Archetypical Patterns in Agent-Based Models

Hofstede, Gert Jan; Chappin, Emile

DOI
10.1007/978-3-030-61503-1_31

Publication date
2021

Document Version
Final published version

Published in
Advances in Social Simulation - Proceedings of the 15th Social Simulation Conference, 2019

Chapter 31
Archetypical Patterns in Agent-Based Models

Gert Jan Hofstede and Emile Chappin

Abstract Complex systems produce recognizable self-organized patterns across time. This conceptual paper consists of a systematic reflection on what kinds of archetypical patterns systems can show, and in what kinds of cases these patterns could occur. Agent-based models are used to exemplify each pattern. We present a classification of the breadth of typical patterns that agent-based models can show when one runs them. The patterns fall into three categories: resource use, contagion, and output patterns. These are pattern archetypes; most real-world systems, and also most models, could and will show combinations of the patterns. In real systems, the patterns will occur as phases and building blocks of developments. These are patterns frequently occurring in real-world systems. The classification is the first of its kind. It provides a way of thinking and a language to non-mathematicians. This classification should be beneficial to those researchers who are familiar with a real-world pattern in their discipline of interest, and try to get a grasp of pattern causation. It can also serve in education, for giving students from a variety of disciplines an idea of the possibilities of agent-based models.

Keywords Agent-based model · Pattern · Tragedy of the commons · Fixes that fail · Power law · Tipping point

G. J. Hofstede (✉)
Information Technology Group, Social Sciences, Wageningen University, Wageningen, The Netherlands
e-mail: gertjan.hofstede@wur.nl

North-West University, Potchefstroom, South Africa

E. Chappin
Delft University of Technology, Technology, Policy and Management Complexity, Delft, The Netherlands
e-mail: E.J.L.Chappin@tudelft.nl

P. Ahrweiler and M. Neumann (eds.), *Advances in Social Simulation*,
Springer Proceedings in Complexity,
https://doi.org/10.1007/978-3-030-61503-1_31

Introduction

In this age of interconnection of local social and economic systems, of human-induced climate change and of global pandemics, there is an increased need for recognizing, clustering and exploring (possible) patterns in complex systems. Science provides many methods to do so. Scientists in various disciplines and policy makers in many of today's socio-technical systems need to be able to recognize patterns and to know whether and how they could intervene to improve the systems they are part of and govern [1]. Some interventions could be crucial, others futile; and the difference can be clarified by studying underlying patterns in the behaviour of the systems that should be managed [2].

A pattern is a regularity in the world that can be observed by the senses or is inferred from collected data. A pattern can occur in many guises, for instance geometrical patterns representing mathematical principles, spatial-temporal patterns capturing developments of systems over time, and abstract patterns that describe conceptual ideas. This paper aims to identify and summarise archetypical patterns in complex systems including the mechanisms that form a possible cause for the pattern to come about. A key method in the complexity science that deals with patterns is agent-based modelling (ABM) [3]. ABM enables an explicit discussion on how patterns emerge on the system level out of lower level interactions. We use this key feature of ABM to define and describe a set of archetypical patterns. The actual patterns shown by real-world systems may be a combination of archetypical patterns, often containing recognizable elements of one or more of them. We use ABM for its conceptual closeness with real systems that consist of elements interacting across space and time. This will make our classification and examples accessible to practitioners of complex systems.

The article is structured as follows. Section "Materials and Methods" describes materials and methods. Section "Agent-Based Models and Patterns" discusses the elements of agent-based models that make them such prolific reproducers of patterns. Section "Mechanisms" lists a set of mechanisms by which patterns can be generated for ABMs. Section "Overview of Patterns" provides an overview of ten archetypical patterns with examples for all those patterns. We illustrate the patterns with simulation results from Netlogo [4] models from its online model library in more detail. These examples are accessible to anyone for closer scrutiny. The paper ends with a discussion and outlook in Sect. "Conclusions and Discussion".

Materials and Methods

This article derives from the experience of the authors with studying complex systems in various domains. In particular we have been modelling policy-relevant socio-technical and socio-ecological systems since many years. For this article, we deliberately use example models in the Netlogo models library [4]. This allows any reader access to the materials and methods to reproduce the figures and experiment with the models. We characterize each pattern with an appropriate model, in terms of a single run or of a set of runs with that model.

Agent-Based Models and Patterns

Agent-based modelling does not model the patterns we observe in complex systems directly. It models smaller building blocks: actions and interactions of agents and their environment that make up the system. If the growth of the anthill is the pattern, ABMs model the individual ants. They do so through agent capabilities such as observation and various kinds of action. These basic components and possible conceptualizations in agent-based modelling allow for a myriad of system-level patterns to occur during model runs. [3]. This is a strength, since systems in the real world also display these complex patterns [5]. Both in an ABM and in the real world, the patterns we observe are the result of what one would call in everyday language "a set of coincidences". One thing always leads to another. System elements act, react, and interact, and over time, self-organize into a pattern without necessarily having any intention of producing that pattern. In other words, patterns are emergent from the aggregated behaviour of agents. As a consequence, it is often difficult for developers of ABMs to understand why their models behave as they do—this is a serious limitation to be confident that models are actually useful [6].

The reasons why ABMs are so versatile are that:

- ABMs are primarily declared at the level of agent types. Any model run can be populated by many agents. The processes have a lot of freedom, e.g. in sequence, and in random differences across model runs.
- ABMs have agents of potentially many types and possible conceptualizations, and in potentially high numbers [7, 8]. Agents can be linked in a variety of ways. For instance, they could be neighbours, share characteristics, exchange information, beget one another, or serve as food to one another. ABM can be spatial and locations can be conceptualized in many ways.
- ABMs are temporal and model runs show path dependency. Agents can affect, or can be affected by, other agents, by aspects specific to their location and by system-wide developments. This latter possibility is often called 'second-order causation', or sometimes, with a term by Rosaria Conte, 'immergence', to reflect the fact that the system pattern that occurs during a run exerts a top-down influence on the agents [9].
- The consequence of the above points is that ABMs simulate patterns that are not directly coded, but instead emerge as a result of the aggregated actions of agents while a model runs. These patterns are often recognised by those who know the real-world systems on which models are based. The ABM thus helps these real-world system experts to investigate how these patterns are caused: 'abduction' is the methodological term for this.

Mimicking real-world systems, ABMs can show a huge variety of patterns. The same ABM can show several different patterns within a run, or across runs, depending on inputs and settings. Pattern-oriented modelling (POM) is one of the ways of establishing the validity of an ABM [10, 11]. If a model can be shown to replicate several patterns occurring in the corresponding real-world system, this increases confidence in its validity to make claims about real-world systems. These patterns are often spatio-temporal, by the nature of agent-based models as playing out over simulated time in a simulated two-dimensional space. In this case they can be visually observed in the simulated world while the simulation is running. They could also involve the fluctuation of one or more output variables over time. In that case one could show them by plotting the variable(s) over simulated time.

The variety of patterns occurring in real-world systems is staggering. A number of them that occur frequently and have recognizable 'Gestalt' have been named and are recognized in their occurrence, and sometimes in ways to deal with them. For instance, the 'tragedy of the commons' [12]. We can call such patterns 'archetypical'. How many of these archetypical patterns exist? It is a very relevant question for policy makers, researchers and anyone else that considers the possible dynamics of complex systems. Unfortunately, a systematic overview of the patterns that emerge from ABMs, in analogy to what has been developed for System Dynamics [13, 14], is still lacking. We build a systematic analysis for this purpose and provide a first reflection on what archetypical patterns can result from ABMs.

Mechanisms

Researchers and policy analysts are the architects and builders of ABMs: they conceptualize agents and decide what agents 'can do'. More precisely, they determine and code what agents perceive and deduce, whether agents and/or their environment are heterogeneous, how agents move, meet and interact, when they are born, and when they die. In other words, they build the mechanisms by which the agents—when the model runs over simulated time—collectively bring about system patterns.

Most of these patterns could be obtained in a variety of ways, employing a variety of mechanisms. Often, the modeller has some real-world knowledge or relevant theory about agents from which possible mechanisms can be deduced; in the absence of knowledge, Occam's Razor suggests that the simplest possible mechanism may be preferred on account of sparsity. Actually, one merit of ABM is that they often show that surprisingly simple agents and mechanisms can cause realistic, recognizable patterns. In other words, they show the multi-level nature of causation in systems: system-wide behaviours do not require system-wide causes. We give a rough categorization of mechanisms:

- **Direct interaction** between agents. Agents affect other agents' parameters through some form of direct interaction: killing, eating, taking or providing resources, communicating…

- **Indirect interaction** between agents. Agents may coordinate in a decentral fashion, not by directly interacting or even observing, but by responding to the same central entity/parameter that they shape together. Alternatively, agents shape the system developments together through 'stigmergy': a form of second-order causation in which agents do not directly interact, but change their environment and thus alter the decision-making of those that come after them. This occurs e.g. when agents form 'elephant paths' by being attracted to other agents' traces.
- **External drivers** may affect agents. These are exogenous to the model, so agents do not affect these values themselves. Exogenous changes may cause particular system patterns to emerge.
- **Memory**. Memory in the system can cause various effects, such as piling up, lack of responsiveness, or learning. Small effects repeated many times can lead to piling up, resulting in unanticipated system behaviour. This can happen when densities matter and are affected by these changes. Lack of responsiveness, also called myopia, can occur when obsolete information is kept and crowds out new information. Memory is then substituted for observation. More advanced forms of relying on memory could be called learning.

Overview of Patterns

This section briefly describes ten patterns, which are numbered below and grouped in the following classes: resource use (3.2), contagion (3.3), and output pattern (3.4). Table 31.1 provides an overview. Besides the category and name, it provides six emergent characteristics of each pattern. All of these occur as a result of the combined actions of agents.

Two emergent characteristics could occur at some place or time during a model run:

- *Positive feedback loop:* does it include self-reinforcing feedback behaviour;
- *Balancing loop with delay:* does it include self-limiting feedback behaviour.

Four emergent characteristics are properties of an entire model run

- *Finite:* Does it have a definite end;
- *Asymptotic:* if there is a central output variable, does it tend to a fixed value;
- *Repetitive:* does it repeat itself;
- *Ergodic:* does each run, if let go long enough, produce all the possible model states;

We see the patterns in Table 31.1 as archetypical: each of them is typical for ABMs, in the sense that they are frequently encountered, and they are also elementary, in the sense that they are basic and emerge from simple models. The patterns are not necessarily mutually exclusive and systems/models could show several of these patterns, or combinations of them, across simulated time.

Table 31.1 The overview of ten patterns. The table first gives an overview of possible mechanisms causing these patterns. ‘yes’ means a mechanism has to occur for producing a pattern, ‘no’ that it cannot occur. The table also gives the occurrence of four likely emergent characteristics of model runs

Pattern name	Mechanisms				Emergent characteristics					
	Direct interaction	Indirect interaction	External drivers	Memory	Positive loop	Balancing loop	Finite	Asymptotic	Repetitive	Ergodic
Resource use										
1. Tragedy of the commons	Maybe	Yes	Maybe	Maybe	Yes	No	Yes	Yes/No	No	No
2. Fixes that fail	Maybe	Yes	Maybe	Maybe	Yes	No	Yes	Yes/No	No	No
3. Sprawl	Maybe	Yes	Maybe	Maybe	Yes	Yes	No	No	Yes/No	No
4. Coexistence	Maybe	Yes	Maybe	Maybe	Yes	Yes	No	No	Yes/No	Yes
5. Ecosystem engineering	Maybe	Yes	Maybe	Maybe	Yes	Yes	No	No	Yes/No	No
Contagion										
6. Synchronization	Yes	Maybe	Maybe	Maybe	Yes	No	No	Yes	No	No
7. Spatial clustering	Yes	Maybe	Maybe	Maybe	Yes	Yes	No	Yes	No	No
8. Attribute patterns	Yes	Maybe	Maybe	Maybe	Yes	Yes	No	Yes	No	No
Output pattern										
9. Tipping point in output	Maybe	Maybe	Maybe	Yes	No	No	Yes	Yes/No	No	No
10. Power law distribution	Maybe	Maybe	Maybe	Maybe	No	No	Yes/No	No	No	No

Mechanisms are about what agents do to one another, while patterns are the emergent results. Any or all of the mechanisms can occur in a system. Some are indispensable for producing a certain pattern, but most are optional. Their precise nature and relative strength can vary. There could be mutual influences between mechanisms. Memory could lead to transition from one pattern into another, for instance through selective reproduction of agents with certain traits. Table 31.1 summarizes the complexity of pattern causation.

Agent-based models are intrinsically about more than one level of aggregation. In a running model, or in a real system for that matter, there can be more than two such levels of aggregation. A mechanism can lead to emergent results that themselves can serve as a mechanism for higher-level aggregated patterns. For instance, a plague organism can go through a phase of exponential growth before mechanisms start to occur that slow down this growth. We call such a phenomenon an ‘emergent mechanism’ to indicate that it operates not at the level of individual agents but at the level of collections of agents.

In fact, the distinction between ‘emergent characteristic’ and ‘pattern’ can be hard to make. ‘Pattern’ is intrinsically a recursive concept. For instance, positive or balanced loops can be considered patterns in their own right. However, they often occur for a time in a sequence of patterned elements that have a recognizable name of their own. Therefore, we consider them to be emergent mechanisms that can occur during a model run as part of its overall pattern. That is why we list them as columns in Table 31.1.

Section “Emergent Characteristics: Examples for Feedback Loops” describes emergent characteristics for feedbacks. Sections “Resource Use Patterns”, “Contagion Patterns”, and “Output Patterns” describes the ten patterns as listed in Table 31.1. We present these results in the following way:

- Title and brief description
- Iconic example(s) from the real world with a Netlogo example if available. To illustrate the ubiquity of each pattern, we give examples from four fields: society, biology, physics, and man-made technical constructions. We also present typical simulation results. Note that some patterns are apparent from observing one model run, e.g. patterns across simulated space/time. Others are apparent from graphs created based on a model run or even across runs. Making patterns apparent is one of the necessary skills of modellers.

Emergent Characteristics: Examples for Feedback Loops

We first give examples for feedback loops, which are two important emergent characteristics. All resource use or contagion patterns involve feedbacks. In resource use, agent behaviour feeds back into resource availability for the next time step. In contagion, agent behaviours directly feed back into one another. Mutual feedback loops are thus elements of all the first eight patterns of Table 31.1. These loops can occur as patterns in their own right, show repeating waves of changes. However, they

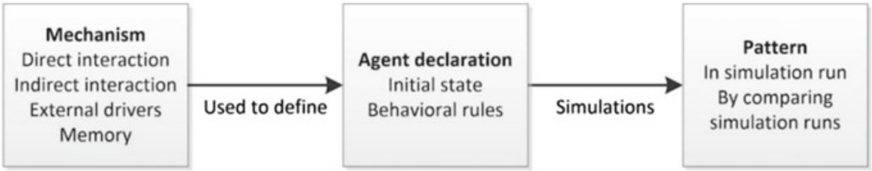


Fig. 31.1 Mechanisms, agent declarations, and patterns

tend to occur only locally or for a while, in the context of a wider-scale pattern. Both positive and balanced loops could result from the same model, depending on model settings as to delay, limits, mutual feedback, and number of variables. Because these are so simple, and occur as building blocks in the context of the eight other patterns, we treat them as emergent mechanisms in this overview.

Positive feedback loop. This is a loop the amplitude of which grows indefinitely. The resulting pattern has the shape of an exponential curve. It is obviously not sustainable for ever. The mechanism is that in each time step, the current quantity N of a variable (N > 1) gets multiplied by a factor that is proportional to N. This causes N to grow ever faster. Examples:

- Society: disease spread in the early phases of an epidemic or pandemic (Netlogo: epidemics). The start of hype cycles.
- Biology: population size in the absence of size-dependent mortality. In Netlogo, the sheep population in Netlogo sheep-wolves simulation in a run without grass, after all wolves have died out (see Fig. 31.1).
- Physics: objects falling to the ground without resistance.
- Engineering: two microphones circuiting by picking up and reinforcing one another’s signal.

Balancing loop with delay. A wave pattern that continues indefinitely. A phenomenon of this category is sometimes called ‘limits to growth’: feedback loops that self-limit with resource availability, leading to more or less stable oscillations, whether desired or not.

Many positive feedback loops will balance at some point when other aspects of a simulation come in effect. As a result, positive feedback loops and balancing loops can result from the same model, depending on model settings as to delay, limits, mutual feedback, and number of variables.

Mechanism: a positive feedback is countered by a negative feedback, thereby providing balance, pushing the system towards a particular equilibrium or towards a kind of oscillation. Examples:

- Society: market cycles in which quantity produced and price fluctuate. The pork cycle is the archetype.

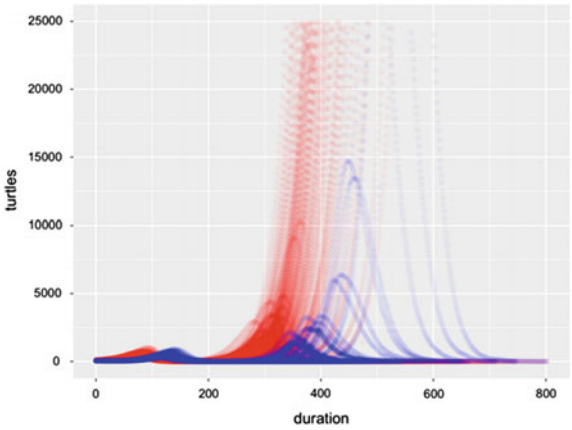


Fig. 31.2 Results from 1,000 runs with Wolf Sheep Predation model from the Netlogo 6.1 model library. Not all runs are similar, but the pattern is that either wolves (in blue) die out, which results in exponential growth of sheep (in red); or alternatively, if sheep die out first, the wolves die out afterwards

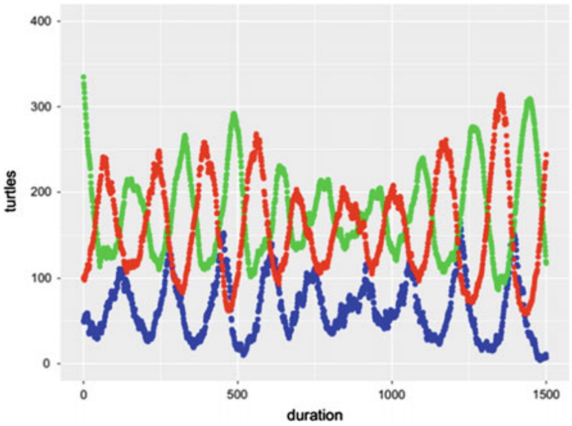
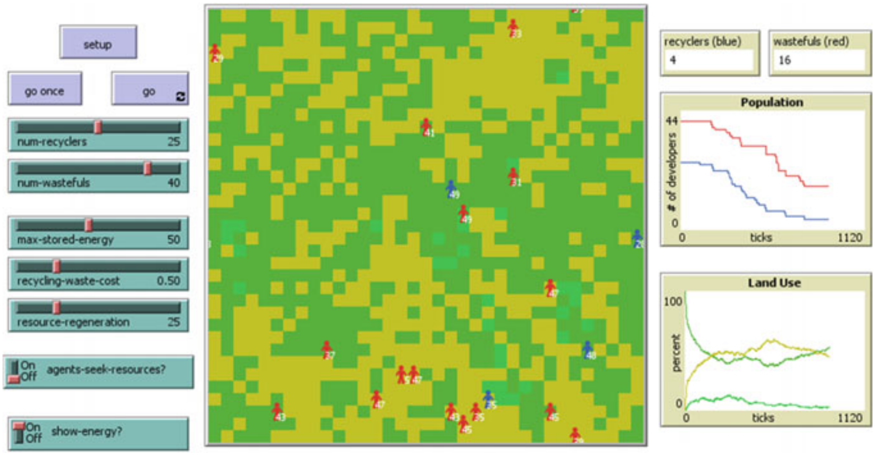


Fig. 31.3 Results from a single run with Wolf Sheep Predation model from the Netlogo model library with grass enabled. Not all runs are similar: this particular run show balancing loops with delay, which show as cyclical behaviour (or a dynamic equilibrium) of grass (green), sheep (red) and wolves (blue)

- Biology: populations that stabilize at similar birth/death rates and without net migration. In Netlogo: wolf-sheep-grass model, when population variations are buffered by the availability of grass for the sheep (see Figs. 31.2, 31.3).
- Physics: evapotranspiration—rainfall cycles.
- Engineering: thermostat of a shower.

Resource Use Patterns

Resources are quantities that agents need for survival, and may exhaust in doing so. Agents or grid cells (‘patches’) may also generate resources. Typically, agent motivations include behaviour directed at finding these resources. In the following patterns, which can contain elements of those mentioned above, an additional element is that the locations (grid cells, patches) in the model contain resources. Agents use



num-recyclers 25

num-wastefuls 40

max-stored-energy 50

recycling-waste-cost 0.50

resource-regeneration 25

On Off agents-seek-resources?

On Off show-energy?

recyclers (blue) 4

wastefuls (red) 16

Fig. 31.4 Recycling model after a tragedy of the commons has occurred. Green patches are fertile; yellow ones are degraded; lime ones have been restored by the blue recycler agents. Bottom right graph: the environment quickly degraded in the first 100 ticks. Top right graph: 'Recyclers' first started to die, exhausted from clearing up; then 'wastefuls' followed, not finding any nourishment. After 100 ticks, the environment starts to recover, but with a much lower population

these, leading to various possible patterns with a spatial component. This means the agents do not interact with one another other than through shared resource use; the technical term for such indirect interaction is 'stigmergy'.

1. **Tragedy of the Commons**. This is a missing feedback loop between resource users and resources that leads to resource exhaustion and then death of all resource users. In the typical case, where resources are supposed to be shared and maintained by a commons, but free-riding occurs, it is frequently named 'Tragedy of the Commons'. Netlogo: recycling (Fig. 31.4). There are also other cases. A calamity such as a wildfire is also characterized by a lack of feedback and by exhaustion of resources, if not resource users. e.g. Netlogo: fire. Examples:

   - Society: tragedy of the commons. Climate change.
   - Biology: locust plague.
   - Physics: mineral exhaustion.
   - Engineering: the interventions needed to deal with strategies of software using internet bandwidth for non-crucial processes, while the internet may break down in times of a crisis.

2. **Fixes that fail**. Feedbacks are made by agents with an aim in mind that worsen the system-level predicament they were intended to solve. Examples:

   - Society: gun purchase to defend against violence.
   - Biology: lemmings taking to the sea. They sacrifice themselves for the good of the ecosystem, but that is unlikely to be their aim.

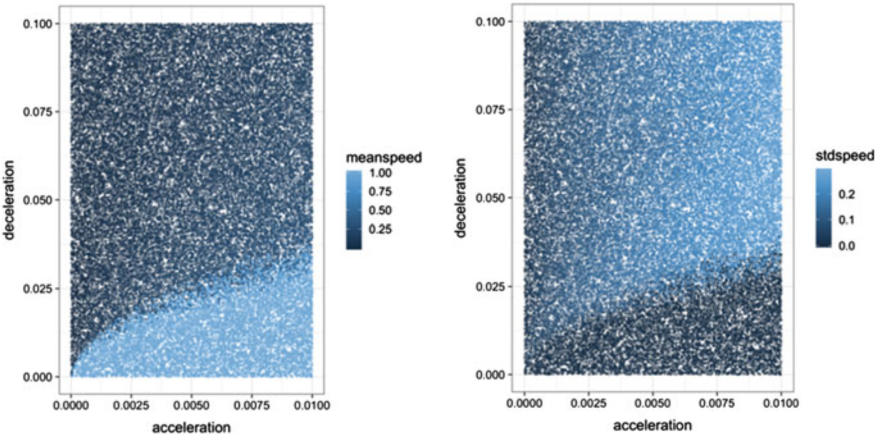


Fig. 31.5 Results from 50,000 runs with the Traffic Basic model from the NetLogo model library, varying deceleration and acceleration speeds, and varying the number of cars between 18 and 22. On the left, the average car speed and on the right the standard deviation of speeds between cars, both at time tick 20,000. Fast deceleration (late braking) is a fix that fails: it leads to traffic jams, in particular in combination with slow acceleration. The cross-over between jammed traffic and flowing traffic is surprisingly sharp

- Physics: unstable control of dynamical systems that are too slow or fast in their response to sensors
- Engineering: Netlogo: traffic basic, where abrupt braking can worsen the queueing time (Fig. 31.5).

3. **Sprawl**. A spatially bound process of resource depletion and renewal leads to spatial patterns. This occurs when positive feedbacks spread slowly. Examples:

    - Society: urban settlement Netlogo: urban sprawl (Fig. 31.6).
    - Biology: growth of lichens on stones and trees. Witch circles of mushrooms. Vegetation in arid landscapes, where roots can improve the capacity of the soil to retain water, leading to sprawling patches.
    - Physics: ripples.
    - Engineering: spread of technical innovations.

4. **Coexistence**. These are situations where populations coexist that compete for a resource, creating an ecosystem. Each population creates circumstances that facilitates the development of another type of agent. This could result in a feedback loop akin to sprawl mechanisms, but caused by several types of agents instead of one. The mutual influences need not be intentionally beneficial; the populations could be competing. This is for instance the case in Netlogo: cooperation (blue and red cows). One species may drive the other to extinction, but under some parameter settings, coexistence occurs (Fig. 31.7). Examples:

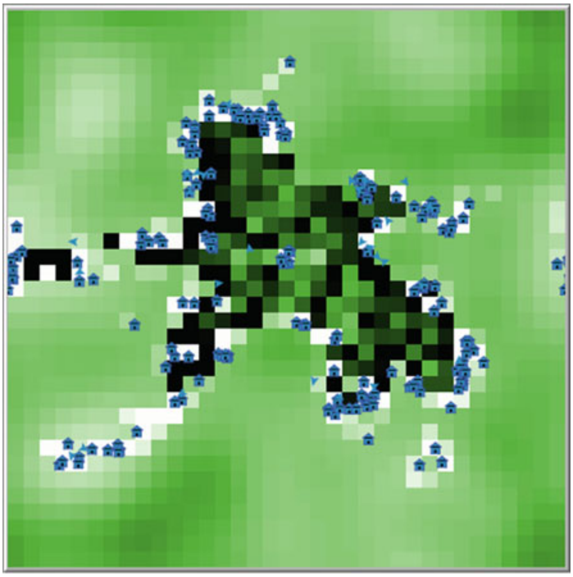
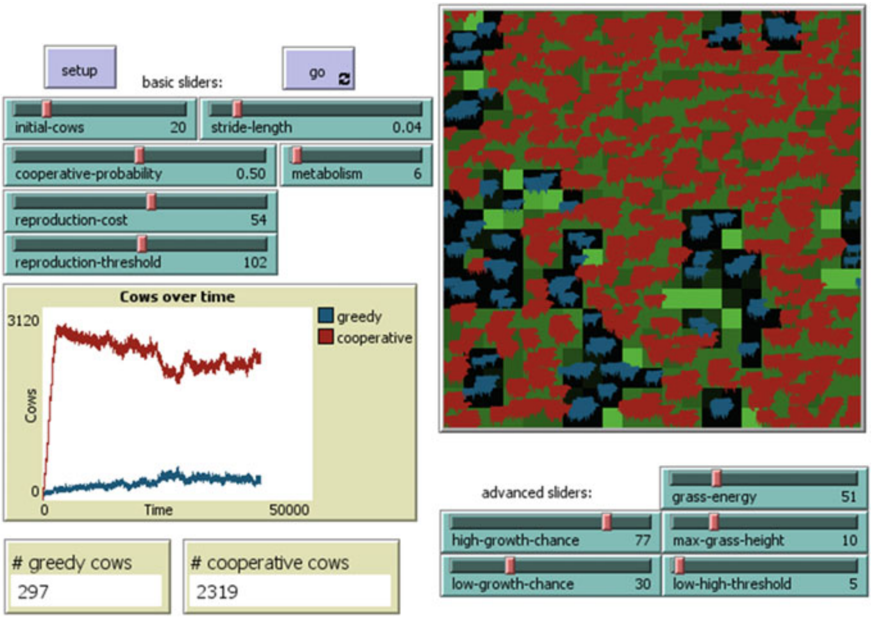


Fig. 31.6 Snapshot of a run with the Urban Sprawl model from the NetLogo model library. During the first phase of the simulation run, agents spread slowly from the centre, increasing the fitness of their environment (lighter patches). Later, the environment collapses (flips to black patches) and slowly recovers (patches become lighter and are re-colonized)

- Society: economy. Actors complement one another in creating and using resources.
- Biology: coexistence of different types of animals that use the same resource. Netlogo: cooperation (Fig. 31.7).
- Physics: the effects that gravity has on star systems, keeping planets in orbit.
- Engineering: various modes of transport. New ones typically did not supplant but complement the existing ones.

5. **Ecosystem engineering**. This is the more general case of which Sprawl is a special one. The joint activities of agents lead to differentiation of an environment that is at first undifferentiated. So here, the 'resource' is not necessarily in the patch itself, but in the emergent configuration of patches. Locations randomly picked thus become endowed with new roles. Social animals do this, e.g. humans or ants. Examples:

- Society: paths. Netlogo: paths (Fig. 31.8). institutions. Institutions effectively constitute niches that allow certain actions and inhibit others. Informal institutions can be considered paths in symbolic space.
- Biology: beavers, ants and termites. Netlogo: termites.
- Physics: star and planet formation.
- Engineering: centrifuges.



initial-cows 20
stride-length 0.04
cooperative-probability 0.50
metabolism 6
reproduction-cost 54
reproduction-threshold 102
# greedy cows 297
# cooperative cows 2319
grass-energy 51
high-growth-chance 77
max-grass-height 10
low-growth-chance 30
low-high-threshold 5

Fig. 31.7 'cooperative' (red) cows that leave some grass behind quickly colonize the empty world. 'Greedy' (blue) cows finish the food on their patch and then some migrate, the others die out, after which the patch recovers (light green). When the world is full, a dynamic equilibrium occurs between the cooperative and the greedy cows. If the cows were quicker (increased stride-length), the greedy ones would exterminate the cooperative ones with a hit-and-run strategy

Contagion Patterns

These patterns involve mutual observation and adjustment by agents, with an important role for space, time or other observable attributes.

6. **Synchronization**. Agents come to synchronize their behaviour in time, leading to temporal clustering, in other words to a system-level pulsing pattern. Synchronous collective actions are real-world cases. Netlogo: fireflies (Fig. 31.9). In human societies, the expectation of future events can lead to temporal clustering, or in crowd formation. Examples:
   - Society: pork cycles [15]. Crowd formation. Netlogo: El Farol bar.
   - Biology: pulsing by fireflies, leading to clearer attraction of potential mates. Netlogo: fireflies.
   - Physics: solar cycles.
   - Engineering: time steps in computer memory.
7. **Spatial clustering**. Agents orient themselves in space by observing and copying their neighbours, leading to spatial grouping and patterned movement. This is also a case of para-synchronization: agents copy one another's behaviour with

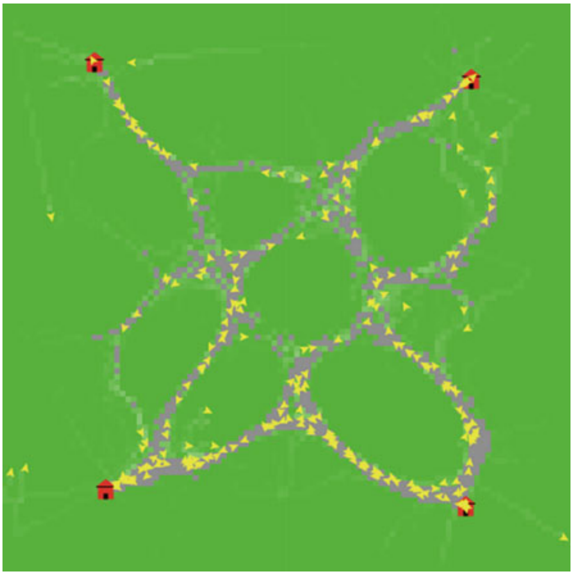
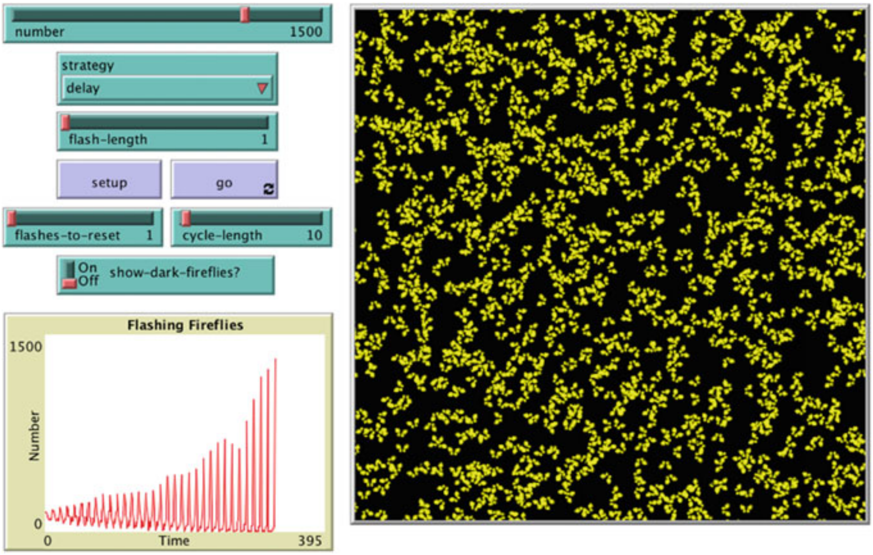


Fig. 31.8 Paths model. The agents flatten the grass (lighter shade) by walking. Repeated walking over a patch creates a path (grey). They have walked randomly, creating random paths because they prefer to walk where others have flattened the grass. Then they were given targets (four red houses). They keep using their existing paths. These decay only slowly



number 1500

strategy delay

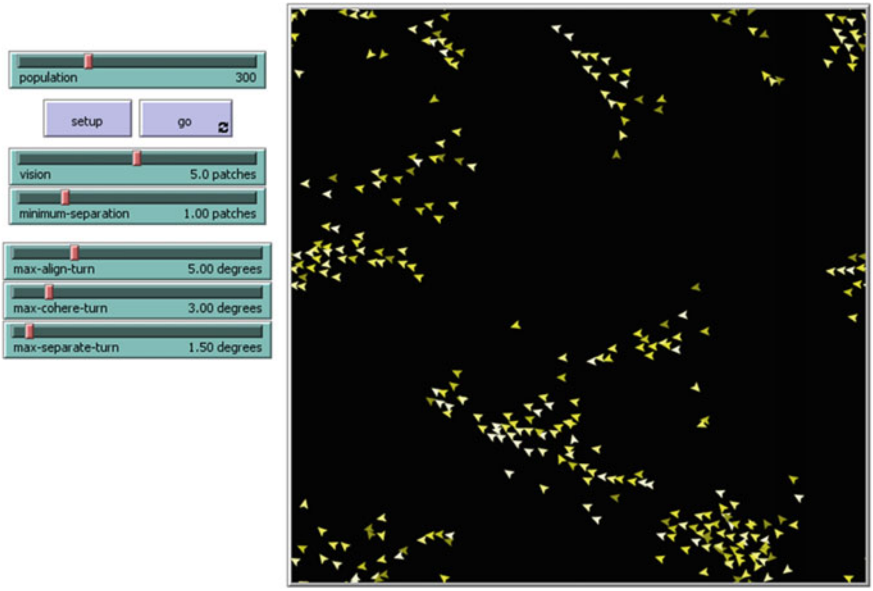
flash-length 1

flashes-to-reset 1

cycle-length 10

On Off show-dark-fireflies?

Fig. 31.9 Fireflies model. The agents synchronize flashing only on the basis of individual interactions, by the strategy to delay their next flash. Depending on the settings, the portion of the population that synchronizes grows during the simulation


population 300

vision 5.0 patches

minimum-separation 1.00 patches

max-align-turn 5.00 degrees

max-cohere-turn 3.00 degrees

max-separate-turn 1.50 degrees

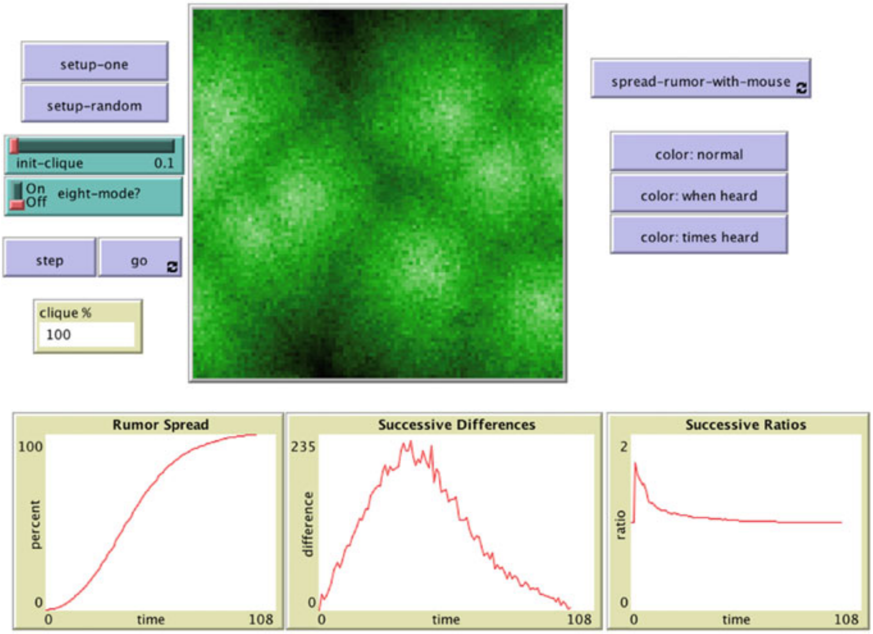
Fig. 31.10 Birds adapting their movement only to their close neighbours, distributed at random at the start of a run, end up in flocks

a delay. Where patches copy one another's attributes rather than agents, this is called diffusion. A useful property of agent-based models is that the spatial parameters can be used to symbolize something else. This allows to visualise abstract issues, e.g. position in some symbolic space. Examples:

- Society: creating and keeping walking lanes in busy places.
- Biology: path finding by ants. Netlogo: ants. Flocking by birds, herding by mammals, schooling by fish. Netlogo: flocking (Fig. 31.10).
- Physics: magnetism. Structure of crystals. density-dependent clustering of material around stars into planets.
- Engineering: nano-engineering of surfaces.

8. **Attribute patterns**. Agents adopt the same attributes, typically because deviants are weeded out. This leads to something one could call 'attribute value clustering'. If the attribute is opinion, it is called opinion dynamics. Agents copy values in binary or higher-order opinion space. This is a much-modelled phenomenon, for instance in the context of voting. Examples:

- Society: opinion dynamics. Netlogo: Rumor Mill (see Fig. 31.11).
- Biology: mimicry, where organisms evolved to look like others so that they can stay out of harm's way. E.g. cuckoo eggs. Netlogo: mimicry.
- Physics: in the physical world, this amounts to the same as clustering.
- Engineering: mimicry of artefacts such as bottles and car clutches, so that their usage or function becomes apparent.



init-clique 0.1

eight-mode? On Off

clique % 100

Fig. 31.11 Rumor Mill showing the spread of a rumor in a typical run. In the final system state, everyone has heard of the rumor, but not equally frequently, which is shown in the number of times a rumor was heard. The original sources for the rumor can be seen in the brightest areas

Output Patterns

9. **Tipping point in system output**. A positive feedback loop which leads to an irreversible termination condition after which a significantly different pattern takes over. This may involve processes such as dying or burning. In mathematical terms, this is a mechanism for bifurcation. In laymen's terms, a causal chain that was of little importance in the system now becomes dominant. It could be argued that a tipping point is not a pattern, but merely a transition from one pattern into another one. Small deviations in starting conditions may lead to changes in these effects, causing transitions in system patterns for which it is hard to predict whether, and at what moment in the simulation, they will occur. Examples:

   - Society: a behaviour dies out. Netlogo: altruism. Or a behaviour starts and supplants others, as e.g. in hypes or memes for greeting.
   - Biology: a population dies out, e.g. Cooperation (Fig. 31.7), or Wolf-sheep (see Fig. 31.1). Or a niche is created that allows a population to settle, as in ecosystem engineering.

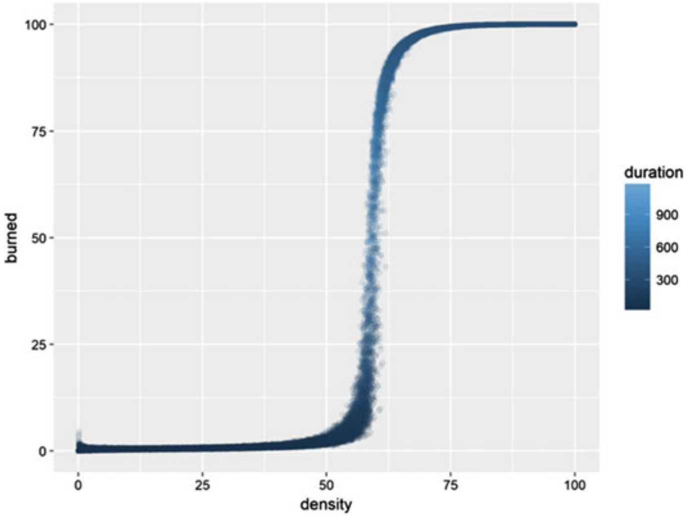
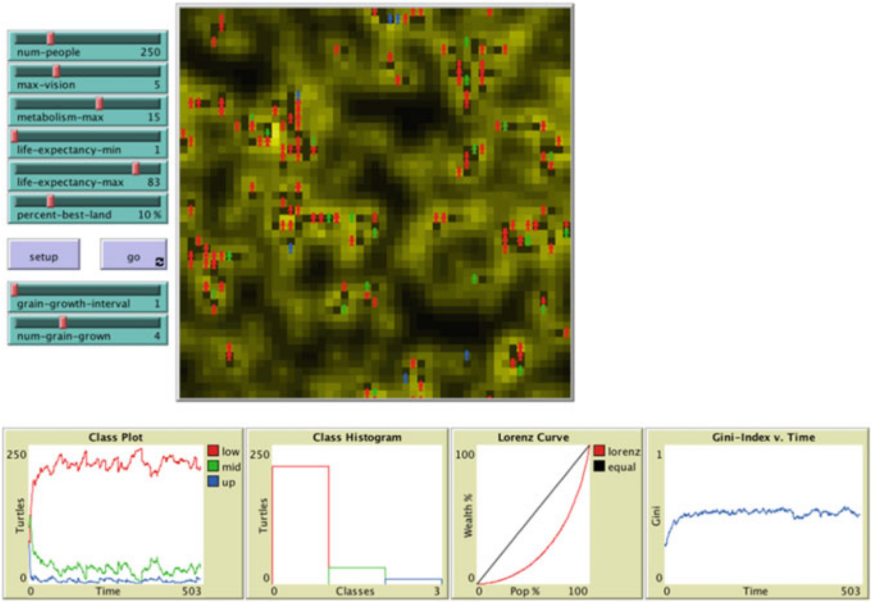


Fig. 31.12 Results from 50,000 runs with Fire model from the Netlogo model library. The fraction of the forest that burns shows a tipping point around 59% density. The results show stochasticity—the actual tipping point depends on the spatial distribution of trees. Below the tipping point the fire dies out. Above the tipping point the forest burns quickly. The fire moves slower, and hence the simulation runs take longer, around the tipping point

- Physics: a resource gets depleted. For instance, a forest could burn. In Netlogo, check the fire model. See also Fig. 31.12, which shows the tipping point for the fire model.
- Engineering: a buffer is depleted. See e.g. the Netlogo traffic models, that can be stable for a wide parameter space, until the road is full and one extra car pushes it into a traffic jam.

10. **Power law distribution**. A variable acquires a power law distribution of its frequency during the model run. Typically, this is not obvious during the model run, but clear from the distributions of output variables. Power law distributions are caused by the laws of chance affecting the probability of a variable acquiring certain values. Power laws across agents or patches are often observed. This means there is an inverse relationship between the value of some variable, and the frequency with which it occurs. This is relevant because it shows that nothing but chance is needed to account for inequality in a distribution. Power law distributions are found in innumerable phenomena. Examples:

- Society: wealth distribution in society. Netlogo: wealth distribution (Fig. 31.13).
- Biology: numerous frequency distribution in biology [16].
- Physics: earthquake frequency versus magnitude.
- Engineering: number of incoming Web links versus frequency of occurrence.



num-people 250
max-vision 5
metabolism-max 15
life-expectancy-min 1
life-expectancy-max 83
percent-best-land 10 %
grain-growth-interval 1
num-grain-grown 4

Fig. 31.13 Results from wealth distribution. Fire model from the Netlogo model library. Already with a very simple representation of the economy, a large lower class forms when turtles are assigned to the lower, middle or upper class based on their wealth in relation to the maximum wealth in the population

Conclusions and Discussion

The ability to recognize patterns in the behaviours of real-world systems is crucial in science and in policy. Yet a systematic understanding of archetypical patterns in complex systems is lacking. Agent-based modelling enable us to chart archetypical patterns and consider their causation by the mutual influences and interactions of agents such as people, politics, molecules or animals. This article presents ten archetypical patterns in three classes: resource use with unintended outcomes; contagion across time, space or symbolic space; output patterns.

Our aim in compiling these patterns is to help practitioners across various scientific fields. Humans are good at intuitively recognizing patterns, and the article builds on that strength: the ambition is to provide the concepts to describe typical patterns. This enables a discussion what simulation results one could expect, whether one already built a model or not. Scientists in many disciplines, as well as policy makers, can use them as a reference to ask themselves which of these archetypes occur in the systems they study. The typology also is an invitation to finding variants and other patterns. For instance, what about the causation of other fat-tailed distributions than power laws that are observed in reality? No doubt many other extensions are possible and indeed desirable.

This paper can be input to developments for further standardization and promoting good modelling practices [17]. It could be used when carrying out Pattern-Oriented Modelling, which measures the quality of an agent-based model by how many of the patterns from the corresponding real world system the model can produce [10].

The staggering universe of possibilities that agent-based models offer can use more structure. We envision that scholars new to computational modelling, students, policy makers, and first-time modellers are encouraged to start modelling equipped with the variety of examples of mechanisms causing patterns. The typology may train them as to what behaviour to expect, both in their own models and in those of others.

Acknowledgements The paper profited from comments by the participants of the Social Simulation Conference, September 2019 in Mainz, where an earlier version was presented. Thanks also to Tim Verwaart for clear-headed comments.

Data availability All models used to generate the simulation figures in this paper were taken from the library of Netlogo. https://ccl.northwestern.edu/netlogo/. The analysis code for the models with multiple runs is available on request.

References

1. E.J.L. Chappin, Simulating energy transitions (2011), http://chappin.com/ChappinEJL-PhDthesis.pdf
2. F. Squazzoni, J.G. Polhill, B. Edmonds, P. Ahrweiler, P. Antosz, G. Scholz, E. Chappin, M. Borit, H. Verhagen, F. Giardini, N. Gilbert, Computational models that matter during a global pandemic outbreak: a call to action. J. Artif. Soc. Soc. Simul. **23**, 10 (2020). https://doi.org/10.18564/jasss.4298
3. N. Gilbert, *Agent-Based Models* (SAGE Publications Ltd, 2007)
4. U. Wilensky, NetLogo (2010)
5. J.M. Epstein, Agent-based computational models and generative social science. Complexity **4**, 41–60 (1999)
6. B. Edmonds, Different modelling purposes, in *Simulating Social Complexity: A Handbook*, eds. by B. Edmonds, R. Meyer (Springer Publishing, Cham, 2017), pp. 39–58. https://doi.org/10.1007/978-3-319-66948-9_4
7. M.A. Janssen, J.A. Baggio, Using agent-based models to compare behavioral theories on experimental data: Application for irrigation games. J. Environ. Psychol. (2016). https://doi.org/10.1016/j.jenvp.2016.04.018
8. M. Schlüter, A. Baeza, G. Dressler, K. Frank, J. Groeneveld, W. Jager, M.A. Janssen, R.R.J. McAllister, B. Müller, K. Orach, N. Schwarz, N. Wijermans, A framework for mapping and comparing behavioural theories in models of social-ecological systems. Ecol. Econ. **131**, 21–35 (2017). https://doi.org/10.1016/j.ecolecon.2016.08.008
9. N. Gilbert, R. Conte, *Artificial Societies* (University College London Press, 1995)
10. V. Grimm, E. Revilla, U. Berger, F. Jeltsch, W.M. Mooij, H.H. Thulke, T. Weiner, J. Wiegand, D.L. DeAngelis, Pattern-oriented modeling of agent-based complex systems. Science **30**, 987–991 (2005)
11. S.F. Railsback, V. Grimm, *Agent-Based and Individual-Based Modeling: A Practical Introduction* (Princeton University Press, Princeton, 2012)
12. G. Hardin, The tragedy of the commons. Science 1243–1248 (1968)

13. E. Pruyt, *Small System Dynamics Models for Big Issues: Triple Jump towards Real-World Complexity* (2013)
14. P.M. Senge, *The Fifth Discipline: The Art & Practice of The Learning Organization* (1990)
15. S.A. Osinga, M.A. Kramer, G.J. Hofstede, A.J.M. Beulens, An agent-based management approach to smoothen the pork cycle in China, in *Emergent Results of Artificial Economics*. Lecture Notes in Economics and Mathematical Systems. https://doi.org/10.1007/978-3-642-21108-9_3
16. P.A. Marquet, R.A. Quiñones, S. Abades, F. Labra, M. Tognelli, M. Arim, M. Rivadeneira, Scaling and power-laws in ecological systems. J. Exp. Biol. **208**, 1749 (2005). https://doi.org/10.1242/jeb.01588
17. I. Nikolic, Z. Lukszo, E. Chappin, M. Warnier, J. Kwakkel, P. Bots, F. Brazier, Guide for good modelling practice in policy support: white-paper TSE7190007 commissioned by the Netherlands Enterprise Agency (RVO). Delft University of Technology (2019). https://doi.org/10.4233/uuid:cbe7a9cb-6585-4dd5-a34b-0d3507d4f188