–32.

Shalizi, C., Shalizi, K., and Haslinger, R. (2004). Quantifying Self-Organization with Optimal Predictors. *Phys. Rev. Lett.*, 93:118701.

Shalizi, C. R. (2001). *Causal Architecture, Complexity and Self-organization in Time Series and Cellular Automata*. PhD thesis, The University of Wisconsin - Madison.

Shalizi, C. R. and Shalizi, K. L. (2003). Quantifying Self-Organization in Cyclic Cellular Automata. In Schimansky-Geier, L., Abbott, D., Neiman, A., and den Broeck, C. V., editors, *Noise in Complex Systems and Stochastic Dynamics, Proceedings of SPIE*, volume 5114, pages 108 – 117. Bellingham, Washington.

Shannon, C. E. (2001). A Mathematical Theory of Communication. *ACM SIGMOBILE Mobile Computing and Communications Review*, 5(1):3–55.

Tanenbaum, A. S. (2002). *Computer Networks*. Pearson Education, 4th edition.

Tennenhouse, D. (2000). Proactive Computing. *Communications of the ACM*, 43(5):43–50.

Tomforde, S., Cakar, E., and Hähner, J. (2009). Dynamic Control of Network Protocols - A new vision for future self-organised networks. In Filipe, J., Cetto, J. A., and Ferrier, J.-L., editors, *Proceedings of the 6th International Conference on Informatics in Control, Automation, and Robotics (ICINCO'09), held in Milan, Italy (2 - 5 July, 2009)*, pages 285 – 290, Milan. INSTICC.

Tomforde, S., Hähner, J., and Sick, B. (2014). Interwoven Systems. *Informatik-Spektrum*, 37(5):483–487. Aktuelles Schlagwort.

Tomforde, S., Prothmann, H., Branke, J., Hähner, J., Mnif, M., Müller-Schloer, C., Richter, U., and Schmeck, H. (2011). Observation and Control of Organic Systems. In Müller-Schloer, C., Schmeck, H., and Ungerer, T., editors, *Organic Computing - A Paradigm Shift for Complex Systems*, Autonomic Systems, pages 325 – 338. Birkhäuser Verlag.

Tomforde, S., Prothmann, H., Branke, J., Hähner, J., Müller-Schloer, C., and Schmeck, H. (2010a). Possibilities and Limitations of Decentralised Traffic Control Systems. In *2010 IEEE World Congress on Computational Intelligence (IEEE WCCI 2010), held 18 Jul - 23 Jul 2010 in Narcelona, Spain*, pages 3298–3306. IEEE.

Tomforde, S., Prothmann, H., Rochner, F., Branke, J., Hähner, J., Müller-Schloer, C., and Schmeck, H. (2008). Decentralised Progressive Signal Systems for Organic Traffic Control. In Brueckner, S., Robertson, P., and Bellur, U., editors, *Proceedings of the 2nd IEEE International Conference on Self-Adaption and Self-Organization (SASO'08), held in Venice, Italy (October 20 - 24, 2008)*, pages 413–422. IEEE.

Tomforde, S., Zgeras, I., Hähner, J., and Müller-Schloer, C. (2010b). Adaptive control of Wireless Sensor Networks. In *Proceedings of the 7th International Conference on Autonomic and Trusted Computing (ATC'10), held in Xi'an, China (October 26-29, 2010)*, pages 77 – 91.

Van Dyke Parunak, H. and Brueckner, S. (2001). Entropy and Self-organization in Multi-agent Systems. In *Proceedings of the Fifth International Conference on Autonomous Agents, AGENTS'01 Autonomous Agents 2001, Montreal, QC, Canada - May 28 - June 01, 2001*, AGENTS '01, pages 124–130, New York, NY, USA. ACM.

Weiser, M. (1991). The computer for the 21st century. *Scientific American*, 265(3):66–75.

Wooldridge, M. J. (2009). *An Introduction to MultiAgent Systems*. John Wiley & Sons Publishers, Hoboken, NJ, US, 2nd edition.

Wright, W., Smith, R. E., Danek, M., and Greenway, P. (2000). A Measure of Emergence in an Adapting, Multi-Agent Context. In *From Animals to Animats 6: Proceedings of the 6th Int. Conf. on Simulation of Adaptive Behavior*, pages 20 – 27. Springer.

Wright, W., Smith, R. E., Danek, M., and Greenway, P. (2001). A Generalisable Measure of Self-Organisation and Emergence. In *Proc. of the Int. Conf. on Artificial Neural Networks (ICANN'01)*, volume 2130 of *LNCS*, pages 857 – 864. Springer.

Zadeh, L. A. (1963). On the Definition of Adaptivity. *Proceedings of the IEEE*, 3(1):470 – 496.