15. Characterization of Self-Organization

Hermann De Meer, Christian Koppen (University of Passau)

15.1 Introduction

Self-organization is used in many disciplines to refer to several, related phenomenons. Some of the more prominent phenomenons summarized under the umbrella of self-organization are autonomy, self-maintenance, optimization, adaptivity, rearrangement, reproduction or emergence. An exact match, however, has yet to be accomplished. Even in the context of this book on Peer-to-Peer systems, self-organization is used in various forms to relate to several interesting but distinct properties of Peer-to-Peer networking. Before Peer-to-Peer networks are analyzed in more detail in Chapter 16 for their degree of affinity to self-organization, we juxtapose selected but prominent definitions and criteria of self-organization from all disciplines in this chapter. The purpose of that exercise is to broaden scope and horizon of understanding self-organization in the context of Peer-to-Peer networks. It is hoped such an approach may spearhead new developments and stimulate innovative discussions. Due to the nature of some of the disciplines, the definitions may lack mathematical preciseness and some ambiguities may not be overcome. It is still believed by comparing and relating the existing manifold perspectives and concepts a more objective and thought-provoking discussion in the context of Peer-to-Peer networking can result. This is particularly so as self-organization may offer great potentials and pose high risks by the same token.

The notion of self-organization is not a new one. In fact, its roots may even be traced back to ancient times. It was Aristoteles who stated that "The whole is more than the sum of its parts" [32, 10f-1045a], a simple definition for a phenomenon that nowadays is called *emergence* which is attributed to self-organizing systems. In the 20th century, the pioneer discipline engaging in self-organization was the science of cybernetics, originated as theory of communication and control of regulatory feedback in the 1940s; cyberneticists study the fundamentals of organizational forms of machines and human beings. The name and concept of self-organization as it is understood today emerged in the 1960s, when related principles were detected in different scientific disciplines.

The biologists Varela and Maturana formed the term of *autopoiesis*, a way of organization that every *living* organism seems to exhibit [402, 403]. The chemist Ilya Prigogine observed the formation of order in a special class of chemical system which he then called *dissipative* [495]. The biochemists Eigen and Schuster detected *autocatalytic hypercycles*, a form of chemical

R. Steinmetz and K. Wehrle (Eds.): P2P Systems and Applications, LNCS 3485, pp. 227-246, 2005.

molecules which align with each other and reproduce themselves to build up and maintain a stable structure [551]. The physicist Haken analyzed the *laser* and found that the atoms and molecules organize themselves so that a homogeneous ray of light is generated [271]. The field of *synergetics* resulted as a whole new discipline from this research [272]. These are just some examples, a lot more have appeared in natural, social, economic and information sciences.

Each approach has revealed some basic principles of self-organization, and it is interesting to see how similarities can be identified. Self-organization may appear in different facets or it can be seen as an assembly of several, many or all of the described properties. We first describe the various properties in more detail in Section 15.2, use them to characterize self-organization in Section 15.3 and then apply the results to examples in computer science, emphasizing of what we see as positive impact of self-organization on these example areas, in Section 15.4. Section 15.5 concludes this chapter.

15.2 Basic Definitions

Self-organization is used in many disciplines to refer to several, related phenomenons. Selected but prominent definitions and criteria of self-organization from all disciplines are summarized in this section. The ultimate purpose is to broaden scope and horizon of understanding self-organization in the context of Peer-to-Peer networks. It is hoped such an approach may spearhead new developments and stimulate innovative discussions. Due to the nature of some of the disciplines, some of the definitions of this section may lack mathematical preciseness.

15.2.1 System

Definition: **System**

A system is a set of components *that have* relations *between each other and form a unified whole. A system distinguishes itself from its environment.*

A simple example of a system is a computer network where computers are the components and connections between the computers are the relations. The network can be seen as a single entity (instead of an accumulation of computers) which differs from its environment, e.g., the users of the network.

Another example for a system is a cup of water. It consists of the H_2O-molecules and the intermolecular forces between them. The environment is, strictly speaking, the ceramics of the cup and the rest of the universe. In most cases, it is sufficient to consider only the relevant parts of the environment, such as the cup in this example. The system is not observed as a giant set of single molecules, but as one big entity.

15.2.2 Complexity

The term "complexity" is used to denote diverse concepts coming from domains which exhibit strong differences, like social [57], economic [120] and computer sciences [628]. Even the *theory of complexity* is used to denominate multiple disciplines, including theoretical computer science, systems theory and chaos theory. For example, in theoretical computer science the Landau symbols (e.g., $\mathcal{O}(n), \omega(n), \Theta(n)$) are used to describe the time or space complexity of an algorithm, independently of a certain implementation. The Kolmogorow complexity, the degree of order in a string, is determined by the size of the *shortest* computer program that creates this string. A general definition of complexity is therefore hard to achieve.

> Definition: **Complexity**
>
> *We use the term complexity to denote the existence of system properties that make it difficult to describe the semantics of a system's overall behavior in an arbitrary language, even if complete information about its components and interactions is known. [58]*

We differentiate this meaning from another meaning of the term complexity in the sense of "complicatedness" which describes the number of elements or components of a system [623, Komplexität (orig. in German)].

Complexity as used here is independent of the language that is used for the system description. There are many different perspectives on a system, which lead to different descriptions, although the system remains the same for all descriptions. A cup of water can be described by some *formulae* for the H_2O molecules, the water's volume and temperature. It may as well be described using a large *table* that contains position, size and speed of each molecule. The formulae are a more compact description than the table, but the complexity of the system stays the same for both descriptions.

A system does not need to keep its complexity over time. If it changes so that its behavior can be described more compactly, the complexity of the system is reduced. An example for this are the Bénard convection cells, which are best explained by a small experiment [440]. Consider a closed and completely filled pot of water. It can be easily described because the molecules don't move. Now consider it being heated from the bottom and cooling down from the top. The molecules first move up and then back down because they cool down. More information (the effecting powers) is needed to describe the system – its complexity is increased. If the temperature difference is increased further the water flows suddenly in form of big rolls which are called *convection cells* (Figure 15.1). The water moves up on one side and down on the other side, in a regular movement. The more regular movement can now be easier described by a formula due to a reduced complexity.

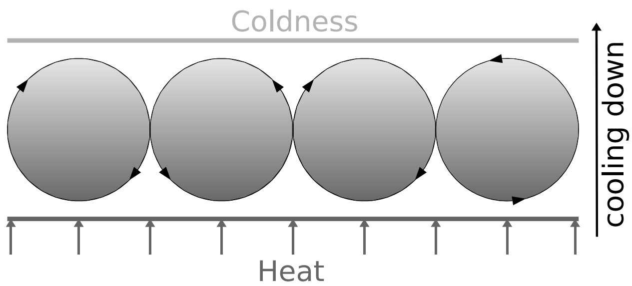

Fig. 15.1: Bénard convection cells.

15.2.3 Feedback

> Definition: **Feedback**
>
> *We use the term "feedback" to describe "the return to the input of a part of the output of a machine, system, or process (as for producing changes in an electronic circuit that improve performance or in an automatic control device that provide self-corrective action)" [409].*

Feedback can lead to effects that do not proportionally depend on the causes [246]. Feedback allows the amplification or attenuation of external influences within a system. An amplification is due to *positive feedback* and an attenuation is due to *negative feedback*.

15.2.4 Emergence

The term "emergence" is used in various disciplines [190] and there is by far no general agreement about its meaning: "First, it is often applied to situations, agent behaviors, that are surprising and not fully understood. Second, it refers to a property of a system that is not contained in any one of its parts. This is the typical usage in the fields of artificial life, dynamical systems, and neural networks for phenomena of self-organization. Third, it concerns behavior resulting from the agent-environment interaction whenever the behavior is not preprogrammed" [488]. The term "emergence" is also used to describe

something *new* or *unknown* – which heavily depends on the current knowledge of the observer [194, p. 4]. We do not pursue further such a view.
We use the following definition of "emergence" for our studies:

> Definition: **Emergence**
>
> *"Emergence refers to unexpected global system properties, not present in any of the individual subsystems, that emerge from component interactions." [104]*

Emergent properties are influenced by, but cannot completely be inferred from, characteristics of the components. Interactions between the components are necessary for emergence [300]. An example for emergence is an ant colony where no central control instance exists that decides about the ants' behavior. Each ant reacts to local stimuli (e.g., in form of chemical substances or contact with other ants) but the combination of ants and their behavior form a working colony which is therefore emergent.
Another example of emergence are deadlocks. When multiple CPUs operate without being affected by each other, there is no reason for a deadlock. But as soon as the CPUs are connected so that they become dependent on each other, deadlocks may occur.

Emergence can often be explained by the *process* of the components' interactions, but not just by the character or count of the components. An important detail is that the emergent structure or property can influence the components retroactively, which is a form of feedback. The rolls in the Bénard system described above are an example for such an emergent structure. Once formed by the water molecules, the (macroscopic) structure of a roll influences the movement of the (microscopic) molecules.

15.2.5 Complex System

> Definition: **Complex system**
>
> *"Complex systems are systems with multiple interacting components whose behavior cannot be simply inferred from the behavior of the components."* [552]

If definitions 15.2.1 and 15.2.2 were combined, a complex system would be a system with a description that needs a lot of information, or in other words, whose behavior cannot be described in a compact manner. "What distinguishes a complex system from a merely complicated one is that in a complex system, some behaviors and patterns emerge as a result of the patterns of relationship between the elements." [623, Emergence]. So *emergence* is a necessary property of complex systems.

15.2.6 Criticality

The term "criticality" is used in many domains but has acquired special importance in the field of thermodynamics. Criticality "is used in connection with phase transitions. When the temperature of the system is precisely equal to the transition temperature, something extraordinary happens. [...] The system becomes critical in the sense that all members of the system influence each other." [323] Since this definition is not general enough to be valid in the context of self-organization, we denote a group of system components an "assembly" and use the following definition:

> Definition: **Criticality**
>
> *"An assembly in which a chain reaction is possible is called critical, and is said to have obtained criticality."[623, Criticality]*

Per Bak identifies a relationship between criticality and order. He uses the term to describe a critical point between order and disorder in complex systems [45]. The degree of order of a complex system can reach from total order to pure disorder. In total order, all relations are structured homogeneously and are stable, no unpredictable behavior can be detected in this case. The water molecules in an ice crystal are an example for this. Due to the minimal amount of energy in the system, every component has a stable position that will not change without external perturbation. In pure disorder, it's much harder to find a rule for the behavior of the components since no persistent stability can be observed. Systems in this state are usually analyzed with stochastic methods. For example, the molecules of water vapor move very fast and collide often, which makes it hard to anticipate their position and movement in future states. States that show little order are also called *subcritical*, those which are largely structured are called *supercritical*.

On the border between these two states lies criticality, which has a big impact on the stability of a system. Under certain circumstances, local perturbations may propagate so that all or most components may eventually get perturbed (if the relations are not static and able to propagate the influence). A chain reaction could easily result so that stability could be lost altogether. Whereas in a disordered system, the influence of a single component is most likely to be absorbed without a big impact on the overall system. Systems that reside in the state of criticality are basically stable, at the same time having the ability to change by keeping perturbations locally [647].

A simple example for a system in a critical state is the well-known Abelian Sandpile Model, first proposed in [47]. In this model, single grains of sand are repeatedly dropped onto a random field of a $n \times n$-grid. If the height of a field exceeds a critical value (e.g., 4), the grains of this field topple down and are equally distributed among the four neighbor fields. Grains that leave the grid are taken out of the system. Figure 15.2 shows a simulation run of the model.

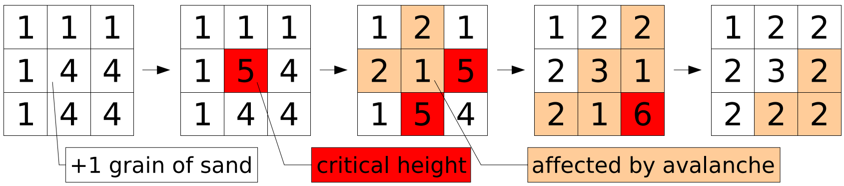

Fig. 15.2: Example for a simulation run of the Abelian Sandpile Model. A single grain of sand causes an avalanche that affects all but the upper left field.

Simulations have shown that this system moves into a critical state, many fields have a height between 1 and 3 – they are stable because another grain of sand will not change the system structure. At the same time, a few fields reside at the critical value of 4 – another grain of sand will cause an avalanche that changes the system's structure [647]. A combination of critical and non-critical field states causes the system to remain stable in most of the cases (because under the assumption of a random distribution of newly dropped sand grains, the probability of hitting a stable field is high), at the same time keeping the possibility for change (at least a few fields can cause a toppling). Bak concludes: "A frozen state cannot evolve. A chaotic state cannot remember the past. That leaves the critical state as the only alternative." [46, p. 6]

Note that some systems (e.g., the Abelian Sandpile Model) have the ability to move themselves into a critical state without external influences. This phenomenon is called *self-organized criticality* [45] and can be observed in most diverse real systems such as earthquakes, stock exchange crashes, traffic jams or sun storms [172].

15.2.7 Hierarchy & Heterarchy

Definition: **Hierarchy**

For this context, we define a hierarchy as a rooted tree. "A tree is an undirected simple graph G" satisfying the condition that "any two vertices in G can be connected by a unique simple path. [...] A tree is called a rooted tree if one vertex has been designated the root, in which case the edges have a natural orientation, towards or away from the root." [623, Rooted Tree]

If a communication system is organized in form of a hierarchy, the communication path between the components is unique, i.e. there is exactly one path between two arbitrary nodes. A level can be assigned to each element, i.e., its distance from the root. This allows us to order the elements partially. Thus, a hierarchy can be seen as an indication of order.

> Definition: **Heterarchy**
>
> *"A heterarchy is a type of network structure that allows a high degree of connectivity. By contrast, in a hierarchy every node is connected to at most one parent node and zero or more child nodes. In a heterarchy, however, a node can be connected to any of its surrounding nodes."*
> [623, Heterarchy]

Compared to a hierarchy, a heterarchy describes a more general type of network; a heterarchy may contain or resemble a tree, but is not limited to it. Although not completely precise, the term *heterarchy* is often used to point out the differences to a hierarchy [609]: since there is no root and all nodes may be arbitrarily cross-linked to each other, no level assignment and thus no order between the nodes can be detected.

If hierarchical and heterarchical organizational forms are applied to communication networks, advantages and disadvantages can be seen. In a hierarchy, there is exactly one single path between two arbitrary nodes A and B. This means that no communication between A and B is possible when one of the nodes on the path between A and B fails; the system would break into two disjoint parts then. A heterarchy can offer a higher fault tolerance because there may be more than one path between A and B. On the other hand, a lot of communication overhead can occur in a heterarchy. This would be the case if *all* nodes were contacted, or if the point of contact was not known in advance. The high number of connections then leads to a lot of communication.

The possibility for a high number of edges in a heterarchy allows for feedback behavior because any node may be able to reach any other node with a small number of hops. In a hierarchy, only neighbor nodes can contact each other so that information usually propagates slower. Thus, a heterarchy can be an indication for feedback.

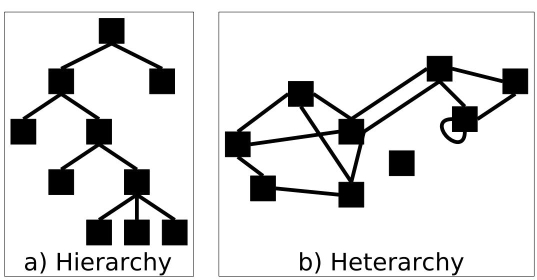

Fig. 15.3: Schematic outline of a) hierarchy and b) heterarchy.

15.2.8 Stigmergy

> Definition: **Stigmergy**
>
> *"Stigmergy defines a paradigm of indirect and asynchronous communication mediated by an environment."* [173, Stigmergy]

It is used mainly in decentralized systems where the individual components of the system communicate with each other by modifying their local environment. An example are ants that can perceive the concentration of pheromones in their environment and adapt their behavior to it. The *source* of the pheromones (usually dropped by other ants) is thereby unimportant. Thus, the environment can mediate triggering ants to behave in a certain way depending on the pheromone concentration. Considering that ants *themselves* are responsible for the pheromone concentration in their environment, stigmergy can be seen as a means to enable self-organization.

15.2.9 Perturbation

> Definition: **Perturbation**
>
> *A perturbation is a disturbance which causes an act of compensation, whereby the disturbance may be experienced in a positive or negative way.* [594, p. 118] (orig. in German)

Perturbations are central to system adaptations mediated by the environment.

15.3 Characteristics of Self-Organization

We now point out some characteristics that can be observed in different systems referred to as self-organizing. A more detailed description about their occurrence in natural, social and information systems is given in [362].

15.3.1 Self-Determined Boundaries

Every system is separated from its environment, so every self-organizing systems is, too. But to enable self-organization, it is important that the border between system and environment is defined *by the system* itself. Otherwise, the environment could shape the system arbitrarily or even completely tear it down. An example for this self-determination is the semi-permeable membrane of a cell. With its help, the cell determines which substances gain access and which are rejected.

15.3.2 Operational Closure & Energetic Openness

To build up its boundaries, a system must be able to operate independently of its environment. That does not mean that the system could survive in complete autarchy – the definition of a system clearly states that there must always exist an environment. A system cannot evolve if it does not obtain some input from its environment such as matter, energy, or information. Thus, a constellation must be found that allows autonomy from the environment on the one hand and interaction with it on the other hand. A system that fulfills these requirements is called *operationally closed* and *energetically open* [402].

15.3.3 Independence of Identity and Structure

Maturana makes a clear distinction between the concept of *organisation* and *structure* of a self-organizing system:

"The term ORGANISATION denotes the relations that must exist between the components of something to recognize it as member of a certain class. The STRUCTURE of something are the components and relations that constitute a certain unity in a concrete manner and realize its organisation."[403, orig. in German] In other words, the structure is a certain instance of components and relations within a system. It may vary (e.g., removal or change of a single component) without the collapse of the system. On the other hand, the organisation is the set of relationships or dependencies that *must essentially* exist for a certain system. With its help, the system can be classified independently of its current structure. In a way, it constitutes the identity of a system, because it does not vary as long as the system exists. To circumvent ambiguities about the term *organisation*, we will denote Maturana's organisation as *identity* from now on.

The distinction between identity and structure allows to explain flexibility and adaptivity, because an influence on a system's structure does not necessarily influence its identity. Thus, the system can be maintained under perturbations. An example for this from software engineering is the principle of information hiding, which postulates that the user should know *what* a certain algorithm does (identity), but not *how* (structure). This allows independence between definition and the current implementation.

15.3.4 Maintenance

To be able to exist over time, a self-organizing system must try to maintain itself. Therefore it needs abilities to repair or recreate defect components. New components do not have to be isomorphic to replaced ones – as long as the identity is sustained the structure may change. The components have to be *viable*, i.e., capable of maintaining identity. Often there are a lot of possibilities for viability so that a form of mutation, one of the basic vehicles for the evolution of a system, can be applied. Another feature of viability (in contrast to isomorphism) is that the system is able to adapt to changes from the environment. If a required input is no longer available the system may change over to use another one and change its structure accordingly.

An example for a self-maintaining entity is a human being – it is permanently recreating its constituting components (the cells). No isomorphic copies are created, though, the appearance of a human being changes over time.

15.3.5 Feedback & Heterarchy

If a system is perturbed, it may have to restructure itself in order to maintain itself. For this reason, the perturbed components must be able to communicate with the rest of the system. A prerequisite for this requirement is the cross-linking of components so that feedback between them becomes possible. This kind of linking allows the components to perform bidirectional communication, amplification or attenuation of external influences, or also recursive application of transformations. All these mechanisms can lead to feedback. Since a heterarchy allows for such cross-linked relations, it is has particular importance for self-organizing systems.

15.3.6 Feedback

Positive feedback can be used to build up viable structures very fast, an example for this is the laser [301]. It consists of a laser medium (e.g., a gas) between two mirrors. If low voltage is imposed the atoms of the laser medium are oscillating in different frequencies so that different colors are produced. The higher the voltage, the more atoms influence their neighbors to oscillate in a viable frequency (which depends on the atoms and the distance of mirrors) – the frequency is amplified through positive feedback.

In this example, not only viable but all kinds of frequencies are produced initially. As the system evolves, the less appropriate ones are sorted out. Randomness plays an important role, because there's no need to know in advance exactly what frequencies are viable.

Negative feedback prevents the system from growing so fast that it would collapse. Even if it is built of viable components only, it can reach a critical size where it might break or cannot react to perturbations fast enough. Therefore, it is necessary to damp positive feedback eventually. In a thermostat the positive feedback of increasing the water flow (and thus the heat) is opposed to the negative feedback of decreasing it. This allows the self-regulatory adaption of temperature independently from the environment.

15.3.7 Criticality

Systems such as the Abelian Sandpile Model (see Section 15.2.6) show that local influences can have global effects: every cell can only influence its 4 neighbor cells. It depends only on the system's state if the addition of another grain of sand triggers a massive toppling, which may reach every cell on the grid. In this case, the reason for the uncertainty of effect is that the system resides in the state of criticality. As described in Section 15.2.6, criticality offers a basic stability as well as the capability for changes. Both properties are essential for evolving systems – instability causes breakdown, inflexibility prevents from growth and adaptivity. Other phenomena which are regarded as being self-organizing, like an evolution [45] or earthquakes [172], are also assumed to reside in a critical state. Therefore, criticality can be seen as an indication for self-organization.

15.3.8 Emergence

It appears that not only the state of criticality but also emergence (see Section 15.2.4) connects local influences and global effects. In systems like ant colonies [174], a set of simple rules in combination with randomness allows the ants to fulfill tasks like building an ant hill or foraging. Although no ant knows the overall environment, the swarm as a whole is able to determine short paths to food sources (a more detailed description is given later in Section 15.4.2). Since the effect depends on the whole system and not only on its parts, it is denoted as *emergent.*

Emergence is often characterized as being unpredictable. Consider the appearance of convection cells in the Bénard system (described in Section 15.2.2). Although the effects of heating and cooling are known as well as the properties of water, the rotational direction of the rolls (clockwise or counter clockwise) cannot be predicted.

15.3.9 Self-Determined Reaction to Perturbations

If a self-organizing system tries to maintain itself, it needs metrics and means to detect and evaluate perturbations. The system can then react, e.g., by adapting the structure to the external influences. Measure and evaluation can be done explicitly, like in a thermostat, but also implicitly. For example, the state of criticality of the Abelian Sandpile Model (see Section 15.2.6) is a way to react to the perturbation of adding grains of sand. The metrics is given by the height of a field, and the evaluation and reaction by the underlying rules for toppling.

The Abelian Sandpile Model also shows that the reaction to such a perturbation is determined autonomously by the system. It is not predictable if the next (randomly located) grain of sand will change the structure of the system or of any part of it. A system can compensate for perturbations on its own, and thus exhibits self-organization.

15.3.10 Reduction of Complexity

Another way that was observed as response to perturbations is the reduction of a system's complexity. An example for this is the Bénard convection (described in Section 15.2.2): if a pot of liquid is heated from the bottom and cooled down from the top, the liquid is shaped in form of *convection rolls*. The behavior of the liquid can then be described by a formula, the behavior of the molecules is determined by a rule. This means that less information is required for their description compared to the state when there was no rule – the complexity is reduced.

Another example is the coupling of multiple independent systems, as described in [293]. The systems described there have symbiotic relations to each other and form a new and stable supersystem which follows rules (whereas the single systems did not). The coupling is irreversible, i.e., once established, the subsystems cannot survive without each other. An instance of this category is the pollination of flowering plants by insects while these get nectar as food.

15.4 Applications in Computer Science

In this section, we describe three examples which show some of the characteristics specified above and their positive effects.

15.4.1 Small-World and Scale-Free Networks

This is a short overview about small-world and scale-free networks with regard to self-organization. For more details we refer to Chapter 6.

Milgram's Experiment

In 1967, the psychologist Stanley Milgram conducted an experiment, often referred to as the small-world experiment [413]. He asked 60 recruits to forward a letter from Kansas to Massachusetts. The participants were only allowed to pass the letter by hand to friends who they thought might be able to reach the destination, no matter if directly or via a "friend of a friend". The perhaps surprising outcome was that the letters reached their target in six steps on average. This lead to the term "small-world".

Small-World Networks

A small-world network is characterized by high sparsity and high clustering. This can be measured by the overall number of edges, which is in $\mathcal{O}(n)$ where n is the number of nodes (in contrast to $\mathcal{O}(n^2)$ for a fully meshed network), and the clustering coefficient which is defined for each node i by:

$$C(i) = \frac{2 \cdot e(i)}{deg(i)(deg(i) - 1)}, \tag{15.1}$$

where $e(i)$ is the number of edges between neighbors of i and $deg(i)$ corresponds to the degree of i [615]. Since $deg(i)$ is the number of neighbors of i, the clustering coefficient describes the relation of actually existing edges between neighbors of i and the maximum number of *possible* edges between them. The more edges actually exist, the higher is the clustering coefficient.

A small-world network can be constructed from an existing regular structured network [615]. In this case, every edge (i, j) is changed to (i, k) with a certain probability p, where k is chosen uniformly at random over all nodes but i and j (with duplicate edges forbidden). The higher p is chosen, the more the network becomes random. Small-world properties are observed somewhere between low and high p [163]. The appropriate value for p is often hard to determine. To cope with these issues, an alternative model was developed.

Scale-Free Networks

The scale-free (SF) model [60] describes networks in a more dynamical way than the small-world model and is mainly based on the two mechanisms:

– Dynamic construction and
– Preferential attachment.

The construction process usually starts with m nodes and no edges. New nodes are added incrementally, and a constant number of edges is attached to them (to stay within $\mathcal{O}(n)$). The probability that an edge from a new node n is connected to a certain node with degree k_i is given by:

$$P(n \to k_i) = \frac{k_i}{\sum_j k_j}. \tag{15.2}$$

Due to the additivity and homogeneity (on algebraic grounds) of this function, the correlation between the degree of a node and the probability for new nodes to connect to it is linear. This leads to a *scale-free* network, the structure of the system is independent of its current size (Figure 15.4). Scale-free networks show the property that most nodes have a small number of connections while only a few are highly meshed ("hubs"). This relation can be mathematically described by a power law: $P(k) \sim k^{-\gamma}$ (where γ is a system constant). Therefore, these networks are also called *power law networks*. Sometimes they are also denoted *fractal networks*, signifying a correlation to fractals[1]. For power law networks as well as fractals, a part of the system has the same structure as the whole (Figure 15.4). This property, which is called *self-similarity* for fractals, is just another expression for freedom of scale.

Connection to Self-Organization

In addition to Milgram's small-world experiment about social networks [413], other popular networks such as the Internet [325] and the WWW [10], the pattern of viral infections [59], relations between actors or scientists and even computer programs [434] show small-world and scale-free properties. In all cases, the properties appeared as an accidental feature rather than intentionally incorporated – the properties arose in a *self-organized* form.

The incremental building of a scale-free network exposes another correlation to self-organization: every node has only a *local* view on the system and makes a *local* decision about its connections. This results in a *global* scale-free structure, apart from external influences. This structure is a mixed form of hierarchy and heterarchy. It supports scalability, the overhead for connecting to it is constant and the diameter is very small. It is also robust, failure or attack of a random node have little effect on the overall system with high probability due to a high degree of redundancy [18]. However, a targeted attack on hubs can divide the network into disjoint parts (again of the same structure). Randomness is used for the linking between components, which results in flexibility and makes the occurrence of single points of failures (SPoFs) less likely.

---

[1] A fractal is a geometric object which can be divided into parts, each of which is similar to the original object. [623, Fractal].

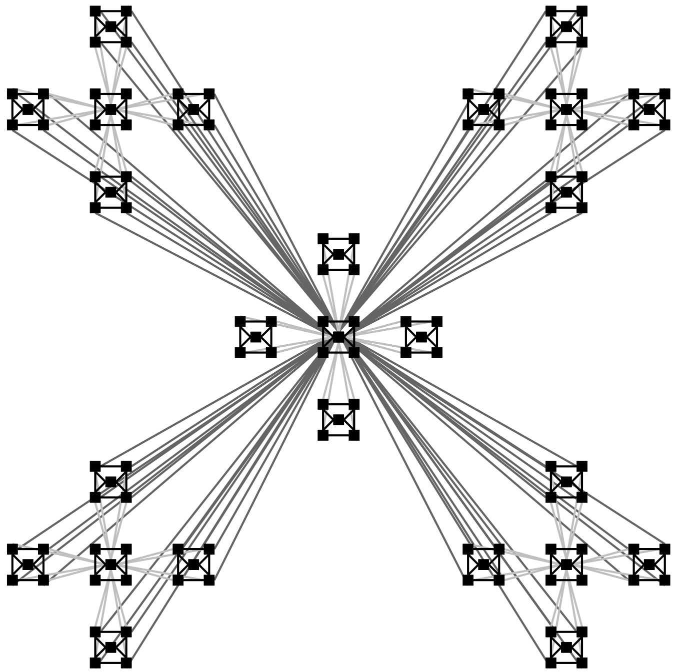

Fig. 15.4: Example for a hierarchical, fractal, scale-free network. Every part of the system has the same structure as the whole.

15.4.2 Swarming

Swarm intelligence is a property that has been investigated in the area of artificial intelligence. Swarm intelligence indicates the ability of single agents to make correct decisions without knowledge of all other members of the swarm and the overall system. A system composed of such agents is coordinated without a dedicated control instance. According to [415], there are five fundamental principles of swarms:

– Proximity – a swarm can make simple calculations about time and space.
– Quality – a swarm can react to indications by the environment.
– Diverse response – activities can be performed in different ways.
– Stability – not every environmental change modifies the swarm.
– Adaptability – a swarm can change its behavior if that seems promising.

These properties have been implemented in many agent-based models. We use the class of *ant algorithms* [87] for our further explanations.

Ant Algorithms

Ant algorithms are used to solve a problem by means of many agents called *ants*. The strategies of these ants have been inferred by watching nature, where real ants use chemical substances ("pheromones") to communicate with each other. Ants spread pheromones while moving around and detecting the trails of their conspecifics. The substance evaporates over time. Ants follow traces with a probability proportional to the strength of the pheromone signals. One application of ant algorithms is the traveling salesman problem (TSP), i.e., finding the shortest cycle through a given number of cities. Ant algorithms are based on a model of foraging behavior of real ants as illustrated in Figure 15.5.

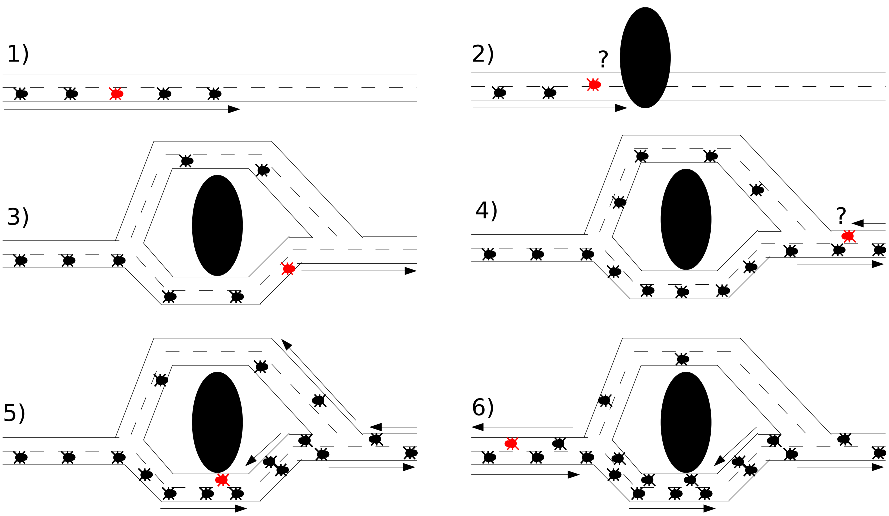

Fig. 15.5: How real ants find the shortest path. 1) Ants move from their nest to a food source. 2) They arrive at a decision point. 3) Some ants choose the upper path, some the lower path. The choice is random. 4) Since ants move approximately at constant speed, the ants which choose the lower, shorter path reach the opposite decision point faster than those which choose the upper, longer, path. 5) Pheromone accumulates at a higher rate on the shorter path, so consequently more and more ants choose this path. 6) The decision of the following ants is influenced by the higher pheromone concentration.

A swarm of ants can find cooperatively an approximative shortest path from their nest to a source of food. This observation has been taken advantage of for finding a solutions to the well-known TSP problem [174].

Connection to Self-Organization

A swarm of artificial as well as real ants exhibits many indications of a self-organizing system. First of all, ants are organized in a form of heterarchy (all ants are equally important), although they don't communicate directly. The ants use stigmergy for communication rather than an explicit communication channel. Spreading of pheromones is a form of positive feedback while the evaporation corresponds to negative feedback. No ant has global knowledge but acts upon very basic, local rules. The use of randomness makes it even simpler. While the collective swarm may find the shortest path (or at least a good approximation of it) based on teamwork, a single ant, following simple rules, could have never completed the task on its own. So the overall collective behavior can be called *emergent* based on interactions the constituent elements that follow simple rules for their behavior.

15.4.3 Cellular Automata

The concept of *cellular automata* can be traced back to the 1940s when John von Neumann investigated self-replicating systems [610]. A cellular automaton can be explained as an accumulation of many deterministic finite automata which all have the same set of rules. It consists of an infinite, n-dimensional grid of homogeneous cells c_i which have a state $s(c_i, t)$ at time t. For each cell c_i, a neighborhood $N(c_i)$ is defined. It can be chosen according to certain metrics, e.g., the two neighbors in each dimension, or the two neighbors and additionally c_i itself. For each combination of states of $N(c_i)$, a rule is defined which determines $s(c_i, t+1)$. The rules are valid for all cells so that the total number of rules is given by the number of cells in the neighborhood to the power of the number of possible states $|N|^{|s|}$.

A very simple example is the *mod* 2-automaton. It is one-dimensional ($n = 1$), has two possible states ($s(c,t) \in \{0,1\}$) and has only the single rule for all cells y:

$$\forall y : N(y) = (x, y, z) \quad \Rightarrow \quad s(y, t+1) = (x + z) \; mod \; 2. \tag{15.3}$$

Eq. (15.3) implies that $s(y, t+1)$ is independent of $s(y, t)$. A plot showing the states of this simple automaton over time is illustrated in Figure 15.6.

Unexpectedly, the automaton builds the structure of a well-known fractal, the Sierpinski-triangle. Other automata show similar behavior, but there are differences which lead to the following classification [628]:

- Class 1 (trivial automata): lead to the same state for each cell, independent of the initial state.
- Class 2 (periodic automata): lead to a fixed state for each cell, dependent on the initial state.

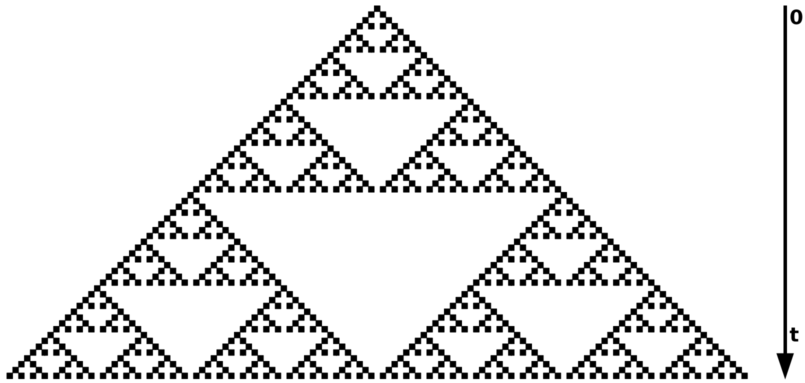

Fig. 15.6: Visualization of the *mod* 2-automaton (time progresses from top to bottom): the Sierpinski-triangle

– Class 3 (chaotic automata): change incessantly without observable structures.
– Class 4: create different, complicated structures that move along the grid over time.

Matthew Cook has shown that class 4 automata can be used for computations and furthermore are Turing-complete [628], which is noteworthy taking the simplicity of the model into account.

Connection to Self-Organization

Animations of cellular automata, showing moving patterns like blinkers or gliders, trigger assumptions about an underlying self-organization. Indeed, several indications can be identified, the iterative and recursive application of simple rules to a local neighborhood may cause an evolution of complex structures. The structures are continuously reproduced. Furthermore, some patterns are self-stabilizing. The current state of a cell is used as input for the modification of future states of the same cell which is a form of feedback. The resulting structures are emergent because no single cell could produce them.

15.5 Conclusions

Self-organization increasingly attracts interest within several areas of computer science. By incorporating self-organization autonomous operation can be fostered. Decentralized control or self-management could be attractive results. Self-organization can be found in many disciplines and a comparative study may therefore reveal stimulating insights. Enabling features of self-organization were identified in this chapter including complexity, feedback, emergence, criticality, heterarchy, stigmergy or perturbation. Characteristics can be used to classify Peer-to-Peer and other systems concerning their degree of affinity to self-organization. Self-organization has widely been praised as being the key for incorporating attractive features into networks and systems. Small-world and scale-free networks are robust as well as efficient, a swarm may subsume a very large number of agents without global control or knowledge, or cellular automata may trigger global structures purely based on locally applied rules. Autonomy, scalability, flexibility or robustness may largely be provided as a result. Other implications of self-organization (for example, an unpredictability of behavior, danger of deadlock creation etc.) may be less attractive.

Most problematic from computer science point of view, however, is the fact that it seems hard to impose control and to exercise management onto those systems for more efficiency, security purpose or general goal orientation. The analysis provided in this chapter aims at providing a first step towards a better understanding of self-organization in general. Further research is needed, however, before more comprehensive and goal-driven control and management will be possible in particular.