

# S1: ODD Guidance and Checklists

**Supplement S1 to:** Grimm V, Railsback SF, Vincenot CE, Berger U, Gallagher C, DeAngelis DL, Edmonds B, Ge J, Giske J, Groeneveld J, Johnston ASA, Milles A, Nabe-Nielsen J, Polhill JG, Radchuk V, Rohwäder MS, Stillman R, Thiele JC, Ayllón D. The ODD protocol for describing agent-based and other simulation models: a second update to improve clarity, replication, and structural realism. Draft MS submitted for publication in *Journal of Artificial Societies and Social Simulation* <http://jasss.soc.surrey.ac.uk> 2019.

This document provides guidance and checklists for writing ODD descriptions of agent-based or other simulation models. It is based on the ODD version published in Grimm et al. (20XX) and earlier versions (Grimm et al. 2006, 2010; citations are provided at the end of this document). You can find further in-depth discussion of ODD and its elements in Grimm and Railsback (2005) and Railsback and Grimm (2019). While the book of Grimm and Railsback (2005) was written before ODD was established, its Chapter 5 provides the most comprehensive explanation of most of the Design concepts (Element 4), focused on ecological models but still useful for other kinds of models. The textbook of Railsback and Grimm (2019) dedicates many chapters to understanding elements of ODD and implementing them in models, using the same version of ODD as this document.

The guidance follows the ODD format, addressing all seven elements. Each element starts with summary guidance on the element’s purpose and contents. These summaries also address common mistakes and questions. (Separate guidance and checklists are provided for each of the Design concepts of Element 4.)

The checklist for each element follows the summary guidance. Each checklist is a list of topics that the ODD description should describe. You can produce an ODD description by addressing these topics completely and accurately.

We also provide examples; these are at the end of the document, but hyperlinks just after each checklist take you to the examples for that element. (For hyperlinks in Word and PDF files, ALT+left arrow and ALT+right arrow act like the “go back” and “go forward” buttons in a web browser.) These examples were selected to illustrate the range of topics and issues that can be described in each element. We consider them good—but not always perfect—examples of what should be in ODD. If you find conflicts between an example and our guidance, please follow the guidance instead of the example.

Please keep these goals of ODD in mind:

- Making model descriptions complete and accurate: it should be possible to reproduce your entire model from its written description.
- Standardizing descriptions for the benefit of users and reviewers: follow the protocol carefully, do not skip elements, and do not make ad hoc changes. It is good, not bad, to copy wording directly from this document. For example, the checklist for Element 1

includes describing “The patterns used as criteria for evaluating the model’s purpose” so we encourage you to start this description with exactly those words. (Of course, you should *not* copy text from the examples we provide from other people’s publications.)

- Providing an overview first, with details in the last three elements. This guidance and the checklists ask for a great amount of information, but ODD is intended to be hierarchical. The “Overview” elements, 1-3, should provide an accurate overview, but full details of model processes should only appear in elements 5-7.
- Encouraging modelers to think about why they made their design decisions. Describing the rationale for the design is not required by ODD but strongly recommended because it makes the model seem less arbitrary and because it encourages you to think about alternative model designs and select among them carefully. The checklists therefore include reminders to document rationale.

We ask you to precede your model description with this text: **“The model description follows the ODD (Overview, Design concepts, Details) protocol for describing individual- and agent-based models (Grimm et al. 2006), as updated by Grimm et al. (20XX).”** The citation of Grimm et al. (2006) will allow us to quantify, review, and improve the use of ODD over time. The second citation tells readers which version of ODD you follow.

The following Table of Contents is included to facilitate navigation, by using the hyperlinks in it.

1. Purpose and patterns .....	4
Guidance .....	4
Checklist .....	6
Element 1 examples .....	6
2. Entities, state variables, and scales .....	6
Guidance .....	7
Checklist .....	10
Element 2 examples .....	10
3. Process overview and scheduling .....	10
Guidance .....	11
Checklist .....	13
Element 3 examples .....	13
4. Design concepts .....	13
Basic principles .....	14
Emergence .....	15
Adaptation .....	15
Objectives .....	16
Learning .....	17
Prediction .....	17
Sensing .....	18
Interaction .....	19
Stochasticity .....	19
Collectives .....	20

Observation .....	21
5. Initialization .....	22
Checklist .....	23
Element 5 examples .....	23
6. Input data .....	23
Guidance .....	24
Checklist .....	24
Element 6 examples .....	25
7. Submodels .....	25
Guidance .....	25
Checklist .....	26
Element 7 examples .....	26
References Cited .....	26
Examples .....	28
1. Purpose and patterns .....	28
Example statements of model purpose .....	28
Example descriptions of patterns .....	30
2. Entities, state variables, and scales .....	32
Example description of entities .....	32
Example description of state variables .....	33
Example descriptions of scales .....	35
3. Process overview and scheduling .....	37
4. Design concepts .....	40
Basic principles .....	40
Emergence .....	41
Adaptation .....	42
Objectives .....	43
Learning .....	44
Prediction .....	44
Sensing .....	45
Interaction .....	46
Stochasticity .....	47
Collectives .....	47
Observation .....	48
5. Initialization .....	50
6. Input data .....	54
7. Submodels .....	56

## 1. Purpose and patterns

Every model has to start from a clear question, problem, or hypothesis; readers cannot understand your model unless they understand its purpose. Therefore, ODD starts with a concise and specific statement of the purpose(s) for which the model was developed. The [examples of Element 1 we provide below](#) categorize model purposes into types of general purpose (e.g., prediction, explanation, description, theoretical exposition, illustration, analogy, and social learning). It is useful to first identify one or more of these general types of model purpose before stating the specific purpose.

The “patterns” part of this element is new in this version of ODD. It helps clarify the model purpose by specifying the criteria you will use to decide when your model is realistic enough to be useful for its purpose. The patterns are observations, at the individual or system level, that are believed to be driven by the same processes, variables, etc. that are important for the model’s purpose. For some of the possible purposes, the model will be assumed useful only if it can reproduce the patterns. For other purposes, not reproducing the patterns can be an important result because it indicates that some mechanism is missing or inadequately represented. These patterns can be observations from the real system being modeled, obtained from data or literature. For models not based on a specific real system, the patterns are often general beliefs about how the system and its agents should behave. Including patterns in ODD is also a way to link it explicitly to “pattern-oriented modeling”, a set of strategies for designing and evaluating ABMs; this link is explained in the main text of Grimm et al. (20XX) and by Railsback and Grimm (2019).

### Guidance

**Make the purpose specific, addressing a clear question.** The purpose statement should be specific enough to enable a reader to independently judge a model’s success at achieving this purpose as well as to serve as the primary “filter” for what is and is not in the model: ideally the model should only include entities and mechanisms that are necessary to meet its purpose. If the purpose statement is only general and unspecific, and especially if it lacks patterns for evaluating the model’s usefulness, then it will be difficult to justify (and make) model design decisions.

Some ODDs state only that the model’s purpose is to “explore,” “investigate,” or “study” some system or phenomenon, which is not specific enough to assess the model or to determine what the model needs to contain. An imprecise purpose such as this is often an indication that the modeler simply assembled some mechanisms in an ABM and explored its results. Studies like this can be made more scientific by stating the purpose as a clear question such as “To test whether the mechanisms A, B, and C can explain the observed phenomena X, Y, and Z.”

**Include the higher-level purpose.** The purpose statement should also clarify the model’s higher-level purpose: whether it is to develop understanding of the system, or to make specific predictions, etc. Different high-level purposes lead to different model designs. Use the general purposes from the [examples of Element 1 we provide below](#) as a guide.

**Tie the purpose to the study’s primary results.** One way to make this statement of model purpose specific enough is to explicitly consider what point you are trying to demonstrate with the model. The statement should allow the readers to clearly judge the extent to which the model is successful. This is closely related to the primary analysis you will conduct with the model. Think about the key graph(s) you will produce in your “Results” section, where you apply the model to your main research question. The model’s purpose should include producing the relationship shown in this graph.

**Define your terms.** If you state that your model’s purpose is (for example) to “predict how the vulnerability of a community to flooding depends on public policy”, you still have not stated a clear model purpose. The term “vulnerability to flooding” could mean many things: drowning, travel delays, property damage, etc.; and “public policy” could refer to zoning, insurance, or emergency response. Be clear about exactly what inputs and results your model addresses.

**Be specific to this version of the model.** To keep the description clear and focused, do not discuss potential future modifications of the model to new purposes or patterns. (Future plans might be described instead in the Discussion section of a publication.) However, if the same model is designed for multiple purposes, those purposes should be described even if they are not addressed in the current publication.

**Do not describe the model yet.** Authors are often tempted to start describing how the model works here in the purpose statement, which is a mistake. Instead, address only the purpose here and then, in subsequent sections, you can tie the model’s design to the purpose by explaining why specific design decisions were necessary for the model’s purpose.

**Make this purpose statement independent.** Model descriptions are typically published in research articles or reports that also include, in their introduction, a statement of the study’s purpose. This ODD element should be independent of any presentation of study purpose elsewhere in the same document, for several reasons: (a) an ODD description should always be complete and understandable by itself, and (b) re-stating the model purpose as specifically as possible always helps modelers (and readers) clarify what should be in the model.

**Use qualitative but testable patterns.** Patterns useful for designing and evaluating ABMs are often general (observed at multiple locations or times) and qualitative. However, using patterns to evaluate a model requires that they be testable: you need a reproducible way to determine whether the model matches the pattern. Making patterns testable can be as simple as stating them as qualitative trends, e.g., that output X increases as variable A decreases. We generally discourage statistical definitions of patterns where the pattern is, in fact, qualitative. There are more appropriate ways of formalizing qualitative patterns, e.g. Thorngate and Edmonds (2013).

**Document the patterns.** A complete description of the patterns used in modeling needs to document why the patterns were selected: what evidence supports them, and what is their role in justifying the purpose? Documenting patterns can range from simply stating them as widespread (or your own) beliefs, to citing multiple empirical studies that observed each pattern. Thorough documentation of several patterns can require substantial text, which could conflict with keeping

this “Overview” part of ODD short. In this case, patterns can be thoroughly documented in a separate section of a report or article and summarized in the ODD description; thorough documentation of the patterns in the ODD description is not essential for it to be complete enough to make the model reproducible.

## Checklist

The ODD text for this element should describe:

- The model’s specific purpose(s).
- The patterns used as criteria for evaluating the model’s suitability for its purpose.

## Element 1 examples

[Purpose statements](#)

[Pattern descriptions](#)

## 2. Entities, state variables, and scales

This ODD element describes the things represented in the model. (Such a description can also be called an “ontology”.) It includes three closely related parts.

The first part describes the model’s entities. An entity is a distinct or separate object or actor that behaves as a unit. ABMs typically have several *kinds* of entities and one or more entities of each kind. Here, describe the kinds of entities: what the entities represent, why they are in the model, and whether the model has one, several, or many entities of each kind. The following types of entities are typical:

- Spatial units such as grid cells or GIS polygons. In spatially explicit models, these entities represent variation in conditions over space. Some models represent agent “locations” in non-geographic spaces, including networks.
- Agents or individuals. An ABM can have one or several types of agents that should each be described separately.
- Collectives (as described in Element 4, below). If the model includes agent collectives that are represented as having their own variables and behaviors, then they are also entities to describe.
- Environment. Many models include a single entity that performs functions such as describing the environment experienced by other entities and keeping track of simulated time. Such an entity can be referred to as the environment, or by using the terms used by modeling platforms for the software object that performs these functions (e.g., the “Observer” in NetLogo (Wilensky 1999), the “model”, “model swarm”, etc., in other platforms). Even if this entity does not represent anything specific in the modeled system, it can be included in ODD if useful for describing the model. Any global variables essential to the model, such as those describing the environment, should be associated with such an entity.

The second part of this element is a description of the state variables (sometimes called “attributes”) of each kind of entity. State variables of an entity are variables that characterize its current state; a state variable either distinguishes an entity from other entities of the same kind (i.e., different entities or agents have different values of the same variable), or traces how the entity changes over time (the value of an entity’s variable changes over simulated time). For example, sex is an agent state variable if there are male and female agents, but not if, as in some ecological models, only females are included; in that case, sex cannot be used to distinguish agents because they all have the same sex. Example agent state variables are weight, sex, age, information known, memory of a specific variable or type of event, wealth, social rank, location (defined by spatial coordinates or by which grid cell or network node the entity occupies), which of several categories of agent it is in (e.g., buyer or seller in a market model where agents can switch roles), and which behavioral strategies the agent is currently using. Example variables of the environment are temperature, immigration rate, or disturbance frequency (ecology), or market price or flooding risk (social sciences).

In addition to simply listing each kind of entity’s state variables, an ODD description should also describe the characteristics of each variable. These characteristics include:

- What exactly does the variable represent (including its units if it has units)?
- Is the variable dynamic (changing over time) or static (never changing in time)?
- Type: is the variable an integer, a floating-point number, a text string, a set of coordinates, a boolean (true/false) value, a probability?
- Range: can the variable have only a few discrete values (e.g., a text string representing the season with values of “spring”, “summer”, “fall”, or “winter”), is it a number with limited range (e.g., a size variable that cannot be negative), or can it have any value?

It is usually convenient to list each kind of entity’s state variables and their characteristics in a table.

The third part of this element is describing the model’s spatial and temporal scales. By “scales” we mean the model’s *extent*—the total amount of space and time represented in a simulation—and *resolution*—the shape and size of the spatial units and the length of time steps. It is important to specify what the model’s spatial units and time steps represent in reality. A simple example description of scales is: “One time step represents one year and simulations were run for 100 years. Each square grid cell represents 1 ha and the model landscape represents 1,000 x 1,000 ha; i.e., 10,000 square kilometers”. Models often use different time scales for different processes, or processes that are triggered only under certain conditions; such variations in scales should be described here.

## Guidance

**Provide the rationale for the choice of entities.** Choosing which kinds of entities to include in a model, and which to leave out, is a very fundamental and important design decision. Often this choice is not straightforward: different models of the same system and problem may contain different entities. Therefore, it is important for making your model scientific instead of arbitrary

to explain why you chose its entities. (Pattern-oriented modeling, discussed at Element 1, is a strategy for making this choice.)

**Include networks and collectives as entities if they have their own state variables and behaviors.** As discussed under ODD Element 4, ABMs can represent networks or other collectives as either (1) entities with their own state variables and behaviors or (2) emergent properties of other agents (e.g., flocks of birds that form as a result of how the birds move). Collectives of the first type should be described in this ODD element, while those of the second type should not. Do not include networks as model entities if they exist only as links among agents or can otherwise be completely described by the characteristics of their members.

**Do not confuse parameters with state variables.** Parameters are coefficients or constants used in model equations and algorithms. ODD authors often mistakenly include parameters in their description of state variables. Be careful to include as state variables only those meeting the criteria stated above: they define how an entity's state varies over time or how state varies among entities of the same type. By these criteria, the variables of a unique entity (there is only one entity of its kind, e.g., the Observer or the global environment) are state variables if they vary over time and parameters if they are static. (Models can be designed so that some parameter values vary among entities. For example, each agent may draw its own value of some parameter from a random distribution, to represent individual variation in a process. Such parameters with values that vary among entities should be treated as state variables.)

**Do not include all of an entity's variables as state variables.** The list of state variables should rarely include all the entity's variables in the computer code. State variables should be "low level" in the sense that they cannot be calculated from other state variables; do not include variables that are readily calculated from other variables. For example, if farms are represented by grid cells, the distance from a farm to a certain service center would not be a state variable because it can be calculated from the farm's and service center's locations. Also do not include variables that are only used in a particular computer implementation instead of being essential to the model; display variables are an example. One way to define state variables is: if you want (as modelers often do) to stop the model and save it in its current state, so it can be re-started in exactly the same state, what is the minimum information you must save? Those essential characteristics of each kind of entity are its state variables.

**Do not yet describe when or why state variables change or what entities do.** This element of ODD is simply to describe what is in the model, not what those things do. Save *all* discussion of what happens during simulations for later elements.

**Describe whether the model represents space, and how.** Not all ABMs are spatial; some do not represent space or the location of agents. The discussion of spatial scales should therefore start by saying whether space is represented. This discussion should also say how space is represented. First, state whether the model is one-, two-, or three-dimensional. Many ABMs are two-dimensional and represent space as a collection of discrete units ("cells" or "patches"), so the location of an agent is defined by which spatial unit it is in. These units are often square grid cells, but it is also common to use hexagonal cells and irregular polygons. Space can also be



represented as continuous, with locations described using real numbers. Both discrete and continuous space can be used in the same model. The NetLogo platform, for example, uses a discrete square grid for spatial variables but allows agent locations to use continuous space. You therefore need to define which of these possibilities are used in your model.

If a model uses toroidal space, the description of spatial scales needs to say so. “Toroidal space” (in NetLogo, “world wrapping”) approximates an infinite space by assuming that opposite sides of a rectangular model extent are connected to each other. (In physics, toroidal space is also often referred to as “periodic boundary conditions.”) An agent can move east off the right edge of the space, for example, and appear at the western edge.

**Describe whether the model represents time, and how.** While not all models include a temporal dimension, almost all ABMs do. You therefore need to describe how time is represented. Most ABMs represent time via discrete time steps: time is assumed to jump ahead one step at a time and all model processes represent what happens during these jumps. However, some models also represent some processes as occurring in continuous time: events can be executed at specific, unique times. An ODD description needs to say whether the model uses time steps and how long these steps are, and also identify any processes that are instead modeled using continuous time. (*How* processes use continuous time should be summarized in Element 3, with full detail in Element 7.)

**Describe whether the model represents any dimensions other than time and space.** It is possible for ABMs to use dimensions other than physical ones. For example, human agents could be represented as inhabiting a two-dimensional space with socioeconomic status on one axis and age on the second. If any such alternative dimensions are used, describe them as you would space.

**For abstract models without specific scales, provide an approximation or conceptual definition of scales.** Some ABMs are not built to represent a specific real system, so their spatial and temporal scales are not clearly specified. For such models, the ODD description should at least describe what the time steps and spatial units represent conceptually and approximate their scale. For example, the well-known Schelling segregation model represents households that move when unhappy but most implementations (e.g., “Segregation” in NetLogo’s models library) do not specify its scales. But assuming that households occupy urban houses and that moving takes several months lets us approximate the spatial scale—grid cells are urban lots—and the time step length—time steps are the time it takes a family to sell one house and buy another.

**When spatial or temporal scales are not fixed, describe typical values.** Some models are designed for application to multiple systems that can vary in spatial resolution or extent, and for simulation experiments that vary in duration. In these cases, the spatial and temporal scales differ among applications so cannot be stated exactly. In such cases, the ODD description should state which scales are fixed vs. variable, and provide the range of typical values for the scales that vary.

**Provide the rationales for spatial and temporal scales.** In any kind of modeling, choosing the scales is well-known as a critical design decision because model scales can have effects that are strong yet difficult to identify. This is an especially important part of your model to justify, and pattern-oriented modeling is again one strategy for doing so.

**Describe scales in a separate paragraph.** While it can be straightforward to describe spatial scales while describing the model’s spatial entities (e.g., “Space is represented as a 100 by 100 grid of square cells that each represent 1 cm<sup>2</sup>”), we recommend writing a separate paragraph that concisely states the spatial and temporal scales. These are fundamental characteristics of a model that readers will often want to find easily.

**Do not discuss model processes and behaviors.** While identifying the entities and their state variables here, it is tempting to also describe how they are initialized and the processes that cause them to change. However, those topics are instead described in elements 3 and 5.

## Checklist

Describe:

- The kinds of entities in the model and what they represent.
- Why those entities were included and, if relevant, why other entities were not included.
- The state variables of each kind of entity, usually in a table. For each state variable, describe exactly what it represents, its units, whether it is dynamic or static, its type, and its range.
- Whether the model represents time and, if so, whether time is represented via discrete time steps, as continuous, or both. If time steps are used, define their length (the temporal resolution). Describe how much time is represented in a simulation (the temporal extent).
- Whether the model represents space and, if so, whether space is represented as discrete spatial units, as continuous, or as both. If discrete spatial units are used, describe their shape and size (the spatial resolution). Describe how much total space is represented (the spatial extent).
- Whether any other dimensions are represented, and how.
- Why the spatial and temporal scales were chosen.

## Element 2 examples

[Example descriptions of entities](#)

[Example descriptions of state variables](#)

[Example descriptions of scales](#)

## 3. Process overview and scheduling

The main purpose of this element is to provide a summary overview of what the model does as it executes: what processes are executed, in what order. (The details of these processes, except for

very simple ones, are fully described in Element 7.) At the same time, the element describes the model’s scheduling—the order in which the processes are executed—in complete detail.

A model’s schedule specifies the order in which it executes each process, on each time step. We recommend describing the schedule here as a list of what simulation modelers call “actions”. An action specifies (1) which entity or entities (2) execute which process or behavior that (3) changes which state variables, and (4) the order in which the entities execute the process. The processes and behaviors, unless extremely simple, should be described only as executing a specific submodel that is fully described in Element 7. (A “submodel” is a part of the model that represents one particular processes; submodels are described independently in Element 7.) A simple example schedule is:

1. The environment executes its “update” submodel, which updates its state variables *date*, *temperature*, *rainfall*, and the value of the habitat cell state variable *food-availability*.
2. The agents each execute their “adapt” submodel, in which they choose a new location (update their state variable for which habitat cell they occupy) in response to the new environmental conditions. The agents do this in order of highest to lowest value of their state variable *body-weight*.
3. The agents execute their “grow and survive” submodel in which they (a) update their *body-weight* variable by how much they grow, and (b) either live or die. The agents execute this action in an order that is randomized each time step.
4. The display and file output are updated.

However, actions can be hierarchical: one action can include executing other actions. The first action in the above example might actually be made up of two sub-actions:

1. The environment executes its “update” submodel, which includes:
  - a. The environment increments its *date* variable by 1 day.
  - b. The environment reads in new values of its variables *temperature* and *rainfall* from the input data (Element 6).
  - c. The spatial cells each execute their “update food” submodel, which calculates a new value of the cell state variable *food-availability* from the updated environment variables. The order in which cells execute is arbitrary.

For most ABMs the Process overview and scheduling element should start with a summary of the schedule: a concise list of actions like the above example. The summary is then followed by discussion as necessary to (1) fully explain any scheduling that does not follow simple time steps, and (2) provide the rationale for the choice of processes in the model and how they are scheduled.

## Guidance

**Be complete: include everything that is executed each time step.** Remember that while this ODD element is intended as an overview of how the model executes, it is the only place in ODD that describes exactly what actions are executed and their order of execution.

**Describe the execution order.** For any action that is executed by more than one entity (e.g., any action executed by some or all agents or by the spatial units), you must specify the order in which the agents execute. Sometimes the execution order is unimportant, but often it is a very important characteristic of the model.

**Make sure the schedule summary says when state variables are updated.** The only purpose of the model's processes and submodels is to update the entity state variables as simulated time proceeds. An essential part of this element is therefore to say exactly when the state variables of each entity are changed. Use the list of state variables in Element 2 as a checklist and make sure this element describes when they are updated. If the agents update a shared variable that affects other agents (e.g., a variable that represents resource availability), say whether the model uses "asynchronous updating", in which the agents change the variable one at a time as they execute a submodel that uses the variable, or "synchronous updating", in which the variable is updated only after all agents have executed.

While Element 3 describes what happens each time step, the time step length and temporal extent of simulations should be described only in Element 2.

**Provide the rationale for the processes and scheduling, as a separate discussion.** This element is the place to explain why you chose to include the processes in the model and to exclude other processes. This is also where to justify why you scheduled the processes as you did: why are the actions in the given order (if it is not completely self-evident), and why did you chose the execution order for the actions executed by multiple entities? (Most commonly, the agents in an ABM execute any particular action in (a) arbitrary order because the action involves no interaction among agents, so order does not matter; (b) randomized order, to avoid artifacts of execution order; or (c) in order of some agent state variable, to represent a hierarchy among agents.) Scheduling can have very strong effects on model results so should be considered carefully. These explanations should be written as paragraphs that come after the schedule summary.

**Keep the overview concise.** This element is not the place to explain model processes in detail. Actions are typically described simply by saying which submodel is executed, with a cross-reference to the submodel's complete description in Element 7. Use submodel names that describe what the submodels do. However, extremely simple processes can be fully described here; for example: "All agents increment their state variable *agent-age* by 1."

**Use diagrams like flow charts if they are helpful for specific actions, but they are rarely sufficient by themselves.** Flow charts are the traditional way of describing the execution of some kinds of computer models, and can be helpful for explaining more complicated scheduling of an ABM. However, our experience has been that flow charts rarely can define an ABM's schedule completely, clearly, and accurately by themselves. Use them if helpful for some particular action, but do not rely on one flow chart or diagram to define a whole model's schedule; and accompany diagrams with sufficient text to make the process overview and schedule completely clear.

**If necessary, use pseudo-code (but not actual program statements) to clarify execution order.** ABMs sometimes include complex logic to determine which entities execute which processes when. Pseudo-code—natural language statements that include logic similar to computer code—can be useful for describing such logic. Do not, though, use actual programming statements from the model’s code in ODD; understanding such statements requires the reader to know the programming language.

**If your model includes actions scheduled as discrete events in continuous time steps, summarize how those events are scheduled.** The above guidance and example schedule apply to models in which all actions are executed on discrete time steps (as discussed for Element 2). If your model also includes actions that are scheduled to execute in continuous time, here you should supplement the process overview and schedule with a list of those actions and say in general how they are scheduled (what causes an event to be scheduled for execution and what determines when it executes). Complete details of how events are scheduled should be provided in the submodel descriptions of Element 7.

## Checklist

Provide:

- A complete but concise model schedule that defines exactly what happens each time step. The schedule should be presented as a hierarchy of “actions” that each describe which entity or entities execute which process or submodel, which state variables are updated, and (for actions executed by multiple entities such as all agents or spatial cells) the order in which the entities execute it.
- A separate discussion, if needed, that summarizes how actions executed in continuous time instead of discrete time steps are scheduled.
- A discussion providing the rationale for why the model includes the processes it does.
- A discussion providing the rationale for how the actions are scheduled. Why are the actions executed in the given order? For those actions executed by multiple entities, why did you choose the order in which the entities execute?

## Element 3 examples

### 4. Design concepts

The Design concepts element describes how 11 concepts that characterize ABMs were implemented in the model. This element is not needed to describe exactly what the model does or to replicate it, but instead is intended to place the model within a common conceptual framework and to show how its developers thought about its design.

The design concepts are intended to capture important characteristics of ABMs that are not described well by traditional description methods such as equations and flow charts. Traditional modeling literature and training do not address most of these concepts, yet most of them are important for most ABMs. Therefore, the concepts serve as a checklist of important design

decisions that should be made consciously and documented. The concepts are also useful as a way to categorize, compare, and review ABMs.

Some of the design concepts—especially Learning and Collectives—are not used in most ABMs, and few models use all the other concepts. For the concepts that your model does not use, provide a simple statement such as “Learning is not implemented” or “The model includes no collectives.”

Occasionally, modelers may identify an important concept underlying the design of their ABM that is not included in the ODD protocol. If such authors are sure that this concept cannot be incorporated in the 11 existing concepts and that it is important to understanding the design of their model, they should give it a short name, clearly announce it as a design concept not included in the ODD protocol, and present it at the end of the Design concepts element.

The Design concepts element is a particularly important place to provide the rationale for model design decisions. Your model will seem much less ad hoc and more scientific if you provide even a short summary of the reasoning and evidence that led to how these concepts were implemented. However, if providing the rationale requires extensive discussion it can be provided in one of the detailed submodel descriptions of Element 7.

For each design concept, we provide a summary of its meaning and provide a checklist of what to describe. The ODD text for this element should be at a conceptual level: instead of providing full detail for how each concept is implemented, provide a general description of its use with (if relevant) citations for literature supporting that use. Use cross-referencing to point readers to the submodel descriptions (Element 7) where full details are described.

### **Basic principles**

This concept relates the model to existing ideas, theories, hypotheses, and modeling approaches, to place the model within its larger context. These principles can occur at both the model level (e.g., does the model address a question that has been addressed with other models and methods?) and at the agent level (e.g., what theory for agent behavior does the model use, and where did this theory come from?). Describing such basic principles makes a model seem more a part of science and not made up without consideration of previous ideas.

Describe:

- The general concepts, theories, hypotheses, or modeling approaches underlying the model’s design, at both the system and agent levels.
- How these basic principles are taken into account. Are they implemented in submodels or is their scope the system level? Is the model designed to provide insights about the basic principles themselves, i.e. their scope, their usefulness in real-world scenarios, validation, or modification?
- Whether the model uses new or existing theory for the agent behaviors from which system dynamics emerge. What literature or concepts are agent behaviors based on?

### [Basic principles examples](#)

#### **Emergence**

This concept addresses a fundamental characteristic of ABMs: that system behavior can emerge from the behavior of agents and from their environment instead of being imposed by equations or rules. However, the degree to which results are emergent or imposed varies widely among models and among the different kinds of results produced by a model. This concept therefore describes which model results emerge from which mechanisms and which results instead are imposed; here, “model results” not only refers to system level dynamics, but also to the behavior of the agents.

This element should identify which model results emerge from, or are imposed by, which mechanisms and behaviors, but should not address *how* such mechanisms and behaviors work; that explanation begins with the following concept.

Describe:

- Which key model results or outputs are modeled as emerging from the adaptive decisions and behaviors of agents. These results are expected to vary in complex and perhaps unpredictable ways when particular characteristics of the agents or their environment change.
- For those emergent results, the agent behaviors and characteristics and environment variables that results emerge from.
- The model results that are modeled not as emergent but as relatively imposed by model rules. These results are relatively predictable and independent of agent behavior.
- For the imposed results, the model mechanisms or rules that impose them.
- The rationale for deciding which model results are more vs. less emergent.

### [Emergence examples](#)

#### **Adaptation**

This concept identifies the agents’ adaptive behaviors: what decisions agents make, in response to what stimuli. All such behaviors should be identified and described separately. The description should include components of behavior such as: the alternatives that agents choose among; the internal and environmental variables that affect the decision; and whether the decision is modeled as “direct objective seeking”, in which agents rank alternatives using a measure of how well each would meet some specific objective (addressed in the Objectives concept, below), or as “indirect objective seeking”, in which agents simply follow rules that reproduce observed behaviors (e.g., “go uphill 70% of the time”) that are implicitly assumed to convey success.

Many ABMs include only very simple behaviors that can be hard to think of as adaptive or even as decisions. An example is the Schelling segregation model: in it, agents simply move to a new randomly-selected location when too few of their neighbors are the same color. However, even such simple responses should be described as adaptive behaviors in ODD: the agents decide whether to stay or move by testing whether an objective measure—the percentage of neighbors having their color—is below a threshold value.

At the other extreme are ABMs that use complex evolved behaviors: each agent has an internal decision model such as an artificial neural network that is evolved in the ABM to produce useful adaptive behavior. This approach can still be described in the framework provided here: what alternatives are considered by the decision model, what its inputs are, and how its outputs are used to determine behavior. An addition step for such models is to also describe the “training conditions”: what problem were the agents given to solve in the evolution of their decision models? Such adaptive behavior models can be considered indirect objective seeking because the agents have been trained via evolution to produce behaviors successful under the training conditions.

Describe, for each adaptive behavior of the agents:

- What decision is made: what about themselves the agents are changing.
- The alternatives that agents choose from.
- The inputs that drive each decision: the internal and environmental variables that affect it.
- Whether the behavior is modeled via direct objective-seeking—evaluating some measure of its objectives for each alternative—or instead via indirect objective-seeking—causing agents to behave in a way assumed to convey success, often because it reproduces observed behaviors.
- If direct objective-seeking is used, how the objective measure is used to select which alternative to execute (e.g., whether the agent chooses the alternative with the highest objective measure value, or the first one that meets a threshold value).

### [Adaptation examples](#)

#### **Objectives**

This concept applies to adaptive behaviors that use direct objective-seeking; it defines the objective measure used to evaluate decision alternatives. (In economics, the term “utility function” is often used for an objective measure; in ecology, the term “fitness measure” is used.) If adaptive behaviors are modeled as explicitly acting to increase some measure of the individual’s success at meeting some objective, what is that measure, what does it represent, and how is it calculated? Objective measures are typically estimates of future success that depend on the decision being modeled; they can be thought of as the agent’s internal model of how its objectives will be met by the alternative being considered.



Note that agents that are part of a Collective (defined below) or larger system—members of a team, social insects, leaves of a plant, cells in a tissue—can be modeled as having objectives that serve not themselves but the larger system they belong to.

Describe, for each adaptive behavior modeled as direct objective-seeking:

- What the objective measure represents: what characteristic of agent success does it model? An example from economics is expected wealth at some future time; in an ecological model an organism might use the probability of surviving until a future time or the expected number of future offspring.
- What variables of the agent and its environment drive the objective measure.
- How the measure is calculated. If it is a simple equation or algorithm, it can be described completely here; otherwise, provide a cross-reference to the submodel in Element 7 that describes it completely. Keep in mind that some other design concepts (e.g., Prediction, Sensing) may also describe parts of the objective measure.
- The rationale behind the objective measure: why does it include the variables and processes it does?

### [Objectives examples](#)

### **Learning**

This concept refers to agents that change how they produce adaptive behavior over time as a consequence of their experience. Learning does not refer to how adaptation depends on state variables that change over time; instead, it refers to how agents change their decision-making methods (the algorithms or perhaps only the parameters of those algorithms) as a consequence of their experience. While memory can be essential to learning, not all adaptive behaviors that use memory also use learning. Few ABMs so far have included learning, even though a great deal of research and theory addresses how humans, organizations, and other organisms learn.

Describe:

- Which adaptive behaviors of agents are modeled in a way that includes learning.
- How learning is represented, especially the extent to which the representation is based on existing learning theory.
- The rationale for including (or, if relevant, excluding) learning in the adaptive behavior, and the rationale for how learning is modeled.

### [Learning examples](#)

### **Prediction**

Prediction is fundamental to successful decision-making (Budaev et al. 2019) and, often, to modeling adaptation in an ABM. (Railsback and Harvey 2020 extensively discuss the use of simple predictions in modeling difficult decisions by model agents.) Some ABMs use “explicit prediction”: the agents’ adaptive behaviors or learning are based on explicit estimates of future conditions (future values of both agent and environment state variables) and future consequences

of decisions. For these ABMs, explain how agents predict future conditions and decision consequences. What internal models of future conditions or decision consequences do agents use to make predictions for decision-making?

Models that do not include explicit prediction often include “implicit prediction”: hidden or implied assumptions about the future consequences of decisions. A classic example of implicit prediction is that following a gradient of increasing food scent will lead an agent to food.

Describe:

- How the models of adaptive behavior use either explicit or implicit prediction.
- The rationale for how prediction is represented: is the model designed to represent how the agents actually make predictions? Or is prediction modeled as it is simply because it produces useful behavior?

### [Prediction examples](#)

#### **Sensing**

This concept addresses what information agents “know” and use in their behaviors. The ability to represent how agents can have limited or only local information is a key characteristic of ABMs. Here, say which state variables of which entities an agent is assumed to “sense”, and how.

Most often, sensing is modeled by simply assuming that the agent accurately knows the values of some variables, neglecting how the agent gets those values or any uncertainty in their values. But ABMs can model the actual sensing process—how an agent gathers information about its world—when that process is important to the model’s purpose. And ABMs can represent uncertainty in sensing: you could assume, for example, that your agents base some decision on the sensed value of a neighbor’s variable, when that sensed value includes random noise. (In fact, sensing itself can be modeled as an adaptive behavior: agents can decide how much of their resources to invest to collecting information when more information supports better decisions but is costly to obtain.)

Describing sensing includes stating which variables of which entities are sensed. The description must cover what an agent knows about its own state: we need to say explicitly which of an agent’s own state variables it is assumed able to use in its behavior. When agents sense variables from other entities, such as the spatial unit they occupy or other agents, we must specify exactly how they determine which entities they sense values from. In ABMs, sensing is often assumed to be local, but can happen through networks or can even be assumed to be global.

Describe:

- What state variables, of themselves and other entities, agents are assumed to sense and use in their behaviors. Say exactly what defines or limits the range over which agents can sense information.

- How the agents are assumed to sense each such variable: are they assumed simply to know the value accurately? Or does the model represent the mechanisms of sensing, or uncertainty in sensed values?
- The rationale for sensing assumptions.

### [Sensing examples](#)

#### **Interaction**

The ability to represent interaction as local instead of global is another key characteristic of ABMs. This concept addresses which agents interact with each other and how.

We distinguish two very common kinds of interaction among agents: direct and mediated. Direct interaction is when one agent identifies one or more other agents and directly affects them, e.g. by trading with them, having some kind of contest with them, or eating them. Mediated interaction occurs when one agent affects others indirectly by producing or consuming a shared resource; competition for resources is typically modeled as a mediated interaction.

Communication is an important type of interaction in some ABMs: agents interact by sharing information. Like other kinds of interaction, communication can be either direct or mediated. An example of mediated communication is one insect depositing a pheromone that indicates to other insects that food was found.

Describe:

- The kinds of interaction among agents in the model, including whether each kind is represented as direct or mediated interaction.
- For each kind of interaction, the range (over space, time, a network, etc.) over which agents interact. What determines which agents interact with whom?
- The rationale for how interaction is modeled.

### [Interaction examples](#)

#### **Stochasticity**

Here, describe where and how stochastic processes—those driven by pseudorandom numbers—are used in the model. While some ABMs base most of their processes on random events, others can produce highly variable results with no stochasticity at all.

In general, stochastic processes are used when we want some part of a model to have variation (among entities, over time, etc.) but we do not want to model the mechanisms that cause the variability. It may be critical for a model to include how weather affects some system such as the electric power grid, but we certainly do not want to add the enormous complexity of predicting weather to the model; instead, we simply model the timing and duration of weather events as stochastic processes.

One common use of stochasticity in ABMs is to insert variability in initial conditions: when we create our agents (and other entities) at the start of a simulation (Element 5, below) we do not

want them to be identical, so we use pseudorandom number distributions to set the initial values of some state variables.

A second common use is to simplify submodels by assuming they are partly stochastic. Assuming that an agent dies if a random number between 0.0 and 1.0 is greater than its survival probability is a very common example: we do not want to represent all the detail of when agents die.

A third use of stochasticity is modeling agent behaviors in a way that causes the model agents to use different alternatives with the same frequency as real agents have been observed to. For example, a sociological model could use a stochastic process to model the age at which people marry, comparing random numbers to the marriage rates observed in real people. This approach would impose the observed marriage rates on the simulated population.

Describe:

- Which processes are modeled as stochastic, using pseudorandom number distributions to determine the outcome.
- Why stochasticity was used in each such process. Often, the reason is simply to make the process variable without having to model the causes of variability; or the reason could be to make model events or behaviors occur with a specified frequency.

### [Stochasticity examples](#)

#### **Collectives**

Collectives are aggregations of agents that affect, and are affected by, the agents. They can be an important intermediate level of organization in an ABM; examples include social groups, fish schools and bird flocks, and human networks and organizations. If the agents in a model can belong to aggregations, and those aggregations have characteristics that are different from those of agents but depend on the agents belonging to them, and the member agents are affected by the characteristics of the aggregations, then the aggregations should be described here as collectives.

Collectives can be modeled in two ways. First, they can be represented entirely as an emergent property of the agents, such as a flock of birds that assembles as a result of the flight rules given to the simulated birds. In this case, the collectives are not explicitly represented in the model: they do not have state variables or behaviors of their own. Second (and more common) is representing collectives explicitly as a type of entity in the model that does have state variables and its own behaviors. Social groups of animals or people (dog packs, political parties) have been represented this way.

Describe:

- Any collectives that are in the model.
- Whether the collectives are modeled as emerging entirely from agent behaviors, or instead as explicit entities.

- In overview, how the collectives interact with each other and the agents to drive the behaviors of the entire system. (The details of these interactions will appear in other ODD elements.)

### [Collectives examples](#)

#### **Observation**

This concept describes how information from the ABM is collected and analyzed, which can strongly affect what users understand and believe about the model. This concept can also be another place to tie the model design back to the purpose stated in Element 1: once the model is built and running, how is it used to address its purpose? This concept is not intended to document how simulation experiments and model analyses are conducted, but instead to describe how information is collected from the model for use in such analyses. Observation is important because ABMs can be complex and produce many kinds of output: it is impossible to observe and analyze everything that happens in such a model so we must explain what results we do observe.

Observation almost always includes summary statistics on the state of the agents and, perhaps, other entities such as spatial units and collectives. The ODD description needs to state how such statistics were observed: which state variables of which agents (e.g., were agents categorized?) were observed at what times, and how they were summarized. It is especially important to understand whether analyses considered only measures of central tendency (e.g., mean values of variables across agents) or also observed variability among agents, e.g., by looking at distributions of variable values across all agents.

Modelers sometimes also collect observations at the agent level, e.g., by selecting one or more agents and having them record their state over simulated time. Such observations can be useful for understanding behaviors that emerge in a model.

The ability to legitimately compare simulation results to data collected in the real world can be a major observation concern, leading some modelers to simulate, in their ABM, the data collection methods used in empirical studies. This “virtual scientist” technique (modeling the data collector; Zurell et al. 2010) strives to understand the biases and uncertainties in the empirical data by reproducing them in an ABM where unbiased and accurate observations are also possible.

Describe:

- The key outputs of the model used for analyses and how they were observed from the simulations. Such outputs may be simple and straightforward (e.g., means of agent state variables observed once per simulated week), or fairly complex (e.g., the frequency with which the simulated population went extinct within 100 simulated years, out of 1000 model runs).
- Any “virtual scientist” or other special techniques used to improve comparison of model results to empirical observations.

[Observation examples](#)

## 5. Initialization

Elements 5-7 are the “Details” part of ODD: now your goal is to provide complete detail so that your model and its results can be reproduced. This element describes exactly how the model is initialized: how all its entities are created before the simulations start. Describing initialization includes specifying how many entities of each kind are created and how their state variables are given their initial values.

### Guidance

**Describe everything required to set up the model.** Initializing state variables usually includes more than just assigning numbers to entities; it can include setting agent locations in spatial models, building networks of agents, and creating any collectives that exist at the start of a simulation. Any actions or submodels that are executed only once, at the start of a simulation, should be considered part of initialization and described here.

**Explain whether initialization is intended to be case-specific or generic.** If your model is designed to simulate only one particular case or study system, say so. In that case, this ODD element should fully describe the specific initialization data used for that case. If instead your model is designed to be generally applicable to multiple sites, then the Initialization element must be more generic. Instead of describing specific data sources, it can describe the kinds of data and information needed to apply the model to different sites.

**Explain whether initialization is always the same or differs among scenarios.** Models are used by simulating different scenarios. Scenarios can differ from each other in how the model is initialized (e.g., by simulating different numbers of agents or different environments), or by varying factors such as input data (Element 6) or parameter values that affect results only after simulations start. In the first case, with scenarios differing in initial conditions, then the goal of simulation experiments is to understand the effects of initial conditions. In the second case, the goal is to understand the effects of processes happening after initialization and modelers often try specifically to eliminate effects of initial conditions (e.g., by ignoring results until the model has run long enough for initial conditions to have minimal effect). If it is clear which of these two kinds of scenarios will be used in model analyses, say so. If you vary the initial state of the model as part of simulation experiments, it can be helpful to point out here how you do so: which state variables of which entities are varied among scenarios, and how.

**Describe data imported to create the model world.** If your model uses (for example) GIS data to define its simulated space, or reads in data used to define the initial agent populations, describe such data here. Describe where the data came from, why it was chosen, and how it was prepared. Discuss any uncertainties and limitations of the data, if relevant. (Be sure to understand the difference between initialization data and the input data that are described in Element 6.)

**Do not describe parameters and their values here.** Even though the model’s parameter values may be read in at the start of a simulation, doing so is not considered part of initialization in

ODD because parameter values are not state variables and typically do not change during a simulation. The only parameters that should be described here are any used in initialization; parameters used in the rest of the model should be described in Element 7 with the submodel that uses them.

**Do not describe how new agents or entities are created during a simulation.** If your model creates new agents or other entities as it executes, not just at the beginning, describe that creation process as a submodel, not as part of initialization. If the same submodel is used to create agents during initialization and during a simulation, it can be described in Element 7 and cited from this element.

**Provide the rationale for initialization methods.** Your model becomes more scientific if you can explain such initialization decisions as determining which state variables vary among agents (or other entities) and how. It may be appropriate in this element to present simulation experiments that illustrate the effect of alternative initialization assumptions on model results.

### Checklist

Describe:

- What entities are created upon initialization, what determines how many of each are created, and how all their state variables are given initial values.
- How the initial locations of entities are set, if relevant.
- How any initial collectives, networks, or other intermediate structures are created.
- Any site-specific data used to initialize the model: the data types, their sources, how they were processed, any known biases or uncertainties, and how they were used in initialization.
- Whether simulation experiments will typically use scenarios that differ in initialization methods; if so, say how initialization will vary.
- The rationale for key initialization methods. For example, explain why initial agents vary among each other in the way they do.

### Element 5 examples

## 6. Input data

Model dynamics are often driven in part by input data, meaning time series of either variable values or events that describe the environment or otherwise affect simulations. “Driven” means that model state variables or processes are affected by the input, but the variables or events represented by the input are not affected by the model.

One common use of input data is to represent environment variables that change over time. For example, rainfall input may affect the soil moisture variable of grid cells and, therefore, how the recruitment and growth of model trees change. Geographic or other data can be read in periodically to simulate changes in landscapes or other model “spaces”. Often, the input data are values observed in reality (e.g., from a weather station or stock market records) so that their

statistical qualities (mean, variability, temporal autocorrelation, etc.) are realistic. Alternatively, external models can be used to generate input; output from general circulation models is often used to create input “data” representing climate change scenarios, and land use change

Input data can also describe external events that affect the model during a simulation. Examples include input specifying the times at which new agents are created to represent immigration and the times at which shock events happen.

To make an ABM reproducible, we need to define any input data that drive it. To reproduce specific model results, we need to provide the input data; and to fully justify the results we need to explain where the data came from and document their uncertainties, biases, and other limitations. Even when we are not trying to make a specific set of model results reproducible, we need to state what input data drive the model. Explaining exactly what the input represents and what sources provide useful values can be important for avoiding misuse of a model.

## Guidance

**If the model does not use input data, simply say so.** In such cases, the ODD description should say “The model does not use input data to represent time-varying processes.”

**Input data do not include parameter values or data used to initialize the model.** This element does not describe all “data” read by the computer when a simulation starts. Instead, it should only describe input used as the model executes to represent change over time in some particular state variable(s).

**Define the input data completely.** Say exactly what each input variable represents, provide its units, and describe how the input was or can be obtained. If any methods are used to manipulate data from common sources into the format used by the model, describe them. If relevant, describe uncertainties, biases, or other limitations of the data.

**If your ABM is intended to represent multiple scenarios that use different input data, describe the *kind* of input it needs.** If an ABM will be used for different sites or times, so that you cannot anticipate exactly what input data will be used with it, describe exactly what kind of input it needs.

**If input was generated from another model, document how.** It should not be necessary to completely describe in the ODD how input was generated by another model, if that other model has been published. However, if a “custom” model run was used to create input just for your model, then the methods need to be described either here, in another part of the same publication, or in another document that can be cited here.

## Checklist

Describe:

- Whether the model uses any input data to represent variables that change over time or events that occur during a simulation.
- The input data: what it represents, its units, and its source.



- Any nontrivial methods used to collect or prepare the input.
- If input is from another model, how that model was used to generate the input.

### Element 6 examples

## 7. Submodels

This element is we fully describe any submodels that were cited but not fully described in Element 3 or Element 5. Element 7 should include a subsection for each such submodel.

### Guidance

**Describe each submodel completely.** The primary concern with Element 7 is being complete: fully describing each submodel’s equations, algorithms, parameters, and parameter values. Tables are often used to define parameters in one place; they should include each parameter’s name, meaning, units, type, default value, range of values analyzed in the model, and information about where the values came from, e.g. literature, unpublished data, experiments, expert opinions, etc. Figures can complement the verbal description of the submodels.

Readers should be able to exactly reproduce each submodel from the ODD description. (It is a good exercise to re-implement key submodels from the ODD description in a spreadsheet or similar platform, and then use that independent implementation to explore and calibrate the submodel as discussed below, to verify that the ODD description is complete, and to test the model software.)

**If helpful, break submodels into sub-submodels.** Submodel descriptions can be hierarchical: complex submodels are often best described as a set of sub-submodels that are each described separately. Some submodels may even justify a full ODD description of their own (“Nested ODD”, see Grimm et al. (20XX) and Supplement S4).

**Provide the rationale for submodel design.** For any process in a model, there are undoubtedly many possible submodel designs. Explaining how you arrived at your submodel design—by searching the literature, applying theory, fitting statistical relations to data, exploring alternative approaches, etc.—is very important for developing confidence in the full model.

**Include analyses of submodels.** A second way to develop confidence in submodels and the full ABM is to analyze submodels and show how they behave over all conditions that could occur during simulations. It is very common for ABMs to produce questionable output because their submodels produce unexpected results in some situations, and very difficult to understand such problems when examining the full model. It is also usually very difficult to calibrate a submodel by analyzing results of the full ABM. The way to avoid such problems is to fully implement, calibrate, test, and explore each submodel before putting it in the full ABM. These analyses should be documented in the ODD: do not just describe the submodels but show how they were calibrated and tested, and show (usually via graphs) what results they produce over all the conditions they could encounter in simulations. In addition to inspiring confidence in readers, these analyses invariably save the modeler time by identifying problems earlier. (Note that if

your ODD is part of a TRACE document, these analyses would be presented in a different section of the TRACE document; see Supplement S7.)

## Checklist

Describe:

- All of the submodels, in a separate subsection for each.
- For each submodel, its equations, algorithms, parameters, and how parameter values were selected.
- The analyses used to test, calibrate, and understand submodels by themselves.

## Element 7 examples

## References Cited

- Budaev, S., C. Jørgensen, M. Mangel, S. Eliassen and J. Giske. 2019. Modeling decision-making from the animal perspective: Bridging ecology and subjective cognition. *Frontiers in Ecology and Evolution* 7:164. doi: 10.3389/fevo.2019.00164
- Grimm, V. and S. F. Railsback. 2005. *Individual-based modeling and ecology*. Princeton University Press, Princeton, New Jersey.
- Grimm, V., U. Berger, F. Bastiansen, S. Eliassen, V. Ginot, J. Giske, J. Goss-Custard, T. Grand, S. Heinz, G. Huse, A. Huth, J. U. Jepsen, C. Jørgensen, W. M. Mooij, B. Müller, G. Pe'er, C. Piou, S. F. Railsback, A. M. Robbins, M. M. Robbins, E. Rossmannith, N. Rüger, E. Strand, S. Souissi, R. A. Stillman, R. Vabø, U. Visser, and D. L. DeAngelis. 2006. A standard protocol for describing individual-based and agent-based models. *Ecological Modelling* 198:115-296. doi:10.1016/j.ecolmodel.2006.04.023
- Grimm, V., U. Berger, D. L. DeAngelis, J. G. Polhill, J. Giske, and S. F. Railsback. 2010. The ODD protocol: a review and first update. *Ecological Modelling* 221:2760-2768. doi:10.1016/j.ecolmodel.2010.08.019.
- Railsback, S. F. 2001. Concepts from complex adaptive systems as a framework for individual-based modelling. *Ecological Modelling* 139:47-62. doi:10.1016/S0304-3800(01)00228-9
- Railsback, S. F. and V. Grimm. 2019. *Agent-based and individual-based modeling: a practical introduction, 2nd edition*. Princeton University Press, Princeton, New Jersey.
- Railsback, S. F. and B. C. Harvey. 2020. *Modeling populations of adaptive individuals*. Princeton University Press, Princeton, New Jersey.
- Thorngate, W., & Edmonds, B. (2013). Measuring simulation-observation fit: An introduction to ordinal pattern analysis. *Journal of Artificial Societies and Social Simulation: Journal of Artificial Societies and Social Simulation*, 16(2):14 (<http://jasss.soc.surrey.ac.uk/16/2/4.html>).

- Wilensky, U. (1999). NetLogo. Evanston, IL: Center for connected learning and computer-based modeling, Northwestern University.
- Zurell, D., U. Berger, J. S. Cabral, F. Jeltsch, C. N. Meynard, T. Münkemüller, N. Nehrass, J. Pagel, B. Reineking, B. Schröder, and V. Grimm. 2010. The virtual ecologist approach: simulating data and observers. *Oikos* 221:98-105. doi:10.1111/j.1600-0706.2009.18284.x

## Examples

### 1. Purpose and patterns

#### Example statements of model purpose

These examples are organized by the seven general model purposes identified by Edmonds et al. (2019. Different modelling purposes. *Journal of Artificial Societies and Social Simulation* 22:6; <http://jasss.soc.surrey.ac.uk/22/3/6.html>). The definitions of these purposes is quoted from Edmonds et al. (This does not list all possible purposes but includes the relevant kinds for the vast majority of academic model published.)

Example purpose of *prediction* (“to reliably anticipate well-defined aspects of data that is not currently known to a useful degree of accuracy *via computations using the model*”):

[Carter N, Levin SA, Barlow A, Grimm V. 2015. Modeling tiger population and territory dynamics using an agent-based approach. *Ecological Modelling* 312: 347-362]

The proximate purpose of the model is to predict the dynamics of the number, location, and size of tiger territories in response to habitat quality and tiger density...  
The ultimate purpose of the model, which will be presented in follow-up work, is to explore human-tiger interactions.

Example purpose of *explanation* (“establishing a possible causal chain from a set-up to its consequences *in terms of the mechanisms in a simulation*”):

[The Simplified Fish Cannibalism Model; unpublished description by SF Railsback of a model based on that of: DeAngelis DL, Cox DK, and Coutant CC. 1980. Cannibalism and size dispersal in young-of-the-year largemouth bass: experiment and model. *Ecological Modelling* 8:133-48.]

The purpose of this model is to illustrate a potential explanation for how small changes in the initial size distribution and growth of a cohort of cannibalistic fish can produce large differences in later size distribution, an example of how positive feedback can cause rapid divergence within a system. The mechanism driving the potential explanation is interaction among slow growth from feeding on invertebrates, rapid growth from cannibalism, and “gape limitation” which allows one fish to eat another only if the two fish are sufficiently different in size.

Example purpose of *description* (“to partially represent what is important of a specific observed case (or small set of closely related cases)”):

[Arfaoui N, Brouillat E, Saint Jean M. 2014. Policy design and technological substitution: Investigating the REACH regulation in an agent-based model. *Ecological Economics* 107: 347-365]

The purpose of our model is to understand how different configurations in the policy design of REACH affect the dynamics of eco-innovation and shape market selection and innovation. In our model we take into account supplier–user interactions because

they represent an essential element in the development of new technologies, particularly in the chemical industry. Additionally, some stylized facts that illustrate the competition between organic solvents and biosolvents in the surface treatment activity are considered. The objective is to examine to which extent different combinations of flexible and stringent instruments of REACH can lead to the development and diffusion of alternative solvents (biosolvents).

Example purpose of *theoretical exposition* (“establishing then characterising (or assessing) hypotheses about the general behaviour of a set of mechanisms (using a simulation)”):

[Polhill JG, Parker D, Brown D, Grimm V. 2008. Using the ODD protocol for describing three agent-based social simulation models of land-use change. *Journal of Artificial Societies and Social Simulation* 11: 3]

... SLUDGE is an abstract model designed for theoretical exploration and hypothesis generation. Specifically, SLUDGE was designed to extend existing analytical microeconomic theory to examine relationships between externalities, market mechanisms, and the efficiency of free-market land use patterns.

[Kahl CH, Hansen H. 2015. Simulating creativity from a systems perspective: CRESY. *Journal of Artificial Societies and Social Simulation* 18: 4]

CRESY simulates creativity as an emergent phenomenon resulting from variation, selection and retention processes (Csikszentmihalyi 1988, 1999; Ford & Kuenzi 2007; Kahl 2009; Rigney 2001). In particular, it demonstrates the effects creators and evaluators have on emerging artifact domains. It was built based on stylized facts from the domain of creativity research in psychology in order to reflect common research practice there (Figure 3).

An abstract model, CRESY was specifically designed for theoretical exploration and hypotheses generation (Dörner 1994; Esser & Troitzsch 1991; Ostrom 1988; Troitzsch 2013; Ueckert 1983; Witte 1991). It constitutes a form of research called theory-based exploration, in which models, concepts, stylized facts and observations are used as input to build a tentative model explored via computer simulation particularly in order to generate new hypotheses and recommendations as output (Bortz & Döring 2002, Ch. 6; Kahl 2012, Ch. 4; Sugden 2000). In contrast to explanatory research, the objective of exploratory research is to build theory instead of testing it. In relating this method to the model’s target, Sternberg (2006) noted creativity research is currently advancing not via its answers, but via its questions.

Example purpose of *illustration* (“to communicate or make clear an idea, theory or explanation”):

[Cardona A. 2013. Closing the group or the market? The two sides of Weber’s concept of closure and their relevance for the study of intergroup inequality. SFB 882 Working Paper Series, No. 15: From Heterogeneities to Inequalities. DFG Research Center (SFB), Bielefeld, Germany]

The purpose of the model is to illustrate how individual competition, group closure, and market closure individually or in combination are causally sufficient to produce intergroup inequality. The model does not attempt to realistically replicate any empirically observable system, but instead aims at revealing the distinct causal paths by which each of these processes affect the distribution of resources among groups.

Example purpose of *analogy* (“when processes illustrated by a simulation are used as a way of thinking about something in an informal manner”):

[Railsback SF and Grimm VG. 2017. Unpublished description of the culture dissemination model of: Axelrod R. 1997. The dissemination of culture: a model with local convergence and global polarization. *Journal of Conflict Research* 41: 203-226.]

The model’s purpose is stated [by Axelrod 1997] as being “to show the consequences of a few simple assumptions about how people (or groups) are influenced by those around them.” These assumptions are about how people learn culture from each other and, therefore, how culture spreads and how societies influence each others’ culture. In particular, the model assumes that people or societies share culture locally, and share more with others that are more similar to themselves.

Example purpose of *social learning* (“to encapsulate a shared understanding (or set of understandings) of a group of people” developed during a collaborative model-building process):

[Dumrongrojwattana P, Le Page C, Gajasen N, Trébuil G. 2011. Co-constructing an agent-based model to mediate land use conflict between herders and foresters in Northern Thailand. *Journal of Land Use Science* 6: 101-120]

This model was used for improving the researchers understanding on vegetation dynamics in relation to cattle management and reforestation effort, and facilitating the communication among village herders and NKU foresters through scenarios exploration. Moreover, stakeholders’ interactions and decisions making during G&S sessions will be observed and used for further building an ABM.

### Example descriptions of patterns

The “patterns” part of Element 1 is new in this version of ODD. Therefore, only a few ODD documents include it.

[Model of vertical migration by *Daphnia*. Section 5.5 of: Railsback, S. F. and B. C. Harvey. 2020. Modeling populations of adaptive individuals. Princeton University Press, Princeton, New Jersey.]

The purpose of this model is to explain and understand diurnal vertical migration (VM) in zooplankton and how it interacts with a life history tradeoff, the allocation of mass to reproduction or growth. The model is based explicitly on the cladoceran *Daphnia magna* (which we refer to simply as *Daphnia*).

We evaluate our model by its ability to reproduce three patterns. The first two are of primary importance because they were observed in extensive laboratory experiments by Loose and Dawidowicz (1994) and appear driven directly by the risk-growth tradeoff (Figure 5.1)...

**Pattern 1: Response of VM to predation risk.** This pattern reflects how daphnia VM in the laboratory changed as the perceived risk of predation by fish increased. In the laboratory experiments, perceived risk was increased by adding fish kairomones—chemicals produced by fish that daphnia can use to sense fish density—and perceived risk was quantified as fish concentration (fish/L). With no perceived risk, daphnia stayed near the surface throughout the diurnal cycle. At low fish concentrations, daphnia remained near the surface or migrated only to shallow depths. With high risk, daphnia stayed near the bottom. But at intermediate risk, daphnia exhibited VM, with their mean elevation in the water column low during the day and rising toward the surface at night. And at the lowest fish concentration producing VM, daphnia did not begin migration until growing to a threshold body size.

**Pattern 2. Selection of slightly shallower depths with low food.** Under high perceived risk of fish predation, a decrease in food availability resulted in daphnia selecting shallower depths. However, this change in elevation was small and occurred only in daytime.

**Pattern 3. Response of reproductive allocation to predation risk.** Under low or absent perceived fish predation risk, the daphnia in Fiksen’s model initially allocated almost all their assimilated mass to growth instead of reproduction. This allocation let them reach maximum size quickly and switch to high reproductive output. At higher risk, the daphnia allocated some mass to reproduction from the start, producing some offspring before reaching maximum size. However, at very high risk the daphnia fed very little, grew slowly, and allocated only a moderate fraction of mass to reproduction...

An example of patterns used to develop and test theory for adaptive behavior of agents:

[Woodhoopoe Model from Chapter 19 of: Railsback SF, Grimm V. 2019. *Agent-based and Individual-based Modeling: A Practical Introduction, Second Edition*. Princeton University Press, Princeton, N.J.]

We also provide three patterns that a theory for scouting forays should cause the ABM to reproduce... The first pattern is the characteristic group size distribution explained in figure 19.2. The second pattern is simply that the mean age of adult birds undertaking scouting forays is lower than the mean age of all subordinates, averaged over the entire simulation. The third pattern is that the number of forays per month is lowest in the months just before, during, and after breeding, which happens in December.

[Caption of Figure 19.2, which explains the first pattern] Characteristic woodhoopoe group size distribution... The histogram shows how many social groups there are (y-axis) for each number of birds per group (x-axis)... The number of groups with zero adults (first bar) is very low, often zero. The number of groups with only 1 adult (second bar) is low, and the numbers of groups with 2–4 adults are highest. As the number of adults increases above 4, the number of groups becomes rapidly smaller...

An example abstract model based only on simple, general patterns:

[Telemarketer Model from Chapter 13 of: Railsback SF, Grimm V. 2019. *Agent-based and Individual-based Modeling: A Practical Introduction, Second Edition*. Princeton University Press, Princeton, N.J.]

The real purpose of this model is to illustrate several kinds of interactions: how to model them and what their effects are on system behavior. While the model system is imaginary and none too serious, it does represent a common ABM scenario; a system of agents that compete for a resource and grow or fail as a result. For such a model, **we define only simple, general patterns as the criteria for its usefulness:** [that] two measures of system performance—the changes over time in the number of telemarketers in business and the median time that telemarketers stay in business—depend on how many agents there are and how they interact.

[Back to guidance](#)

## 2. Entities, state variables, and scales

We provide separate example descriptions of entities, state variables, and scales.

### Example description of entities

[Knoeri C, Nikolic I, Althaus HJ, Binder CR. 2014. *Enhancing recycling of construction materials: An agent based model with empirically based decision parameters*. *Journal of Artificial Societies and Social Simulation* 17(3)]

The following entities are included in the model: agents representing construction stakeholders (i.e. awarding authorities, engineers, architects and contractors), projects, grid cells (i.e. virtual geographical location) and the global environment representing the construction market (i.e. construction investments and materials available).

*Awarding authorities* represent private persons, companies, or public authorities awarding prime building contracts, for different purposes (e.g. personal use, economic reasons, public building requirements). *Engineers* represent the actors responsible for the static design of the concrete structure in buildings; *architects* the stakeholders designing and supervising the construction, and *contractors* the



companies providing the concrete work... In total, 5788 agents are implemented, representing the statistical distribution of construction stakeholders in the case study. *Projects* represent the individual construction projects on which these agents interact... Per year about 450 projects are executed. *Grid cells* represent virtual construction sites of 30×30m... The *observer* or *global environment* (i.e. construction market) is the only entity on the system level, defining the annual construction investments and the potential recycling aggregates supply.

**Example description of state variables**

An example describing state variables in tables. (This example uses the term “observer” for the entity that represents the global environment. The table of observer variables is therefore a table of the variables that represent the environment.):

[Railsback SF, Harvey BC, Ayllón D. InSTREAM 7 Model Description. Unpublished report.]

The **observer** is a single entity that controls the global variables and submodels. Observer state variables (Table 1) are global variables that change over time. (Static observer variables—those that do not change over simulated time—are considered parameters and are defined with the submodels they are used in.)

Table 1: Observer state variables

Variable name	Variable type and units	Meaning
<i>sim-time</i>	Date-time (date and time variables include a calendar date and time of day, e.g., 28 March 2019 14:20)	The date and time at the end of the current time step.
<i>prev-time</i>	Date-time	The date and time at the start of the current time step (and therefore at the end of the previous time step).
<i>step-length</i>	Real number; d	The length (in fraction of a day) of the current time step: the difference between <i>sim-time</i> and <i>prev-time</i> .
<i>day-length</i>	Real; d	The length of daytime (when the sun is above the horizon) for the current day; this is <i>not</i> the length of the day phase.

...

**Cells** are the lowest-level habitat entities; habitat variation within a cell is not represented (except via cell variables such as the fraction of the cell providing velocity shelter). Each cell represents a two-dimensional polygon in the horizontal plane. Most habitat characteristics are represented as cell state variables (Table 3). Because there are many cells, static cell variables are treated as state variables instead of parameters.

Table 3: Cell state variables

Variable name	Variable type and units	Meaning
(cell location; the software uses NetLogo's built-in coordinate variables)	Real, static; cm.	The X and Y coordinates of the cell's centroid.
<i>cell-area</i>	Real, static; cm <sup>2</sup>	The cell's area.
<i>cell-depth</i>	Real, dynamic; cm	The cell's current depth.
<i>cell-velocity</i>	Real, dynamic; cm/s	The current water velocity.

Example description of collectives as entities and their state variables:

[Wild Dog Model Model from Chapter 16 of: Railsback SF, Grimm V. 2019. Agent-based and Individual-based Modeling: A Practical Introduction, Second Edition. Princeton University Press, Princeton, N.J.]

The model includes three kinds of agent: dogs, dog packs, and disperser groups. Dogs have state variables for age in years, sex, the pack or disperser group they belong to (to keep track of which dogs belong to which pack), and social status. The social status of a dog can be (a) “pup”, meaning its age is less than one; (b) “yearling”, with age between 1 and 2; (c) “subordinate”, meaning age is greater than 2 but the dog is not an alpha; (d) “alpha”, meaning the dominant individual of its sex in a pack; and (e) “disperser”, meaning the dog currently belongs to a disperser group, not a pack.

Dog packs have no state variables except for a list (or, in NetLogo, an agentset) of the dogs belonging to the pack. Disperser groups have a state variable for sex (all members are of the same sex) and an agentset of member dogs.

An example that provides the rationale for selection of state variables:

[Backmann P, Grimm V, Jetschke G, Lin Y, Vos M, Baldwin IT, Van Dam NM. 2019. Delayed Chemical Defense: Timely Expulsion of Herbivores Can Reduce Competition with Neighboring Plants. *The American Naturalist* 193: 125-139]

The entities in the model are plants, larvae and patches. All state variables are given in table SO1.

#### Plants

**Rationale:** The plants represent the fast-growing tobacco plant, *Nicotiana attenuata*. These plants are native in semi-arid regions of southwestern USA. They are growing under high competition pressure in monoculture-like natural populations and defend against herbivores/pathogens etc. by producing induced defenses. Plants are also characterized by their circular "zone-of-influence" (ZOI), which are derived separately from the above- and below-ground biomass (see below).

#### Larvae

**Rationale:** The insect larvae are mobile, exponentially growing herbivores feeding on tobacco plants. During their growth they pass through five instars. At the beginning, larvae are bound to their host-plant, however, after reaching a certain weight and instar (third instar) they are able to move between plants, if necessary. Larvae choose to leave their host plant for two reasons: either the plant is nearly entirely consumed, or the defense-level (the percentage of defense compounds within the plant tissue) has reached a certain threshold. The latter is due to the fact, that larvae are affected by the defense-concentration in the plant tissue; the higher the concentration, the lower their performance, meaning that they show a decreased growth rate and an increased mortality rate. However, switching plants as well comes to a cost, more energy is needed and the probability of being predated rises significantly when being on the ground. Normally, larvae tend to chose plants in the neighbourhood as new host plants. We represented this behaviour by a dispersal kernel decreasing inversely proportional with the distance to the former host plant.

...

### Example descriptions of scales

[Railsback SF, Johnson MD. 2011. Pattern-oriented modeling of bird foraging and pest control in coffee farms. *Ecological Modelling*, 222: 3305-3319]

Scales. The model's spatial extent is a square of 200 × 200 square cells, each 5 m × 5 m in size; hence, the total area is 100 ha. This relatively fine resolution was chosen so the model can represent the effects of the small patches of trees and other habitat types that typify Jamaican coffee farms. The model's space is represented as bounded, not toroidal: birds at one edge of the space cannot jump to cells on the opposite edge.

The model runs at a 1-day time step, except that bird habitat selection and foraging is modeled at much shorter time step during daytime hours. This “foraging time step” is a parameter *forage-timestep* with value of 0.0167 h (1 min). The model does not represent night; it assumes all events occur during the day. The parameter *max-forage-hrs-per-day* specifies how many foraging hours there are per day, assumed constant at 12. Hence, birds can move and feed up to 720 times per day.

The time period modeled represents the period when CBB infest berries and when North American birds winter in Jamaica. The model runs for 151 days representing December 1 to April 30.

[Hsu SC, Weng KW, Cui Q, Rand W. 2016. Understanding the complexity of project team member selection through agent-based modeling. *International Journal of Project Management* 34: 82-93]

In our model, the temporal scale is set as days because project duration is often counted in working days. A tick in this ABM means a day. The new projects are also announced by project owners (e.g., the government, commercial entities, etc.) every day... This model sets the simulation time as 3 years because the duration of most projects in the small construction design field range from 2 weeks to 1 year, and simulating 3 years can properly encompass the typical operations of a small project-based organization.

An example using dimensions other than geographic space:

[Natalini D, Bravo G. 2014. Encouraging sustainable transport choices in American households: Results from an empirically grounded agent-based model. *Sustainability* 6: 50-69]

The agents’ position in the 3D space corresponds to their preferences for each mode of transport (i.e., the preferences act as coordinates for the dimensions x, y and z), hence there is no absolute concept of spatial scale in the model (i.e., the fact that two agents are close in the 3D space does not necessarily mean they are neighbours in the reality).

According to their position in the 3D space, the agents are clustered in neighbourhoods. The agents that belong to the same neighbourhood are connected by links, while longer links can span across neighbourhoods. This allows social influence to be taken into account when the agents make decisions. ...

An example abstract model without specific scales:

[Polhill JG, Parker D, Brown D, Grimm V. 2008. Using the ODD protocol for describing three agent-based social simulation models of land-use change. *Journal of Artificial Societies and Social Simulation* 11: 3]

Purpose: ... SLUDGE is an abstract model designed for theoretical exploration and hypothesis generation. Specifically, SLUDGE was designed to extend existing analytical microeconomic theory to examine relationships between externalities, market mechanisms, and the efficiency of free-market land use patterns.

...

Scales: The size of the cell-based Landscape is specified by the user (`Board_size`). Thus, there is no absolute concept of spatial scale in the model (either extent or resolution). A relatively coarse resolution can be implemented by using a small Landscape size and smaller Demand parameter; and a relatively finer resolution model can be implemented through a larger Landscape size and larger Demand parameter. There is also no absolute concept of temporal scale since, as stated above, the model is effectively a search mechanism for a static equilibrium. The number of time steps required to reach an equilibrium depends on the initial conditions of the model, but is usually less than 20.

[Back to guidance](#)

### 3. Process overview and scheduling

[Ayllón D, Railsback SF, Vincenzi S, Groeneveld J, Almodóvar A, Grimm V. 2016. InSTREAM-Gen: Modelling eco-evolutionary dynamics of trout populations under anthropogenic environmental change. *Ecological Modelling* 326: 36-53]

*Processes:* The model is developed to cover the whole life-cycle of a stream-dwelling trout species. It is structured in nine processes: one related to the reach and cells (update of environmental and habitat conditions), five concerning trout (habitat selection, feeding and growth, survival, reproduction, and ageing) and three performed by redds (development, survival, and hatching of eggs and genetic transmission of traits to new trout).

The reach and cells update their state variables every time step over the whole simulation; trout perform each process every time step of the simulation, but for reproduction, which only occurs during the spawning season (every time step), and angling and hooking mortality, which is restricted to the angling season (every time step); trout age every time step but change their age-class once a year (the Julian day they were born); redd's development and survival processes occur on a time-step basis since redd creation until all eggs have hatched; transmission of heritable traits occurs just when the egg hatches and the new trout is created.

*Schedule:* The simulation starts at an initial date set by the user through the input parameter *initial-date*. Environmental and habitat updates are scheduled first because subsequent trout and redd actions depend on the time step's environmental and

habitat conditions. Trout actions occur before redd's because one trout action (reproduction) can cause redd mortality via superimposition. Reproduction is the first trout action because spawning can be assumed the primary activity of a fish on the day it spawns. Spawning also affects habitat selection because 1) spawners move to the spawning habitat when a redd is created and fertilized, and 2) spawners incur on weight, and thus body condition, loss after spawning, which affects their choice of habitat. Habitat selection is the second trout action each time step because it is the way that trout adapt to the new habitat conditions; habitat selection strongly affects both growth and survival. Feeding and growth precedes survival because changes in a trout's length or condition factor affect its probability of survival. Survival has its own sub-schedule because the order in which survival probabilities for the different mortality sources are evaluated strongly affects the number of trout killed by each mortality source. Widespread, less random mortality sources are scheduled first: 1) high temperature, 2) high water velocity, 3) stranding, 4) poor condition, 5) predation by terrestrial animals, 6) predation by piscivorous fish, and 7) angling and hooking. The user has the possibility of choosing which mortality sources can kill trout during the simulation and which ones are not taken into account. Redd actions occur after cell and most trout actions because redds do not affect either habitat or fish, with the exception of creating new trout, which do not execute therefore their first actions until the day after their emergence. Redd survival is the first redd action to be executed. It includes five separate egg mortality sources that follow their own sub-schedule, from least to most random: 1) low temperature, 2) high temperature, 3) scouring, 4) dewatering, 5) superimposition. Trout emergence and genetic transmission of heritable traits is the last redd action. Since survival is scheduled before emergence, trout within redds are subject to redd mortality on the day they emerge (but not to trout mortality). Trout ageing is the last agent's executed action each time step so that both pre-existent and new created trout can increase their age. Finally, observer actions (plotting graphs and writing output files) take place at the end of the time step. All actions occur in the same predetermined order:

1. Reach updates environmental and biological conditions. Cells update depth and velocity as a function of flow, and drift/search food production rate.
2. Trout reproduce:
  - 2.1. Trout become spawners.
  - 2.2. Trout spawn and create redds.
3. Trout select habitat.
4. Trout feed and grow: update length, weight and body condition factor.
5. Trout survive or die.

6. Redds' eggs survive or die.
7. Redds' eggs develop.
8. Redds' eggs hatch, new trout are created and heritable traits are transmitted.
9. Trout age.
10. Observer plots model graphical outputs and write model output files.

An example description of a model that schedules actions as discrete events in continuous time, summarizing how those events are scheduled:

[Evers E, de Vries H, Spruijt BM, Sterck EH. 2014. The EMO-model: an agent-based model of primate social behavior regulated by two emotional dimensions, anxiety-*FEAR* and satisfaction-*LIKE*. *PLoS one* 9: e87955]

Our model is event-driven. While most social behaviors are discrete events in time, moving, resting and grooming are modeled as continuous duration behaviors. Therefore, time is modeled on a continuous scale. During a simulation run, individuals' activations are regulated by a timing regime. The general process overview and the timing regime are illustrated in Figure 1.

Each time, the agent with the lowest schedule time is activated first. Whenever an individual is activated, first all model entities update those state variables that may have increased or decreased over the time interval that has passed since the last activation of an entity (arousal, anxiety, satisfaction, *LIKE* attitudes) (see *Submodels Arousal, Anxiety, Satisfaction and LIKE Attitudes* below for more details). If the activated individual had scheduled a movement action, that action is executed (see *Submodel Movement* below). Else, ego checks the grouping criteria and employs grouping, if necessary (see *Submodel Grouping* below). If no grouping and no movement are to be performed, ego may select either a social behavior, or resting or random movement within the group. Which behavior (and which interaction partner) gets selected depends on ego's own emotional state and its arousal, as well as on its emotional attitudes towards the potential interaction partners (see *Submodel Action Selection* below). Moreover, the selected behavior may affect emotional attitudes of involved individuals. It may also affect the emotional state of ego and involved individuals (not depicted in Figure 1), as well as their schedule time (see Figure 1).

Thus, after activation, the next activation of ego, but also that of interaction partners or bystanders is scheduled anew. The exact time until an individual's next activation depends on the behavior performed, received or observed, respectively (see *Design Concept: Stochasticity* for the random drawing of the schedule times). Movement, resting and grooming are implemented as duration behaviors and are performed in bouts. Here, after starting a movement, resting or grooming bout, ego is activated after some time to decide whether the behavior is to be continued (Table S3). As

social interactions may involve (and therefore activate) other group members, they may also interrupt a grooming or resting bout. As such, whenever ego receives an attack, it is immediately activated to respond with either fleeing or a counter-attack (Table S3). Whenever ego receives a communicative signal (e.g. an aggressive signal) or observes an attack nearby, a fast reaction is required and ego is activated shortly after to select an action (Table S3).

[Back to guidance](#)

## 4. Design concepts

### Basic principles

Example (modified) from a model with basic principles at the system level:

[Wild Dog Model Model, Chapter 16 of: Railsback SF, Grimm V. 2019. *Agent-based and Individual-based Modeling: A Practical Introduction, Second Edition*. Princeton University Press, Princeton, N.J.]

This model addresses a classic problem of conservation ecology known as population viability analysis (PVA). This problem is to estimate the risk of a plant or animal population going extinct within a certain time period, and how that risk could be reduced by alternative management actions. There is an extensive literature on PVA, largely based on models that operate at the population and metapopulation levels (a metapopulation being a collection of populations linked such that individuals can disperse among them) and largely based on stochastic processes driven by probability parameters. This model differs from classical PVA models in two ways. First, it is not simply a population- (or metapopulation-) level model but instead explicitly represents lower levels of organization within the population (see Collectives, below). Second, while this model is highly stochastic (see Stochasticity, below), it is also driven by mechanistic processes and adaptive behavior within the population.

Example (modified) from a model with basic principles at both the system and agent levels:

[Railsback SF, Johnson MD. 2011. *Pattern-oriented modeling of bird foraging and pest control in coffee farms*. *Ecological Modelling*, 222: 3305-3319]

At the system level, this model addresses a well-known management problem of agricultural and ecological systems: how does land use, including both the amounts of different land uses and their spatial arrangement, affect both agricultural production and wildlife conservation? For example, are these two goals best met by separating intensive agriculture from natural reserves or by conducting agriculture in ways that also support wildlife? Such questions are especially interesting and



complex when wildlife provides services such as pest control to agriculture, as in this model.

In its bird foraging submodel (Section 2.3.7.1, below), this model poses a classic problem of optimal foraging theory: how should an individual decide whether to stay and feed in its current location and when to move on to another location? There is extensive literature on this problem, much of it based on the influential “marginal value theorem” of Charnov (1976). However, here this problem is posed in a more complex and realistic context than typically addressed in the foraging theory literature, especially because this is a population model with multiple birds depleting and competing for food. We provide insight into this problem by contrasting four alternative theories for this decision by how well they reproduce a variety observed patterns (Section 3).

Example from a model based on an original idea instead of previous models and ideas:

[From the description by Railsback and Grimm (2019; Chapter 22) of: Wilensky, U. 1997. NetLogo Segregation model. <http://ccl.northwestern.edu/netlogo/models/Segregation>. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL. This model is based on the segregation model of Schelling (Schelling, T. C. 1971. Dynamic models of segregation. *Journal of Mathematical Sociology* 1, 143–86.)]

The basic principle of the Segregation model is the idea that it was designed to illustrate: that strong individual behaviors may not be necessary to produce striking system patterns. Does the presence of strong segregation mean that individual households are highly intolerant, or can such strong patterns emerge in part from the system’s structure? Understanding this principle can be critical for developing policies to address social issues such as segregation.

[Back to guidance](#)

### **Emergence**

Example from a simple demonstration model:

[From the description by Railsback and Grimm (2019; Chapter 22) of the NetLogo Segregation model.]

The key outcomes of the model are segregation patterns—especially, how strongly segregated the entire system is; these outcomes emerge from how households respond to unlike neighbors by moving and, to a lesser extent, from the density of households (the percentage of locations that are occupied by a household).

Example from a more complex model with a mix of emergent and imposed results:

[Railsback SF, Johnson MD. 2011. Pattern-oriented modeling of bird foraging and pest control in coffee farms. *Ecological Modelling*, 222: 3305-3319]

The model's primary results—pest insect infestation rates in two types of coffee production—and intermediate results such as bird abundance and spatial distributions of birds emerge from the amount and spatial distribution of the six habitat types, the seasonal abundance of pest insects, the number of birds, and the foraging behavior of birds. Of the nine characteristic patterns used to design and test the model, patterns 3, 4, 5, 7, and 8 especially emerge from bird foraging behavior and habitat characteristics. These patterns are believed driven by the same mechanisms that produce the primary results, so these patterns must also be emergent to make them useful for testing the model's suitability for its primary purpose. Patterns 1, 2, 6, and 9 are at least partially imposed by model rules and parameters. These four patterns are direct outcomes of lower level processes, especially related to the pest insect's life cycle, from which other patterns emerge; making these patterns emerge from lower-level mechanisms would make the model much more complex and was determined unnecessary for its purpose.

[Back to guidance](#)

### **Adaptation**

A simple example of direct objective seeking as satisficing:

[From the description by Railsback and Grimm (2019; Chapter 22) of the NetLogo Segregation model.]

The model households have one adaptive behavior: deciding whether or not to move to another location and, if so, selecting a new location. The decision of whether to move is modeled as direct objective seeking: a household moves if its objective measure (Objectives, below) is below a “tolerance threshold”. This approach can be considered a type of satisficing—making a decision to achieve an acceptable level of the objective instead of maximizing it. The tolerance threshold is a model parameter named %-similar-wanted. When a household does decide to move, it selects a new location from those not currently occupied using a stochastic process described below (the “move” submodel).

An example of direct objective seeking as optimization:

[Business Investor model. Chapter 10 of: Railsback SF, Grimm V. 2019. Agent-based and Individual-based Modeling: A Practical Introduction, Second Edition. Princeton University Press, Princeton, N.J.]

The adaptive behavior of investor agents is repositioning: the decision of which neighboring business to move to (or whether to stay put), considering the profit and risk of these alternatives. Each time step, investors can reposition to any unoccupied one of their adjacent patches or retain their current position. Which patch to select is modeled as direct objective seeking using optimization: investors select the patch that provides the highest value of the objective measure explained below.

An example of indirect objective seeking:

[Wild Dog Model Model. Chapter 16 of Railsback and Grimm (2019).]

The primary adaptive behavior of dog packs is dispersal: the pack decides whether its subordinate dogs leave the pack in hopes of establishing a new pack. This behavior is modeled using indirect fitness seeking: stochastic rules cause the pack to select each alternative with a frequency similar to the frequency observed in real wild dogs, with the implicit assumption that these observed frequencies occur because they make dogs relatively successful at becoming alpha adults and, therefore, being able to reproduce. Two simple rules are used to model the dispersal behavior. If a pack has only one subordinate dog of its sex, it decides randomly whether this dog “disperses” by forming a new disperser group entity; the probability of dispersing is 0.5. If the pack has more than one subordinate dog of the same sex, those dogs always form a disperser group and leave the pack.”

[Back to guidance](#)

### Objectives

A simple objective:

[From the description by Railsback and Grimm (2019; Chapter 22) of the NetLogo Segregation model.]

The objective measure used by model households to decide whether to move is the percentage of adjacent households that have the same color as the household making the decision. “Adjacent” households are any households on the eight surrounding patches.

A more complex objective measure:

[Business Investor model. Chapter 10 of Railsback and Grimm (2019).]

Investors rate business alternatives by an objective measure (utility measure, in economics) that represents their expected future wealth at the end of a time horizon (T, a number of future years) if they buy and operate the business. This expected future wealth is a function of the investor’s current wealth and the profit and failure risk offered by the patch:  $U = (W + TP) (1 - F)^T$  where U is the expected utility for the patch, W is the investor’s current wealth, P is the annual profit of the patch, and F is the probability per year of the business failing. The term (W + TP) estimates investor wealth at the end of the time horizon if no failures occur. The term (1 - F)<sup>T</sup> is the probability of not having a failure over the time horizon; it reduces utility more as failure risk increases. (Economists might expect to use a utility measure such as present value that includes a discount rate to reduce the value of future profit. We ignore discounting to keep this model simple.)

[Back to guidance](#)

## Learning

Example models that use established theories of human learning:

[Gimona A., Polhill, JG. 2011. Exploring robustness of biodiversity policy with a coupled metacommunity and agent-based model, *Journal of Land Use Science* 6: 175-193]

The adaptive behavior of land managers—deciding what land use to select—is modeled using an approach that includes learning. This submodel (fully explained below at “Land use selection submodel”) is based on “case-based reasoning” theory (Aamodt and Plaza 1994). This approach assumes decisions are based on memory of previous decisions in similar cases and their outcomes. The land manager agents use their own memories of previous decisions, but if their memory contains no similar cases they can use the memory of neighboring land managers. As a land manager executes more land use decisions it therefore develops a base of information that affects future decisions.

[Angourakis A, Santos JI, Galán JM, Balbo AL. 2015. Food for all: an agent-based model to explore the emergence and implications of cooperation for food storage. *Environmental Archaeology* 20: 349-363]

The reinforcement learning mechanism implemented in the model is an adaptation of Bush and Mosteller’s model of reinforcement learning (Bush and Mosteller 1955). At the end of a generation (*learning-generation* constant) each agent considers the times without shortage (*n-non-shortage-ticks* variable) and compares it with the aspiration-threshold ( $T_h$ ). Taking as reference the most frequent action she has undertaken in the former generation, the agent updates her strategy, i.e. the probability to cooperate ( $P_i$ ), following a simple reinforcement learning rule: do it more often, if it led to more steady satisfaction (i.e. fulfilling the aspiration), otherwise try more often the alternative action. The strategy updating takes place in three steps: ...

[Back to guidance](#)

## Prediction

An example of implicit prediction:

[From the description by Railsback and Grimm (2019; Chapter 22) of the NetLogo Segregation model.]

The adaptive behavior of households is based on the implicit prediction that moving when the objective measure is below the tolerance threshold is likely to eventually result in the household occupying a location where the tolerance threshold is met permanently.

Examples of explicit prediction:

[Business Investor model. Chapter 10 of Railsback and Grimm (2019).]

The utility measure estimates utility over a time horizon by using the explicit prediction that profit  $P$  and failure risk  $F$  will remain constant over the time horizon. This assumption is accurate in this model because the patches'  $P$  and  $F$  values are static.

[Model of vertical migration by *Daphnia*. Section 5.5 of: Railsback, S. F. and B. C. Harvey. 2020. Modeling populations of adaptive individuals. Princeton University Press, Princeton, New Jersey.]

In this version we modify the algorithm for predicting future reproduction and survival to include differences between day and night. We do this simply by keeping track of whether each of the future hours evaluated (Step 2 of the algorithm described in Section 5.4) is in day or night, and using memory of habitat conditions during the previous night (or day) to predict future conditions. A *Daphnia* deciding what to do during a daytime hour uses the prediction that future daytime hours until the time horizon will all have the same growth and survival conditions as the patch it is evaluating, and that future nighttime hours will have conditions it “remembers” from the patch and  $\alpha$  value the *Daphnia* actually used during the most recent night hour. At night, a *Daphnia* predicts future day conditions from the patch and  $\alpha$  value it used in the most recent day hour.

[Back to guidance](#)

## Sensing

An example of simple sensing assumptions:

[Railsback SF, Johnson MD. 2011. Pattern-oriented modeling of bird foraging and pest control in coffee farms. *Ecological Modelling*, 222: 3305-3319]

Birds are assumed able to perfectly sense the current availability of both types of prey in cells within a specified radius of their current cell. This radius is constant over time and space and among birds; its value is the model parameter *forage-radius* (meters).

An example of sensing with feedbacks from adaptive behavior:

[Railsback S. F., Harvey BC, Jackson SK, Lamberson RH. 2009. InSTREAM: the individual-based stream trout research and environmental assessment model. PSW-GTR-218, USDA Forest Service, Pacific Southwest Research Station, Albany, California]

Model trout are assumed able to sense habitat conditions and select habitat from among cells within a radius that increases with their size. Specifically, a trout can sense and potentially move to all cells whose centroids are less than *sensing-distance*

from the centroid of the trout's current cell; the value of *sensing-distance* is equal to  $bL^a$  where  $L$  is the trout's current length (cm) and  $a$  and  $b$  are model parameters with standard values of 50 and 2.0. Therefore, sensing is a mechanism for positive feedback of growth: trout that grow more rapidly can sense and potentially occupy a wider range of habitat, which may allow them to grow more rapidly (or, should they choose, to use safer habitat).

[Back to guidance](#)

### Interaction

An example of mediated (indirect) interaction:

[Jensen T, Holtz G, Chappin ÉJ. 2015. Agent-based assessment framework for behavior-changing feedback devices: Spreading of devices and heating behavior. *Technological Forecasting and Social Change* 98: 105-119.]

Interaction occurs through social influence between household agents sharing relationship links. For technology diffusion, adopting peers increases the probability (where this equals not already 1) a household adopts feedback technology. For behavior diffusion, a household agent gradually adapts its energy consumption behavior according to the mean behavior of its peers.

An example of direct interaction:

[Lobo EP, Delic NC, Richardson A, Raviraj V, Halliday GM, Di Girolamo N, ... , Lyons JG. 2016. Self-organized centripetal movement of corneal epithelium in the absence of external cues. *Nature communications* 7: 12388]

Cells respond to the local configurations of other cells by moving in response to a pressure gradient or, in the case of LESC, replicating in response to the death of a neighbor.

An example with both direct and mediated interactions:

[Telemarketer model, Chapter 13 of Railsback and Grimm (2019).]

There are two kinds of interaction in this model: between telemarketers and potential customers, and among the telemarketers. The telemarketers interact directly with potential customers by communicating to find out whether the customers will buy, and then by making the customers change their state to indicate that they bought during the current time step. The telemarketers' interactions with each other are mediated by the resource they compete for: customers. When the territories of telemarketers overlap, customers of one telemarketer are no longer available as potential sales for other telemarketers. Both of these kinds of interaction are local because telemarketers are assumed able to call only customers within a radius of their location. However, this radius increases with telemarketer size (see the

“Telemarketer sales” submodel, below) and interaction approaches global for large telemarketers.

[Back to guidance](#)

### Stochasticity

A simple use of stochasticity:

[From the description by Railsback and Grimm (2019; Chapter 22) of the NetLogo Segregation model.]

*Stochasticity* is used in two ways. First, the model is initialized stochastically in such a way that (a) the total number of households, (b) whether each location is occupied initially, (c) the color of each household, and (d) the number of households of each color, are all stochastic (*Initialization*, below). These initialization methods are stochastic so that the model can be assumed unsegregated at the start of a simulation, and so that each model run produces different results. Second, when a household decides to move, its choice of new location (the “move” submodel) is stochastic (but not completely random). The new location of households when they move is stochastic because modeling the details of movement is unnecessary for this model.

An example of more complex use of stochasticity, including stochastically generated landscapes:

[Railsback SF, Johnson MD. 2011. Pattern-oriented modeling of bird foraging and pest control in coffee farms. *Ecological Modelling*, 222: 3305-3319]

Stochasticity is used in initializing the model (*Initialization*, below) to create irregular clumps of each habitat type; this stochastic process allows creation of multiple landscapes that have the same over-all habitat characteristics (e.g., area of each habitat type, number and mean size of habitat clumps), which are used as replicates in analysis of model responses to habitat availability scenarios. Stochasticity is also used in initialization to assign each bird a home cell, and to impose variability among cells in coffee pest infestation rates (see the “pest infestation rate” submodel, below). During a simulation, the main uses of stochasticity are to (1) avoid a feeding hierarchy by randomizing the order in which birds execute their foraging trait each foraging time step, and (2) decide whether birds die if they did not make their daily intake requirement (see the “mortality” submodel, below). In bird foraging, several cells may offer exactly the same, highest, food intake rate; in such cases, the bird chooses one of these cells randomly.

[Back to guidance](#)

### Collectives

An example of collectives explicitly modeled, as social groups:

[Wild Dog model, Chapter 16 of Railsback and Grimm (2019).]

This model includes two kinds of collectives, groups of wild dogs that strongly affect the individual dogs and are strongly affected by the individuals. The collectives are represented as specific kinds of model entity with their own state variables and behaviors. These entities are called packs and disperser groups; the entities and their state variables are defined above at *Entities, State Variables, and Scales*. These collectives are included in the model because wild dogs have many cooperative behaviors and make decisions critical to the population's abundance and persistence in ways that depend on their group's state. It is much easier to model cooperative behavior and collective decisions as behaviors of collective entities than as behaviors of the individual dogs.

[Back to guidance](#)

### Observation

Example of a model with results compared to observed patterns, including a special procedure to generate observations more comparable to field data:

[[Railsback SF, Johnson MD. 2011. Pattern-oriented modeling of bird foraging and pest control in coffee farms. \*Ecological Modelling\*, 222: 3305-3319](#)]

Graphical output on the model interface shows the habitat type of each cell, via cell color. Bird locations are also shown. The model randomly selects one bird per day and displays a trace of its movement during the day, so foraging patterns can be observed. Summary statistics on the bird population, pest insects, and other food insects, are provided via plots on the interface and output files. Virtual surveys are a special kind of observation designed for comparison of model results to field surveys of bird populations; they simulate the methods used by field biologists to estimate bird densities so that the results are affected by the same biases as the real surveys. This observation technique is fully described below (the virtual survey submodel).

Example from a model that requires complex analysis to meet its purpose:

[[Wild Dog model, Chapter 16 of Railsback and Grimm \(2019\).](#)]

The model's purpose is to study how potential management alternatives affect the persistence of dog populations, and one measure of a simulated population's persistence is the probability that it goes extinct within a certain number of years. This probability of extinction can be estimated as the fraction of replicate simulations in which no dogs are alive at the end. How long these simulations are, and how many replicates are executed, are arbitrary observation decisions. Here, the dog population's persistence is estimated as the fraction of 500 replicate simulations in which there are no dogs alive after 100 years.

Example of a model that requires observation of individual variables to estimate complex statistics at the population level:



[Baggio RA, Araujo SBL, Ayllón D, Boeger WA. 2018. Dams cause genetic homogenization in populations of fish that present homing behavior: Evidence from a demogenetic individual-based model. *Ecological Modelling* 384: 209-220]

The location (spawning area) and genome of each individual are recorded every generation. The changes in the genetic structure are assessed through the global, within-subnetwork (upstream or downstream to the dam) and paired (between two subpopulations) *Fst* index (Weir and Cockerham, 1984, Weir, 1996) before and after the addition of the dam. Global *Fst* is the component of the genetic variation due to differences among all subpopulations. Within-subnetwork *Fst* is the component of the genetic variation due to differences among the subpopulations within each subnetwork. The pairwise *Fst* is the inter-population level genetic variation between two subpopulations. The *Fst* is estimated as

$$Fst = \frac{\sum_{v=1}^g \sigma_{P_v}^2}{\sum_{v=1}^g \sigma_{P_v}^2 + \sum_{v=1}^g \sigma_{G_v}^2} \quad (1)$$

where  $\sigma_{P_v}^2$  is the variance between subpopulations and  $\sigma_{G_v}^2$  is the variance within subpopulations for the locus  $v$ . The  $\sigma_{G_v}^2$  is estimated as the mean square for within populations (*MSG*)

$$\sigma_{G_v}^2 = MSG = \frac{1}{\sum_i (n_i - 1)} \sum_i n_i p_{A_i} (1 - p_{A_i}) \quad (2)$$

where  $p_{A_i}$  is the frequency of the allele A of the locus  $v$  in the  $i^{\text{th}}$  subpopulation, and  $n_i$  is its subpopulation size. The  $\sigma_{P_v}^2$  is estimated as

$$\sigma_{P_v}^2 = \frac{1}{n_c} (MSP - MSG) \quad (3)$$

where *MSP* is the mean square for between populations and  $n_c$  is composed by the average sample sizes and their variance. Then, the  $n_c$  is estimated as

$$n_c = \frac{1}{(r-1)} \left( \sum_{i=1}^r n_i - \frac{\sum_i n_i^2}{\sum_i n_i} \right) \quad (4)$$

where  $r$  is the number of populations and  $n_i$  is the population size of the population  $i$ . The *MSP* is estimated as

$$MSP = \frac{1}{r-1} \sum_i n_i (p_{A_i} - p_A)^2 \quad (5)$$

where  $p_A$  is the frequency of the allele A of the locus  $v$  in the metapopulation (global *Fst*), subnetwork (within-subnetwork *Fst*) and among two populations (pairwise *Fst*). The *Fst* statistics were chosen because it is one of the most widespread indexes used to test genetic differentiation among populations. *Fst* varies from zero to one: zero values represent absence of genetic differentiation, and non-zero values represent some level of genetic differentiation (see Freeland and Petersen, 2011 for details).

Example of a model that uses a “virtual scientist” technique to improve comparison of model results to empirical observations:

[Pais MP, Cabral HN. 2017. Fish behaviour effects on the accuracy and precision of underwater visual census surveys. A virtual ecologist approach using an individual-based model. *Ecological Modelling* 346: 58-69]

The model fits into what has been generically called a virtual ecologist approach (Zurell et al., 2010), where the diver (i.e. the “virtual ecologist”) observes and records virtual fish, while performing a simulated sampling method. In the end of the sampling method, the counts made by the diver are divided by the sample area to calculate estimated densities. In order to calculate the accuracy of the sampling procedure, as well as the bias due to non-instantaneous observation, the real density of every species in the model environment is registered, as well as the density of each species inside the sampling area at the start of the sampling method.

For every species, the model outputs the real density, the estimated density (using the virtual ecologist) and bias due to non-instantaneous sampling. Bias is calculated as the difference between the estimated (non-instantaneous) density and the initial density inside the sample area, divided by the initial density (Ward-Paige et al., 2010). Therefore, a bias of 1 means that the diver counted on average one more fish per square meter for each fish it would have counted if sampling was instantaneous.

[Back to guidance](#)

## 5. Initialization

An example description of how agents are created and their state variables initialized:

[Ayllón D, Railsback SF, Vincenzi S, Groeneveld J, Almodóvar A, Grimm V. 2016. InSTREAM-Gen: Modelling eco-evolutionary dynamics of trout populations under anthropogenic environmental change. *Ecological Modelling* 326: 36-53]

Each individual’s state variable (sex, age, length, and genotypic and phenotypic values of heritable traits) is initialized by drawing from probability distributions describing their variability. Length and genotypic values of heritable traits are truncated at 4 standard deviations from the center of the probability distributions. Length of 0+ trout cannot be lower than a minimum user-defined value *fishMinNewLength*. Trout weight is calculated as a function of length:

$$fishWeight = fishWeightParamA \times (fishLength)^{fishWeightParamB}$$

and condition factor is subsequently calculated as a function of body length and weight. The condition factor variable used in the model (*fishCondition*) can be considered the fraction of “healthy” weight a fish is, given its length (approach

adopted from Van Winkle et al. 1996). The value of *fishCondition* is 1.0 when a fish has a “healthy” weight for its length, according to the length-weight relationship. Trout maximum sustainable swimming speed is a function of the fish’s length and water temperature. It is modelled as a two-term function, where the first term represents how it varies linearly with fish length, while the second modifies maximum swimming speed with a non-linear function of temperature:

$$\begin{aligned}
 \text{fishMaxSwimSpeed [cm s}^{-1}\text{]} &= [\text{fishMaxSwimParamA} \times \text{fishLength} + \\
 &\text{fishMaxSwimParamB}] \\
 &\times [\text{fishMaxSwimParamC} \times (\text{temp})^2 + \text{fishMaxSwimParamD} \times \text{temp} + \\
 &\text{fishMaxSwimParamE}]
 \end{aligned}$$

Status is set to “alive”. Maturity status is set to either “mature” or “non-mature” depending on whether trout’s initial length is over or under the phenotypic value of the length maturity threshold (*fishSpawnMinLength*). The *spawnedThisSeason?* variable is set to “NO”.

Each trout’s location is assigned stochastically while avoiding extremely risky habitat. The model limits the random distribution of trout to cells where the trout are not immediately at high risk of mortality due to high velocity or stranding. Therefore, each trout is located in a random wetted cell (*cellDepth* > 0) with a ratio of cell velocity to the trout’s maximum swimming speed (*cellVelocity* / *fishMaxSwimSpeed*) lower than the parameter *mortFishVelocityV9*, the value at which the probability of surviving high velocity mortality equals 0.9 (see 2.7 Submodels Section 5).

An example description of data imported to create the model world, including the source of the input map and how input was prepared:

[Baum T, Nendel C, Jacomet S, Colobran M, Ebersbach R. 2016. “Slash and burn” or “weed and manure”? A modelling approach to explore hypotheses of late Neolithic crop cultivation in pre-alpine wetland sites. *Vegetation History and Archaeobotany* 25: 611-627]

Landscape initialization: The initial state of the model at the time t=0 is a hypothetical model environment without previous human influence consisting of 200x200 cells of 25x25 metres. The cells are described by various variables provided via input files that are generated using Geographical Information Systems (GIS). In the presented model version, the following data were used, but other spatial data from other landscapes can be applied as well.

The variable soil type may either be Luvisol, Gleysol, Peat bog or Water. The soil data has been simplified and adapted (using data from LGRB 2013) to better resemble conditions before the onset of wide-spread human influence using simple rules. All sub-varieties of a soil type have been treated as being identical (e.g.

Anmoorgley, Quellgley, Auengley have are all treated as Gley); Colluvia have been assigned the value of the dominant neighboring value of Gley or Luvisol, as they were only formed through more intense human influence than in the period in question; the same procedure has been applied to areas with the value no data. The result is a much generalized, but still valid soil map that roughly retrodicts conditions of the mid-Holocene without strong human influence.

The variable travelcost is taken from an externally calculated GIS raster dataset ([www.lgl-bw.de](http://www.lgl-bw.de)). The variable natural soil fertility is dependent on the soil type and has the relative values of 1 for Luvisol, 0.75 for Gleysol and 0.4 for Peat Bog. These values are assumptions.

The natural potential ecosystem of the cell is dependent on the soiltype: Mixed Beech Forest on Luvisol, Alder-Ash-Forest on Gleysol, and Alder Carr on Peat Bog. This very generalized allocation is based upon information given by Rösch et al. (2014, p. 124), Kerig and Lechterbeck (2004, p.24) and Lang (1990). The cells covered with Beech Forest or Alder-Ash-Forest are grouped to stands consisting of 1-16 cells, equaling 0.06-1 ha, sharing the same forest development phase (rejuvenation-, initial-, optimal-, terminal, and decay- phase). This roughly reflects conditions for natural forest dynamics as described e.g. in Emborg et al. (2000), Remmert (1991) or in Leibundgut (1959, 1982, 1993). The area covered by the forest development phases is determined by the ratio of 8/8/28/36/20 as described in Mayer and Neumann (1981). With an assumed duration of one complete cycle of forest development from rejuvenation to decay phase lasting 500 years, this translates into a duration of the phases of 40, 40, 140, 180 and 100 years in this simulation. For Alder Carr, no such dynamics is assumed.

An example description of how initial collectives, networks, or other intermediate structures are created:

[[Fedriani JM, Wiegand T, Ayllón D, Palomares F, Suárez-Esteban A, Grimm V. 2018. Assisting seed dispersers to restore oldfields: An individual-based model of the interactions among badgers, foxes and Iberian pear trees. Journal of Applied Ecology 55: 600-611](#)]

... Then, the dispersers spatial groups are defined, their associated home range being delineated as an ellipse, which is characterized by four parameters: its centre, the most eastern point in the major axis, angle, and eccentricity (Table S2). These parameters are specific for each spatial group, but do not differ between disperser species. All patches in and within the ellipse belong to the home range of the spatial group.

Finally, the dispersers are initialized by setting the total number of dispersers (*number-dispersers* global parameter) and the proportion of foxes vs. badgers by means of the parameter *dispersers-proportion*. Each disperser is randomly given an initial home patch (called den), typically (80% and 90%, for badgers and foxes,

respectively) within the scrubland, on a buffer area (width defined through the parameter *buffer-den*) along the border between the scrubland and the oldfield. This area of den-initialization includes also a small "peninsula" of the scrubland southwestern portion which is very close (but not directly bordering) with the oldfield. In the remaining 20 and 10% of the cases, the initial home patch of each disperser (the den) is assigned randomly, with each patch having an equal probability of selection (irrespective of whether another disperser has been assigned to the same patch), but limited to open habitats (oldfield, marshes, juncus and prairie) within the area defined by the two spatial groups (northern or southern). In this way, foxes and badgers are usually initialized in the scrubland, which is in accordance with our telemetry data. These rules are based on empirical data indicating that most dispersers locate their dens in such portion of the study area. The initial position defines to which spatial group the disperser belongs (see Sensing concept), so that dispersers with an initial y-coordinate over the parameter *limit-social-groups* belong to the northern spatial group, and viceversa. Since the home ranges of the spatial groups overlap, this limit is defined as the average value of the y-coordinates of the most southern patch of the buffer area of the northern spatial group and the most northern buffer-patch of the southern group.

An example description of scenarios defined by different initial conditions:

[Cerdá M, Tracy M, Ahern J, Galea S. 2014. Addressing population health and health inequalities: the role of fundamental causes. *American journal of public health* 104(S4): S609-S619]

At initialization, the agent population consisted of 4000 individuals aged 18 years and older with sociodemographic characteristics assigned to match distributions of the adult population in New York City according to the 2000 US Census (for a table specifying the default values of the initialization parameters of the model, see Appendix 6, available as a supplement to this article at <http://www.ajph.org>). We divided the grid representing the physical space into 16 neighborhoods, and each cell in the grid could be occupied by only 1 agent. Assignment of agent locations and determination of neighborhood boundaries depended on the objectives of the model run with respect to racial and economic residential segregation. We implemented 2 residential segregation scenarios in different model runs: complete segregation of agents by race and income and no racial or economic segregation. To achieve complete segregation by race and income, each of the 16 neighborhoods in the model corresponded to 1 of the 16 possible combinations of race/ethnicity and household income, with only agents assigned that particular combination of race and income residing in that neighborhood. For example, all White agents with an income of \$75 000 or more lived in 1 neighborhood, whereas Black agents with an income of \$75 000 or more lived in another neighborhood. The size of the neighborhood was proportionate to the size of the race/income combination in the total population, with the width of the neighborhood on the grid reflecting the racial distribution and the

height of the neighborhood on the grid reflecting the income distribution (for a snapshot of the grid, see Appendix 7, available as a supplement to this article at <http://www.ajph.org>). By contrast, for populations with no racial or economic segregation, we randomly assigned agents to a location on the grid, which was divided into 16 neighborhoods of equal size, producing neighborhoods that each had residents with a mix of race and income characteristics. ...

An example of providing the rationale for initialization methods:

[Lobo EP, Delic NC, Richardson A, Raviraj V, Halliday GM, Di Girolamo N, ... , Lyons JG. 2016. Self-organized centripetal movement of corneal epithelium in the absence of external cues. *Nature communications* 7: 12388]

Cell positions are initially chosen so that the Voronoi diagram generated is a regular hexagonal grid (the distance between the positions of neighboring cells is a constant,  $s$ , known as the idealized cell diameter). All ESCs are given a random age (uniformly distributed from 0 to their lifespan), and a unique cell lineage identification code. Initially, all TACs are not assigned a lineage identification code. TACs derived from ESCs after initialization inherit the lineage identification code of their parent cell. Cell types are chosen such that LESC have only two adjacent LESC neighbors, and that TACs that have less proliferative capacity are placed towards the center of the cornea. This placement has been validated through simulations where the proliferative capacity of TACs had no spatial relationship, and the simulations were run until homeostasis was reached. At that point, TACs near the center had less proliferative capacity than cells near the limbus.

[Back to guidance](#)

## 6. Input data

An example description of how field data were obtained and processed to develop model input:

[Frank BM, Baret PV. 2013. Simulating brown trout demogenetics in a river/nursery brook system: The individual-based model DemGenTrout. *Ecological Modelling* 248: 184-202]

The following time series of observations were available for the Lesse River/Chicheron Brook hydrological system: water temperatures ( $^{\circ}\text{C}$ ) in both streams, flow rates ( $\text{m}^3 \text{s}^{-1}$ ) in stream L, and water heights (m) in stream C. Water temperatures in both streams were automatically measured and recorded every hour with loggers, from years 2004 to 2009. For the Lesse River, daily flow rates were available from a nearby permanent gauging station located in Daverdisse ( $Q_D$ ) for the years 2004–2009. We used the following equation to compute the flow at the study site ( $Q_L$ ), in which  $A_L$  and  $A_D$  are the watershed areas of the study site and the Daverdisse station, respectively, expressed in square kilometres:  $Q_L = (A_L/A_D) \times Q_D =$

$(195/302) \times Q_D = 0.65 \times Q_D$ . For the Chicheron Brook, flow rates  $Q$  were estimated from water heights  $H$  recorded at a rectangular weir, on days when the trapping facility was checked over the years 2004–2009. We used the following calibration equation provided by E. Dupont (Earth and Life Institute, Croix du Sud 2 Box L7.05.14, 1348 Louvainla-Neuve, Belgium, personal communication, 2009):  $Q = \sqrt{2g} \times 0.2 \times \sqrt{H^3}$  for  $H < 0.22$  m and  $Q = \sqrt{2g} \times (0.2 \times \sqrt{H^3} + 0.368 \times \sqrt{(H-0.22)^3})$  for  $H \geq 0.22$  m, where  $g$  is the gravitational constant.

Each time series was transformed into a weekly time series using the *xts* R package (Ryan and Irich, 2010). Holt-Winters method implemented in R (R Development Core Team, 2011) was used to apply a multiplicative seasonal model to each weekly time series in order to predict 29 years of data (2010–2039) from the 6 years of field data (2004–2009), and then use all 35 years of data as input in the DemGenTrout model. This exponential smoothing technique separates the time series  $t$  into three components, a level  $L_t$ , a trend  $T_t$ , and a seasonal index  $S_t$ :  $X_t = (L_t + T_t \times t) S_t$ , where  $X_t$  is an observation on the time series,  $L_t = \alpha \times (X_t / S_{t-p}) + (1 - \alpha) \times (L_{t-1} + T_{t-1})$ ,  $T_t = \beta \times (L_t - L_{t-1}) + (1 - \beta) \times T_{t-1}$ , and  $S_t = \gamma \times (X_t / L_t) + (1 - \gamma) \times S_{t-p}$  ( $\alpha$ ,  $\beta$  and  $\gamma$  are the smoothing parameters of level, trend, and seasonal index). The prediction function on  $h$  periods is given by:  $X_{t+h} = (L_t + h \times T_t) \times S_{t-p+1+(h-1) \bmod p} + e_t$ , where  $p$  is the period length equal to 52 weeks, and  $e_t$  is the random error component. For water temperatures in streams C and L, parameters were:  $\alpha = 0.9076$  and  $0.6360$ ,  $\beta = 0.0000$  and  $0.0000$ ,  $\gamma = 1.0000$  and  $0.0033$ ,  $e_t$  was a normal random number with mean and standard deviation equal to  $0.00$  and  $0.44$ , and to  $0.00$  and  $1.22$ , respectively. For flow rates in streams C and L, parameters were:  $\alpha = 0.0734$  and  $0.0010$ ,  $\beta = 0.0072$  and  $0.0000$ ,  $\gamma = 0.3670$  and  $0.0000$ ,  $e_t$  was a log-normal random number with mean and standard deviation equal to  $1.00$  and  $3.00$ , and to  $0.01$  and  $0.03$ , respectively.

An example description of input that drives model events (creation of new agents) and updates the environment during a simulation:

[Chudzinska M, Ayllón D, Madsen J., Nabe-Nielsen J. 2016. Discriminating between possible foraging decisions using pattern-oriented modelling: The case of pink-footed geese in Mid-Norway during their spring migration. *Ecological Modelling* 320: 299-315]

Two input data files from external sources are used in the model: number of new super geese arriving to the model and distribution of habitat types within the modelled area. The number of arriving geese was assessed based on counts of geese on the roost sites in the years 2005-2012 and new geese arrive to the model as described in Section 2.2.3 and Appendix A. New habitat map is loaded to the model at the beginning of each period as described in Section 2.2.3 and Appendix A. The position, size and habitat type of each field was derived from regular habitat mapping conducted during field work in 2013 (Chudzinska et al., 2015) (Appendix A).

[Back to guidance](#)

## 7. Submodels

Example descriptions of submodels and their rationales:

[Piou C, Prévost E. 2012. A demo-genetic individual-based model for Atlantic salmon populations: Model structure, parameterization and sensitivity. *Ecological Modelling* 231: 37-52]

Smoltification: SM6. Individual parr need to undergo a smoltification process to be able to leave the river. It is well documented that this physiological, behavioural and morphological process is a function of individual growth (Thorpe, 1977) and that it is initiated much earlier than actually visible (see reviews of McCormick and Saunders, 1987; McCormick et al., 1998). It was therefore opted for an initiation time at the beginning of winter with an actual seaward migration at the beginning of summer (as in Thorpe et al., 1998). We used the probabilistic reaction norm depending on body length and adjusted by Buoro et al. (2010) on the Scorff to simulate this process. An individual parr  $j$  would become a future smolt in autumn following the relation:

$$\text{logit}(P(Sm_j = 1)) = aSm \times (L_j - LmidSm)$$

where  $aSm$  and  $LmidSm$  (Table 2) were population parameters and  $Sm_j$  was the binary indicator state variable. The inter-individual variation for an identical  $L$  was considered in the probabilistic aspect of this implementation.

[Lieder M, Asif FM., Rashid A. 2017. Towards Circular Economy implementation: An agent-based simulation approach for business model changes. *Autonomous Agents and Multi-Agent Systems* 31: 1377-1402]

Social network structure: By applying the concept of homophily the probability of information exchange between two individuals depends on the similarity of their socio-demographic factors [10]. In comparison to a distance-based network this brings the advantage that inter-agent connections can occur throughout the entire continuous modeling space since geographic distance is not the only measure for (dis)similarity. Furthermore, a network structure based on homophily appears more realistic as well as more practical since socio-demographic data are already applied in Sect. 4.6.1 (behavior curves) in order to characterize customer agents. So, the network creation is based on the social (dis)similarity  $s$  and defined as Euclidean distance  $D$  in all dimensions  $d$  (number of socio-demographic factors) [9]. Formally, the (dis)similarity between two customer agents  $x$  and  $y$  can be expressed as follows:

...

An example description of analyses that test, calibrate, and understand a submodel by itself:



[Railsback SF, Harvey BC, Ayllón D. Contingent tradeoff decisions with feedbacks in cyclical environments: testing alternative theories. *Submitted paper*]

#### VIII.S Terrestrial predation trout mortality

...

##### VIII.S.7 Parameter values and submodel exploration

The standard values for terrestrial predation parameters (which we recommend reviewing and adapting for all inSTREAM 7 applications) are in Table 11. To understand this complex submodel, we present plots showing its response to two key variables at a time. In the following figures, the standard parameter values (for large rivers) are used and trout are not hiding. Variables not explicitly varied in the plots have these standard values: turbidity = 5 NTU, sunlight-irradiance = 200 W/m<sup>2</sup>, reach-shading = 0.9, trout-length = 10 cm, trout-activity = “drift” or “search”, cell-velocity = 20 cm/s, and cell-escape-dist = 200 cm. The value of cell-light is in all cases calculated from sunlight-irradiance, reach-shading, turbidity, and cell-depth (Sect. VIII.J) so has a standard value of 118 W/m<sup>2</sup>.

An exploration of how terrestrial predation survival varies with depth, fish length, and light (Figure 16) illustrates how survival is relatively high for small trout but depends sharply on depth for trout above ~5 cm length. Decreasing light levels, from day to twilight to night, increase the survival probability but do not alter the shape of the relation between depth and survival.

For an adult trout, depth and distance to escape cover have effects similar in magnitude (Figure 17). Predation risk can be reduced sharply by using habitat that is either deeper or closer to escape cover.

[Back to guidance](#)