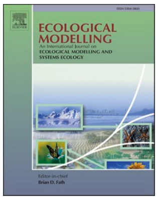
Ecological Modelling 501 (2025) 110967

Contents lists available at ScienceDirect

Ecological Modelling

journal homepage: www.elsevier.com/locate/ecolmodel



Using the ODD protocol and NetLogo to replicate agent-based models

Volker Grimm[a,b,*], Uta Berger[c], Justin M. Calabrese[a,d,e], Ainara Cortés-Avizanda[f,a], Jordi Ferrer[g], Mathias Franz[h,i], Jürgen Groeneveld[a], Florian Hartig[j], Oliver Jakoby[k], Roger Jovani[l], Stephanie Kramer-Schadt[m,n], Tamara Münkemüller[o], Cyril Piou[p], L.S. Premo[q], Sandro Pütz[r,a], Thomas Quintaine[s], Christine Rademacher[t], Nadja Rüger[u,v,w], Amelie Schmolke[k], Jan C. Thiele[x,y], Julia Touza[z], Steven F. Railsback[aa]

[a] *Helmholtz Centre for Environmental Research – UFZ, Department of Ecological Modelling, Leipzig, Germany*
[b] *University of Potsdam, Plant Ecology and Nature Conservation, Potsdam, Germany*
[c] *TUD Dresden University of Technology, Department of Enviornmental Sciences, Dresden, Germany*
[d] *CASUS - Center for Advanced Systems Understanding, Helmholtz-Zentrum Dresden-Rossendorf e.V. (HZDR), Earth System Science, Görlitz, Germany*
[e] *Department of Biology, University of Maryland, College Park, MD, USA*
[f] *EBD-CSIC, Estación Biológica de Doñana, Department of Conservation Biology and Global Change, Sevilla, Spain*
[g] *Lycée International de Barcelona Bon Soleil, Barcelona, Spain*
[h] *Max Planck Institute for Evolutionary Anthropology, Department of Primatology, Leipzig, Germany*
[i] *Freie Universität Berlin, Institute of Biology, Berlin, Germany*
[j] *University of Regensburg, Faculty of Biology and Pre-Clinical Medicine, Group of Theoretical Ecology, Regensburg, Germany*
[k] *RIFCON GmbH, Effect Modelling and Statistics, Hirschberg, Germany*
[l] *EBD-CSIC, Estación Biológica de Doñana, Department of Ecology and Evolution, Sevilla, Spain*
[m] *Leibniz Institute for Zoo and Wildlife Research IZW, Department of Ecological Dynamics, Berlin, Germany*
[n] *Technische Universität Berlin, Institute of Ecology, Berlin, Germany*
[o] *Univ. Grenoble Alpes, Univ. Savoie Mont Blanc, CNRS, LECA, Grenoble, France*
[p] *CIRAD, UMR CBGP, INRAE, IRD, Institut Agro Montpellier, Univ. Montpellier, Montferrier sur Lez, France*
[q] *Washington State University, Department of Anthropology, Pullman, WA, USA*
[r] *Federal Agency for Nature Conservation – BfN, Department of Terrestrial Monitoring, Leipzig, Germany*
[s] *Université de Reims Champagne-Ardenne, Centre de Recherche et Formation en Eco-éthologie, Boult-aux-Bois, France*
[t] *Technische Hochschule Nürnberg Georg Simon Ohm, Nürnberg, Germany*
[u] *German Centre for Integrative Biodiversity Research (iDiv) Halle-Jena-Leipzig, Leipzig, Germany*
[v] *Leipzig University, Institute of Empirical Economic Research, Department of Economics, Leipzig, Germany*
[w] *Smithsonian Tropical Research Institute, Balboa, Ancón, Panama*
[x] *Georg-August-University of Goettingen, Department of Ecoinformatics, Biometrics and Forest Growth, Goettingen, Germany*
[y] *Bremische Bürgerschaft, Bremen, Germany*
[z] *University of York, Department of Environment and Geography and York Environmental Sustainability Institute, York, UK*
[aa] *Lang Railsback and Associates, Arcata, CA, USA*

ARTICLE INFO

Keywords:
Model replication
Standards
ODD protocol
NetLogo
Agent-based model
Theory development

ABSTRACT

Replicating existing models and their key results not only adds credibility to the original work, it also allows modellers to start model development from an existing approach rather than from scratch. New theory can then be developed by changing the assumptions or scenarios tested, or by carrying out more in-depth analysis of the model. However, model replication can be challenging if the original model description is incomplete or ambiguous. Here we show that the use of standards can facilitate and speed up replication: the ODD protocol for describing models, and NetLogo, an easy-to-learn but powerful software platform and language for implementing agent-based models. To demonstrate the benefits of this approach, we conducted a replication experiment on 18 agent-based models from different disciplines. The researchers doing the replications had no or little previous experience using ODD and NetLogo. Their task was to rewrite the original model description using ODD, implement the model in NetLogo and try to replicate at least one exemplary main result. They were also asked to produce, if time allowed, some initial new results with the replicated model, and to record the total time spent on the replication exercise. Replication was successful for 15 out of 18 models. The time taken varied between 2 and

---

* Corresponding author.
E-mail address: volker.grimm@ufz.de (V. Grimm).

https://doi.org/10.1016/j.ecolmodel.2024.110967
Received 29 August 2024; Received in revised form 30 October 2024; Accepted 25 November 2024

12 days, with an average of 5 days. ODD helped to systematically scan the original model description, while NetLogo proved easy and quick to learn, but difficult to debug when implementation problems arose. Although most of the models replicated were relatively simple, we conclude that even for more complex models it can be useful to use ODD and NetLogo for replication, at least for developing a prototype to help decide how to proceed with the replicated model. Overall, the use of both, standard approaches such as ODD and easy to learn but powerful software such as NetLogo, can promote coherence and efficiency within and between different models and modelling communities. Imagine if all modellers spoke ODD and NetLogo as a common language or *lingua franca*.

1. Introduction

Agent-based simulation models are widely used in ecology, environmental and social sciences. Those models are typically spatially explicit and include autonomous and potentially unique decision-making of the central modelling units (the agents) in response to their environment. When these models are used to understand and predict the dynamics of real systems, they often behave in case-specific ways so that insights gained are difficult to generalize to other systems or environmental settings. Thus, a challenge in the field is to identify general principles and theories that explain how the behaviour of agents controls the emergence of structure and dynamics of real systems.

The aim of gaining general insights from agent-based models may be limited by the complexity and diversity of these systems (Antosz et al., 2023; Grimm et al., 2024; Wijermans et al., 2023), but we hold that there are also aspects of modelling practice and culture that hinders the development of more general, but still predictive, models: developing models more or less from scratch seems to be more valued by both individual modellers and the modelling community than starting from existing models.

There are several reasons for this. Modellers often learn more about the system in question during model development than from the final model. It has therefore been recommended to document not only the final model, but also the key analyses and decisions made during the modelling process (Ayllón et al., 2021; Grimm et al., 2014; Schmolke et al., 2010). Moreover, just using an existing model, for example by changing some settings and parameters, carries the risk of using the model beyond its original purpose and blindly trusting its results rather than trying to understand how they are generated.

Here, we suggest a middle way between developing models from scratch and just using existing models: starting new projects with model replication (Thiele and Grimm, 2015). In science, replication generally means repeating an existing study, based on the original description of the materials and methods used, and obtaining consistent results. Replicability is considered a cornerstone of the scientific method. It lends credibility because, in principle, conclusions become independent of the person, place and time involved in a study (Zhang and Robinson, 2021). This is relevant in ecology for both modelling and empirical studies (Filazzola and Cahill Jr, 2021; Fraser et al., 2020; Parker et al., 2016).

Reproducing the results of an original study, however, achieves more than just confirming the original results (Axtell et al., 1996; Gürcan et al., 2023; Zhong and Kim, 2010). Reproducing a study also re-establishes the methods of the original experiment , so that they can now be used to modify the original question and the settings of the experiment, and thereby generate new insights. In modelling, for example, one could perform a Robustness Analysis, that is, try to systematically simplify the model to learn which model assumptions were essential to reproduce the observed patterns (Grimm and Berger, 2016). Or, the original results can be generalised by exploring additional scenarios, and additional, more in-depth simulation experiments can improve our understanding of the dynamics of the system under study (Carney and Davies, 2020; Hauke et al., 2020). The replication exercise can also reveal weaknesses in the assumptions, or implementation, of the original model (Edmonds and Hales, 2003), and replication can also be part of the attempt to facilitate the alignment and comparison of different models that address the same question but give different answers (Banitz et al., 2022; Hales et al., 2003). Such multi-model tests, now often referred to as 'model intercomparison projects' (MIP, e.g. Warszawski et al. (2014)) are challenging but increasingly needed because we need to understand why the answers are different before our models can robustly support decision making.

A further direct benefit of replicating a modelling study is that “the modeller starts from existing hypotheses and submodels and therefore enters the ‚modelling cycle‘ (Grimm and Railsback, 2005) of iterative model formulation, implementation, simulation, and analysis much more quickly. This leaves more time for the critical but time-consuming task of in-depth model analysis, which is often performed only superficially because too much time had been spent developing the first conceptual model.“ (Thiele and Grimm, 2015, p. 692). However, replicating models can be difficult if the model has not been described in detail, and even if it has, there may be subtle details in the implementation of the model in a particular programming language that are not documented (Grimm et al., 2020b).

To standardize model descriptions, the ODD protocol is increasingly used as a standard format for describing agent-based models, and is now also being used for mathematical models (Grimm et al., 2006, 2010, 2020b; Polhill et al., 2008); as for standardized ABM descriptions including human behaviour, see Müller et al. (2014). The ODD protocol recommends a certain structure of the model description and provides a list of items that should be explained. The latest update of ODD (Grimm et al., 2020b) also recommends that the written model description should be explicitly linked to the code that implements the model, to further reduce ambiguity.

Another challenge in model replication is the variety of programming languages, operating systems, and software platforms used to implement models. Ideally, results should be independent of the details of a model’s implementation, but exact matching is rarely possible, often due to the use of different random number generators or different floating-point algorithms. However, even achieving similar output distributions, or at least the same qualitative relationships between model input and output (Axtell et al., 1996), may fail.

Ideally, all models would be described in a format such as ODD and implemented in a generic software platform with a programming language that is easy to learn, understand, and execute. While this ideal seems unrealistic, we will show here that we are not far from it. We will show that translating ad hoc model descriptions into the ODD format and then implementing them using NetLogo[1] (Wilensky, 1999; Wilensky and Rand, 2015), a generic software platform and language originally developed for teaching and for agent-based modelling, is straightforward to learn and can make model replication easier and faster than is generally believed.

Model replication can, as we will show, be easy and fast even for those unfamiliar with ODD and NetLogo. Using ODD and NetLogo, we successfully reimplemented 15 of 18 models and replicated the original results. Below we list these models and summarise our experiences, including the time spent for the replication exercise, challenges and

---

[1] https://ccl.northwestern.edu/netlogo/

recommendations. We will suggest using ODD as a standard for model replication and NetLogo as a versatile and efficient tool for at least prototyping a replicated model.

Our results show that describing a model according to a standard such as the ODD protocol not only makes models transparent, but can also support replication, even for modellers previously unfamiliar with ODD and NetLogo and even if the original model did not use the ODD description. Replication should be even more likely to succeed if the original description follows ODD and is well linked to the accessible code. To our knowledge, this is the first study that tested the potential of ODD to support replication. We argue that standards like ODD thus contribute to Good Modelling Practice, which is increasingly needed and discussed as models are increasingly needed to support environmental and ecological decision making (Jakeman et al., 2024). Moreover, certain software platforms can actually become quasi-standards, as they are widely used within a certain community or domain, such as R for statistical analyses and modelling, Python for data science, and NetLogo for agent-based modelling.

2. Methods

The replications presented here were conducted in 2009 to demonstrate the standardising power of the ODD protocol, which had been published only three years earlier in 2006. However, as ODD was quickly adopted by developers of individual and agent-based models (Grimm et al., 2010), there no longer seemed to be a strong need to advocate the use of the ODD protocol. There were also doubts as to whether NetLogo should really be recommended for wider use in science, given its origins as educational software and the perception that it would not be suitable for more complex models. Also, the importance and benefits of replication were not yet fully articulated or widely accepted.

Now, 15 years later, the situation has changed. The ODD protocol is widely used (Vincenot 2018), and NetLogo has become the model language of choice in the social sciences, and is increasingly being used in ecology and other disciplines (Abar et al., 2017; Railsback et al., 2017; Vincenot, 2018). NetLogo has been shown competent for complex simulation models (e.g. BEEHAVE, Becher et al. 2014) because its built-in primitives are highly efficient, certain programming techniques/tricks can speed it up substantially (Railsback et al. 2017), and it can easily be run on High Performance Computing clusters (e.g., Ayllón et al. (2016), Gallagher et al. (2021)). One advantage of NetLogo is the ease with which models can be adopted and used by others, provided the model is well documented. For example, the honey bee colony model BEEHAVE (Becher et al., 2014) has been used in >25 publications, about two-thirds of them without the involvement of any of the BEEHAVE developers (Groeneveld et al., 2024).

2.1. ODD and netLogo

ODD model descriptions consist of seven elements (Table 1).

The first three elements provide an overview, so that the reader does not need to read the entire model description to get an idea of what the model is and does; they also allow the reader to zoom in on the details of most interest. The fourth element describes important concepts underlying the design of the model; this acts as a checklist so that important design decisions are made explicit. The last three elements provide details of how the model is initiated, the data inputs representing external drivers, and the sub-models used to implement the model's processes.

NetLogo is both a programming language specifically designed for implementing agent- models (including cellular automaton-like models without mobile agents), and a software platform for programming and running models. It is the most widely used "offspring" of Logo, which was originally developed to introduce students or school children to modelling and simulation (Papert, 1985); it therefore still uses the "turtle" label for agents by default, but this can be easily changed to represent any autonomous agent.

Table 1
Partial correspondence of the elements of the ODD protocol and the software platform and programming language NetLogo.

	ODD Protocol	NetLogo elements
Overview	1. Purpose and patterns	Information tab
	2. Entities, state variables and scales	Breeds, turtles-own, patches-own, globals
	3. Process overview and scheduling	„go“ procedure
Desing concepts	4. Design concepts - Emergence - Adaptation - Objectives - Learning - Prediction - Sensing - Interaction - Stochastiticity - Collectives - Observation	Information tab, primitives
		Elements of the Graphical User Interface
Details	5. Initialization	„setup“ procedure
	6. Input data	File input
	7. Submodels	Procedures and reporters

NetLogo is free and open-source software released under a GNU General Public License (GPL), and runs on all major operating systems. It is well documented and comes with training material and a large library of example models, and it has been continuously maintained and developed since its first release in 1999. NetLogo is implemented in Java and Scala, it can be run interactively with an easy-to-build Graphical User Interface, and it allows execution in headless mode on high-performance computing (HPC). NetLogo has been an unusually stable platform: most of our replicated models were implemented in version 4.0.5 of NetLogo in 2009 but we were able to run them all in the current version (6.4.0) with only a few simple update steps and almost no changes in code. NetLogo was specifically designed to facilitate programming agent-models. It includes powerful concepts, such as agent-sets (set of turtles, patches or links with certain properties) and a large number of so-called primitives, which are procedures for representing tasks and algorithms which are needed in virtually all ABMs, such identifying the set of all neighbour grid cells, identifying all turtles within a certain radius, or moving in a certain direction and distance.

Interestingly, some NetLogo conventions closely correspond to the structure of ODD (Table 1). Virtually all NetLogo programs have a "setup" button, which corresponds to ODD's fourth element, "Initialisation", and a "go" button, which corresponds to ODD's "Process Overview and Scheduling" element. NetLogo's "procedures" and "reporters" (subroutines and functions) correspond to the ODD element "Submodels".

2.2. The models

We conducted a replication exercise with 17 previously published models and one unpublished one. There were no restrictions on the choice of model for the replication exercise, except that the replicating person must not have been involved in the development of the model. We avoided overly complex models that would have taken more time to replicate and thoroughly test than would have been useful for a demonstration exercise. The 18 models (Table 2) were from ecology, social sciences/game theory, microbiology, epidemiology, and behavioural ecology.

For the 17 published models, no original code was used initially, and only in two cases was the original code asked for and provided (Fielding, 2004; Wissel, 1992). Only four studies mention the programming language used: Java (Davis et al., 1999), C++ (Fielding, 2004); Voyons

Table 2
Overview of the 18 models that were used for the replication exercise in this study.

Reference	System and question	Replication success[1]	Gaps in original model description[2]	New Results	Time / days[3]
Fielding (2004)	Intraspecific competition and spatial heterogeneity alter life history traits in an individual-based model of grasshoppers.	Partly	Parameter, Code, Equations	No	12
Weiner et al. (2001)	The effects of density, spatial pattern, and competitive symmetry on size variation in simulated plant populations.	Yes	No	Yes	4.5
Hauert and Doebeli (2004) *	Limited evolution of cooperation in a spatially explicit snowdrift game	Yes	No	Yes	4
Wissel (1992)	Modelling the mosaic cycle of a Middle European beech forest	Yes	Rules, Scheduling, Code, Author	No	5
Deffuant et al. (2002)*	How can extremism prevail? A study based on the relative agreement interaction model.	Yes	No	No	4
Silvertown et al. (1992)*	Cellular automaton models of interspecific competition for space – the effect of pattern on process.	Yes	No	No	2
Axelrod (1997)*	The dissemination of culture: A model with local convergence and global polarization.	Yes	No	No	8
DeAngelis et al. (1979)	Cannibalism and size dispersal in young-of-the-year largemouth bass: experiment and model.	Yes	Parameter	Yes	5
Hauert et al. (2002)*	Volunteering as Red Queen mechanism for cooperation in public goods games.	Yes	No	No	5.5
Davis et al. (1999)	Environmental quality predicts parental provisioning decisions.	No	Rules, Equations	No	4.5
O`Keefe (2005)	The evolution of virulence in pathogens with frequency-dependent transmission.	Partly	Parameter, Equations	No	4
Kerr et al. (2002)*	Local dispersal promotes biodiversity in a real-life game of rock–paper–scissors.	Yes	Rules	No	2.25
Hansen et al. (2004)	Modelling the transmission cycle of the fox Tapeworm	Yes	No	Yes	8 N
Thiery et al. (1995)	A model simulating the genesis of banded vegetation patterns in Niger.	Yes	No	No	4
Jackson et al. (2008)*	The effect of social facilitation on foraging success in vultures: a modelling study	Yes	No	No	10 N
Ratz (1995)	Long-term spatial patterns created by fire: a model oriented towards boreal forests.	Yes	No	Yes	8.5
Ginovart et al. (2002)	Individual based simulations of bacterial growth in agar plates.	Yes	Parameters, Authors, Code, Equations	Yes	7 N
A.Costopoulos, R. Jobling (unpublished)*	How does the rate and magnitude of cultural innovation affect the ability of hominin groups to expand their species' ecological range?	No	Parameter ranges, Rules, Authors, Repetitions	Yes	10.5

[1] Yes: main results qualitatively reproduced; Partly: yes, but not for all scenarios; No: replication failed.

[2] Parameter: settings incomplete or unknown; code: available but uncommented; rules/equations/scheduling: partly inconsistent or illogical; authors: were contacted and replied.

[3] 1 day = 8 h.

* Stylized model, i.e. not based on specific data or observed patterns

N: Replicating person was beginner in programming and modelling and got help from colleagues.

(Thiery et al., 1995), and C (Weiner et al., 2001).

2.3. Replication

Those replicating the models were recruited from the Department of Ecological Modelling at the UFZ, or were close collaborators. Their motivation was to learn about the usefulness and potential of ODD and NetLogo as well as the replicability of modelling results. They were mostly experienced modellers who routinely used at least one programming language for their work, but only one of them has a background in computer science and software engineering (JCT). Three replicating persons did not have any experience in programming and modelling, but got some help from experienced colleagues.

The replication tasks were: 1) rewrite the original model description according to the ODD protocol (original ODD version from 2006), 2) re-implement the model using NetLogo, 3) try to replicate at least one exemplary main result, 4) try to, if time allowed, achieve some first new results, either by running new scenarios or by modifying the original model assumptions, and 5) report on the time required for the whole exercise and on the challenges during the entire task.

3. Results

All 18 ODD model descriptions and the corresponding reports about replication of original results, new results, and general comments, are available at https://doi.org/10.5281/zenodo.13983991 (Zenodo) The NetLogo programs, in NetLogo versions 4.0.5 and 6.4.0, are available at https://doi.org/10.5281/zenodo.13984010 (Zenodo).

For 15 of the 18 models, exemplary main results were reproduced; for the two other models, only partial replication was achieved because gaps and ambiguities in the original model description could not be closed or resolved, respectively, and for one, albeit unpublished, model replication failed (Table 2). In the original model descriptions of eight of the models, some key information was missing, including parameter values (4 models), parameter ranges (1), model equations or rules (6), or scheduling of processes (1). In three cases help from the original authors was needed and thankfully provided. In Fig. 1, example output of four of the replicated models is presented.

Those who commented on using ODD in the replication exercises were positive, in particular regarding the O-part as it helped to scan and rewrite the original model description in a systematic way. There was one complaint about some redundancy in ODD, but this has been discussed and justified in Grimm et al. (2010). In most exercises, NetLogo code was included in the ODD model description, but this is against one of the main aims of ODD to be independent of programming languages: ODD is meant as a verbal, or written, description. Still, having a few lines of code included can be acceptable if they serve more or less as pseudocode and help specify implementation details (see, for example, the ODD of the model BEEHAVE, Becher et al. (2014)).

NetLogo turned out to be easy and fast to learn, although this requires getting used to some new and unique concepts, such as "asking" sets of agents, which share a certain trait, to do something instead of using loops and if-then conditions. Also the three novice programmers succeeded, with some help of colleagues with experience in

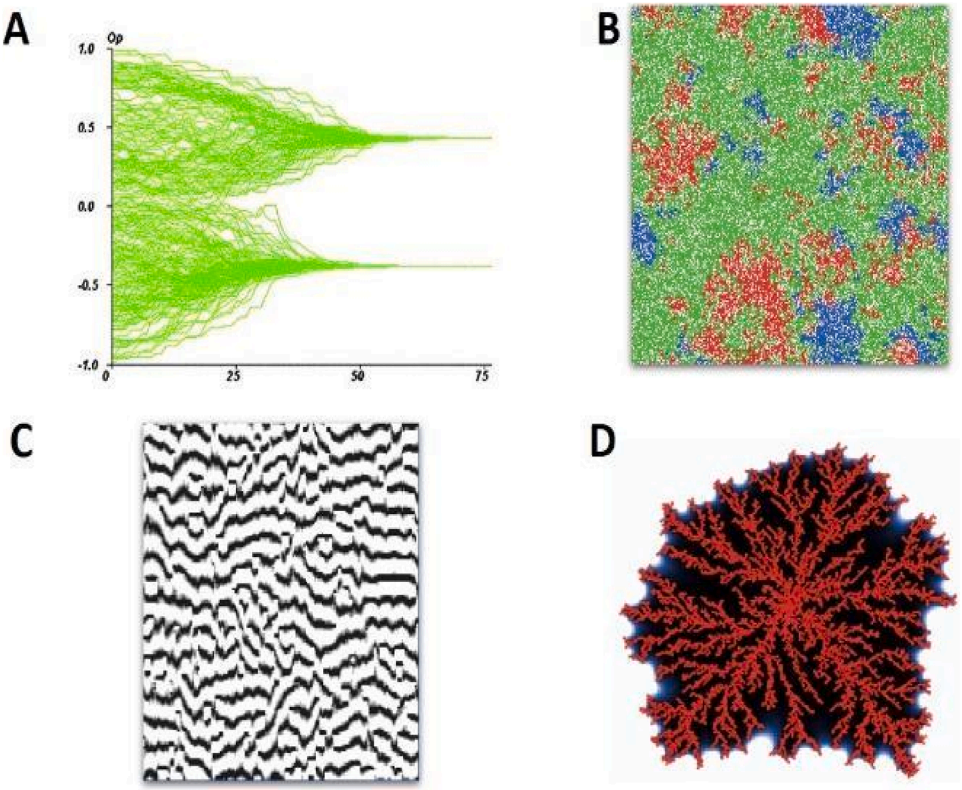


Fig. 1. Output of replicated models that matched the output of four of the original models. A: Opinion dynamics (opinion between -1 and +1 vs time, Deffuant et al., 2002); B: Rock-paper-scissor bacterial community dynamics (spatial distribution of three different strains of bacteria, Kerr et al., 2002); C: Banded vegetation pattern emerging in semi-arid regions (black: wood vegetation, white: other vegetation or bare soil, Thiery et al., 1995); D: Growth of a bacterial colony (Ginovart et al., 2002). See supplements for details.

programming.

Virtually in all replication exercises, the programmer missed having an integrated debugger that allows the program to be executed line-by-line while watching how variables are changing, or to set breakpoints that would halt execution of the program in certain parts of the code or under certain conditions. For debugging, NetLogo allows us to use aspects of the Graphical User Interface for low-level debugging, e.g. by using different colours for different states of the agents or by attaching labels to the agents. Such techniques are too limited for full debugging, so one must use print statements or file output to observe if and how state variables are changing. Another frequent complaint in 2009 was slow execution speed, but, unlike the lack of a debugger, execution speed is no longer an issue. Now, NetLogo is widely used successfully for large models (Railsback et al., 2017) because computers are much faster, because HPC clusters or cloud computing are increasingly available, and because NetLogo's built-in commands (primitives) use highly optimized algorithms.

In six replications, some new results were also produced (Table 2), which was an optional task. The time invested in the replication exercise varied between 2 and 12 days (8 h per day) with a mean of 5 days (Fig. 2) for the relatively simple models we used. In most cases testing and debugging the code was the most time-consuming task.

4. Discussion

The purpose of this study was to show that, using ODD and NetLogo, existing agent-based models can be replicated reliably and with relatively low effort. We have also shown in some of the examples that once one has replicated a model, one can start new research by modifying the model's assumptions and structure, or by running new simulation experiments that help to better understand the original model or apply it to new parameter settings.

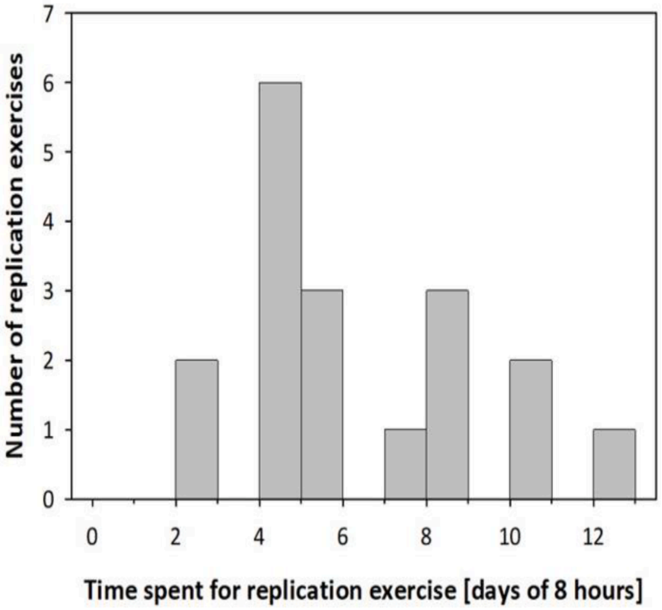


Fig. 2. Histogram of the time spent replicating the 18 models. The mean time spent is five days.

Replication is challenging because model descriptions usually show only the end result of iterative model development. It can be difficult to understand the rationale for all the model's assumptions without knowing the path by which the model was developed, and it can be difficult to describe this rationale in sufficient detail for a model to be fully understood. Also, verbal descriptions of models, even when attempting to be complete, often contain ambiguities that can only be resolved by having access to the code that implements the model.

We found the time taken to replicate to be surprisingly low, averaging just five days. This is even more remarkable considering that those involved in the replication exercises had no or very little prior experience with using NetLogo or ODD. Presumably, replication could be even faster with experience writing and testing NetLogo code.

Obviously, replicating more complex models than those used in this study will take longer. For example, a NetLogo replication of the complex and highly realistic trout model Instream (Railsback and Harvey, 2002) took about six months, but then the model was augmented with population genetics to study the evolution of specific traits under environmental change (Ayllón et al., 2016). In these six months, a deep understanding of the model assumptions and structure, and therefore of fish physiology and behaviour, as well as river flow dynamics and habitat quality, was gained.

Furthermore, the risk of misinterpreting the verbal model description increases for complex models. Donkin et al. (2017) first re-implemented a fairly complex ABM using NetLogo and were unable to replicate the original results at any level, i.e. numerical, distributional or relational (Axtell et al., 1996). They attributed these differences to a misunderstanding of how key mechanisms were interpreted without having the original code. However, when they attempted to replicate the results of their NetLogo implementation using Repast Symphony (North et al., 2013), the results again did not match. They attributed the difference to the different programming styles required for the two platforms. While NetLogo includes many high-level primitives for describing agent behaviour and decisions, Repast, which is a Java library, requires more elements of a model to be implemented from scratch.

We take the study by Donkin et al. (2017) as a caveat to be careful to fully understand how NetLogo's primitives work and, if in doubt, to test them carefully. The same applies, of course, to code you wrote yourself. Therefore, key behaviours should ideally be implemented in two different languages and produce the same results, to be sure that the implementation really does what the model description says it does, and that the behaviour of the submodel of that behaviour is fully understood (see an educational example in Chapter 12 of Railsback and Grimm (2019)).

The advantage of using ODD for replication is that it provides a checklist for verifying all the key features of a model, starting with the purpose, structure and scheduling of the model, and including all details of initialisation, file input and any sub-models. Model replication can fail at any of these levels, and indeed in our study, some examples reported missing information on parameter values and ranges, scheduling, initial settings or, most commonly, details of model equations and algorithms. Thus, the use of ODD could cause reprogramming to be postponed or even cancelled, depending on the size of the information gap and whether it was possible to obtain the missing information from the original authors.

ODD is designed to contain all the information needed to replicate a model, but ambiguities are hard to avoid. Therefore, the latest ODD update (Grimm et al., 2020b) recommends that the ODD description be closely aligned with the code, using the same names for variables, parameters and procedures, and including numbers of equations, model rules or algorithms as comments in the code. With the increasing popularity and acceptance of the FAIR principles (Wilkinson et al., 2016), more and more journals require not only data but also scripts and code to be made available, and it seems natural and very useful to actually provide a model description that is tightly integrated with the code.

In general, we recommend that every modeller should try to replicate a model at least once, because by learning how difficult it can be to replicate a model based on an incomplete or inconsistent model description, one also learns to improve one's own model description and code preparation. In Table 3, we summarise general recommendations for communicating a model and how it has been used, which may serve as a checklist by both model developers and model replicators. If such a checklist were widely followed, it would implicitly establish good modelling practice. Gürcan et al. (2023) provide a more detailed checklist for the replication of agent-based models, using the HUMAT model as an example (Antosz et al., 2022).

The original motivation for the replication exercises presented here was to suggest that ODD and NetLogo can be used as a common language or *lingua franca* in the modelling community. ODD has indeed become a widely used standard (Vincenot, 2018), and NetLogo also has thousands of users, but we are not suggesting that all modellers should use it exclusively. Rather, all modellers might want to know how to use NetLogo. Even when another platform is chosen for a particular model, NetLogo is a very efficient tool for prototyping and replicating (e.g., for code verification) parts or all of an agent-model.

The vision described in the Introduction, that all models would be described in a format such as ODD and implemented in a generic platform and programming language that is easy to learn, understand and execute, is in fact no longer a vision but to a large extent already a reality. ODD and NetLogo as standard tools have the potential to make modelling more coherent and efficient, to promote theory development by starting more projects from existing models rather than from scratch, and to make modelling more transparent and reliable and thus better suited to supporting decision making (Grimm et al., 2020a).

Both ODD and NetLogo are proof of concept that standards are indeed possible and necessary to improve modelling practice and lead to more coherence. Different models addressing the same question often produce incompatible predictions because the modellers have different preferences and backgrounds and use different expert opinions,

Table 3
A short checklist of questions modeller and those interested in model replication should ask themselves to make sure, for modellers, that their model can be both re-run and replicated, and for replicators that they have sufficient information to make replication work.

Replication question	Techniques to ensure replicability
Is the model description complete?, Would I be able to replicate my model solely based on the model description?	Use ODD. Write it by describing what your program does, not what you think the model is. Be specific, explicit, and complete about parameter settings and ranges, scheduling of actions, initial settings, data imported during runtime, and all details of the equations and algorithms used. Let others who do not know your model read the ODD. Try to separately replicate complex submodels yourself in a different language (R, Python, C++, Julia etc.)
Is the model description well-linked to the program implementing the model?	Use the same names for variables, parameters, and procedures in the ODD and in the program. Numbers of equations or algorithms in the ODD should be added as comments in the program.
Did you provide all files to re-run your model?	These include the source code and possibly files for data or parameter input, plus documentation of the software platform and operating system used, and which version.
Are the parameter settings underlying all scenarios and simulation experiments fully specified?	In NetLogo, include any BehaviorSpace experiment setups used, otherwise provide all parameter settings and scenario descriptions used for all results presented.
If your workflow included any scripts for, e.g. processing input data or output analysis, are those scripts included?	Include them. Often a lot of work goes into processing raw data or model output, so this is part of your achievements. Others might fail in using your model if the workflow is not fully documented or the scripts unavailable.

calibration data and decision contexts (Alexander et al., 2017; Bahlburg et al., 2023; Banitz et al., 2022; Grimm, 2023). Standards like ODD do not resolve these differences per se, but they help making them explicit so that models and their underlying assumptions can be systematically compared and the model outcomes integrated (Wimmler et al., 2024). Standards are key to improving and establishing Good Modelling Practice, making models more fit for purpose (Hamilton et al., 2022) and aligning them with FAIR principles (Barton et al., 2022). Therefore, the Open Modelling Foundation (OMF[2]), with >40 modelling organisations as members, is currently working on the development and establishment of standards covering all aspects of modelling and model applications.

CRediT authorship contribution statement

Volker Grimm: Writing – review & editing, Writing – original draft, Conceptualization. **Uta Berger:** Writing – original draft, Software. **Justin M. Calabrese:** Writing – original draft, Software. **Ainara Cortés-Avizanda:** Writing – original draft, Software. **Jordi Ferrer:** Writing – original draft, Software. **Mathias Franz:** Writing – original draft, Software. **Jürgen Groeneveld:** Writing – original draft, Software. **Florian Hartig:** Writing – original draft, Software. **Oliver Jakoby:** Writing – original draft, Software. **Roger Jovani:** Writing – original draft, Software. **Stephanie Kramer-Schadt:** Writing – original draft, Software. **Tamara Münkemüller:** Writing – original draft, Software. **Cyril Piou:** Writing – original draft, Software. **L.S. Premo:** Writing – original draft, Software. **Sandro Pütz:** Writing – original draft. **Thomas Quintaine:** Writing – original draft, Software. **Christine Rademacher:** Writing – original draft, Software. **Nadja Rüger:** Writing – original draft, Software. **Amelie Schmolke:** Writing – original draft, Software. **Jan C. Thiele:** Writing – original draft, Software. **Julia Touza:** Writing – original draft, Software. **Steven F. Railsback:** Writing – review & editing, Writing – original draft.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was partially funded by the Center for Advanced Systems Understanding (CASUS), which is financed by Germany’s Federal Ministry of Education and Research (BMBF) and by the Saxon Ministry for Science, Culture and Tourism (SMWK) with tax funds on the basis of the budget approved by the Saxon State Parliament. We would like to thank two anonymous reviewers for their insightful comments.

Data availability

No data was used for the research described in the article.

References

Abar, S., Theodoropoulos, G.K., Lemarinier, P., O’Hare, G.M., 2017. Agent-based modelling and simulation tools: a review of the state-of-art software. Comput. Sci. Rev. 24, 13–33.

Alexander, P., Prestele, R., Verburg, P.H., Arneth, A., Baranzelli, C., Batista e Silva, F., Brown, C., Butler, A., Calvin, K., Dendoncker, N., 2017. Assessing uncertainties in land cover projections. Glob. Chang. Biol. 23, 767–781.

Antosz, P., Birks, D., Edmonds, B., Heppenstall, A., Meyer, R., Polhill, J.G., O’Sullivan, D., Wijermans, N., 2023. What do you want theory for?-A pragmatic analysis of the roles of “theory” in agent-based modelling. Environ. Modell. Softw. 168, 105802.

Antosz, P., Shults, F.L., Puga-Gonzalez, I., Szczepanska, T., 2022. HUM-e: An Emotive-Socio-cognitive Agent Architecture for Representing Human Decision-Making in Anxiogenic Contexts, Conference of the European Social Simulation Association. Springer, pp. 235–247.

Axelrod, R., 1997. The dissemination of culture - A model with local convergence and global polarization. J. Confl. Resol. 41, 203–226.

Axtell, R., Axelrod, R., Epstein, J.M., Cohen, M.D., 1996. Aligning simulation models: a case study and results. Comput. Mathemat. Organiz. Theory 1, 123–141.

Ayllón, D., Railsback, S.F., Gallagher, C., Augusiak, J., Baveco, H., Berger, U., Charles, S., Martin, R., Focks, A., Galic, N., Liu, C., van Loon, E.E., Nabe-Nielsen, J., Piou, C., Polhill, J.G., Preuss, T.G., Radchuk, V., Schmolke, A., Stadnicka-Michalak, J., Thorbek, P., Grimm, V., 2021. Keeping modelling notebooks with TRACE: good for you and good for environmental research and management support. Environ. Modell. Softw. 136, 104932.

Ayllón, D., Railsback, S.F., Vincenzi, S., Groeneveld, J., Almodóvar, A., Grimm, V., 2016. InSTREAM-Gen: modelling eco-evolutionary dynamics of trout populations under anthropogenic environmental change. Ecol. Modell. 326, 36–53.

Bahlburg, D., Thorpe, S.E., Meyer, B., Berger, U., Murphy, E.J., 2023. An intercomparison of models predicting growth of Antarctic krill (Euphausia superba): the importance of recognizing model specificity. PLoS. One 18, e0286036.

Banitz, T., Schlüter, M., Lindkvist, E., Radosavljevic, S., Johansson, L.-G., Ylikoski, P., Martínez-Peña, R., Grimm, V., 2022. Model-derived causal explanations are inherently constrained by hidden assumptions and context: the example of Baltic cod dynamics. Environ. Modell. Softw. 156, 105489.

Barton, C.M., Ames, D., Chen, M., Frank, K., Jagers, H.R.A., Lee, A., Reis, S., Swantek, L., 2022. Making modeling and software FAIR. Env. Model. Softw. 156, 105496.

Becher, M.A., Grimm, V., Thorbek, P., Horn, J., Kennedy, P.J., Osborne, J.L., 2014. BEEHAVE: a systems model of honeybee colony dynamics and foraging to explore multifactorial causes of colony failure. J. Appl. Ecol. 51, 470–482.

Carney, M., Davies, B., 2020. Agent-based modeling, scientific reproducibility, and taphonomy: a successful model implementation case study. J. Comput. Applicat. Archaeol. 3, 182–196.

Davis, J.N., Todd, P.M., Bullock, S., 1999. Environment quality predicts parental provisioning decisions. Proceed. Royal Soc. London. Series B 266, 1791–1797.

DeAngelis, D.L., Cox, D.K., Coutant, C.C., 1979. Cannibalism and size dispersal in young-of-the-year largemouth bass: experiment and model. Ecol. Modell 8, 133–148.

Deffuant, G., Amblard, F., Weisbuch, G., Faure, T., 2002. How can extremism prevail? A study based on the relative agreement interaction model. JASSS-J. Artif. Societ. Soc. Simul. 5. http://jasss.soc.surrey.ac.uk/5/4/1.html.

Donkin, E., Dennis, P., Ustalakov, A., Warren, J., Clare, A., 2017. Replicating complex agent based models, a formidable task. Environ. Modell. Softw. 92, 142–151.

Edmonds, B., Hales, D., 2003. Replication, replication and replication: some hard lessons from model alignment. J. Artif. Societ. Soc. Simul. 6 <. https://www.jasss.org/6/4/11.html>.

Fielding, D.J., 2004. Intraspecific competition and spatial heterogeneity alter life history traits in an individual-based model of grasshoppers*. Ecol. Modell. 175, 169–187.

Filazzola, A., Cahill Jr, J.F., 2021. Replication in field ecology: identifying challenges and proposing solutions. Methods Ecol. Evol. 12, 1780–1792.

Fraser, H., Barnett, A., Parker, T.H., Fidler, F., 2020. The role of replication studies in ecology. Ecol. Evol. 10, 5197–5207.

Gallagher, C.A., Grimm, V., Kyhn, L.A., Kinze, C.C., Nabe-Nielsen, J., 2021. Movement and seasonal energetics mediate vulnerability to disturbance in marine mammal populations. Am. Nat. 197, 296–311.

Ginovart, M., López, D., Valls, J., Silbert, M., 2002. Individual based simulations of bacterial growth on agar plates. Physica A 305, 604–618.

Grimm, V., 2023. Ecology needs to overcome siloed modelling. Trends Ecol. Evol. 38, 1122–1124.

Grimm, V., Augusiak, J., Focks, A., Frank, B.M., Gabsi, F., Johnston, A.S.A., Liu, C., Martin, B.T., Meli, M., Radchuk, V., Thorbek, P., Railsback, S.F., 2014. Towards better modelling and decision support: documenting model development, testing, and analysis using TRACE. Ecol. Modell. 280, 129–139.

Grimm, V., Berger, U., 2016. Robustness analysis: deconstructing computational models for ecological theory and applications. Ecol. Modell. 326, 162–167.

Grimm, V., Berger, U., Bastiansen, F., Eliassen, S., Ginot, V., Giske, J., Goss-Custard, J., Grand, T., Heinz, S., Huse, G., Huth, A., Jepsen, J.U., Jørgensen, C., Mooij, W.M., Müller, B., Pe’er, G., Piou, C., Railsback, S.F., Robbins, A.M., Robbins, M.M., Rossmanith, E., Rüger, N., Strand, E., Souissi, S., Stillman, R.A., Vabø, R., Visser, U., DeAngelis, D.L., 2006. A standard protocol for describing individual-based and agent-based models. Ecol. Modell 198, 115–126.

Grimm, V., Berger, U., DeAngelis, D.L., Polhill, J.G., Giske, J., Railsback, S.F., 2010. The ODD protocol: a review and first update. Ecol. Modell 221, 2760–2768.

Grimm, V., Berger, U., Meyer, M., Lorscheid, I., 2024. Theory for and from agent-based modelling: insights from a virtual special issue and a vision. Environ. Modell. Softw. 178, 106088.

Grimm, V., Johnston, A.S., Thulke, H.-H., Forbes, V., Thorbek, P., 2020a. Three questions to ask before using model outputs for decision support. Nat. Commun. 11, 1–3.

Grimm, V., Railsback, S.F., 2005. Individual-based modeling and ecology. Princeton University Press. Princeton and, Oxford, p. 420.

Grimm, V., Railsback, S.F., Vincenot, C.E., Berger, U., Gallagher, C., DeAngelis, D.L., Edmonds, B., Ge, J., Giske, J., Groeneveld, J., Johnston, A.S.A., Milles, A., Nabe-Nielsen, J., Polhill, J.G., Radchuk, V., Rohwäder, M.-S., Stillman, R.A., Thiele, J.C., Ayllón, D., 2020b. The ODD protocol for describing agent-based and other simulation models: a second update to improve clarity, replication, and structural realism. J. Artif. Societ. Soc. Simul. 23. https://www.jasss.org/23/2/7.html.

---

[2] https://www.openmodelingfoundation.org/

Groeneveld, J., Martinovič, T., Rossi, T., Salamon, O., Sara-aho, K., Grimm, V., 2024. Prototype biodiversity digital twin: honey bees in agricultural landscapes. Res. Ideas. Outcomes. 10, e125167.

Gürcan, Ö., Szczepanska, T., Antosz, P., 2023. A Guide to *Re*-implementing Agent-Based Models: Experiences from the HUMAT Model, Conference of the European Social Simulation Association. Springer, pp. 519–531.

Hales, D., Rouchier, J., Edmonds, B., 2003. Model-to-model analysis. J. Artif. Societ. Soc. Simul. 6. <https://www.jasss.org/6/4/5.html>.

Hamilton, S.H., Pollino, C.A., Stratford, D.S., Fu, B., Jakeman, A.J., 2022. Fit-for-purpose environmental modeling: Targeting the intersection of usability, reliability and feasibility. Env. Modell. Softw. 148, 105278.

Hansen, F., Jeltsch, F., Tackmann, K., Staubach, C., Thulke, H.-H., 2004. Processes leading to a spatial aggregation of Echinococcus multilocularis in its natural intermediate host Microtus arvalis. Int. J. Parasitol. 34, 37–44.

Hauert, C., De Monte, S., Hofbauer, J., Sigmund, K., 2002. Volunteering as red queen mechanism for cooperation in public goods games. Science 296, 1129–1132.

Hauert, C., Doebeli, M., 2004. Spatial structure often inhibits the evolution of cooperation in the snowdrift game. Nature 428, 643–646.

Hauke, J., Achter, S., Meyer, M., 2020. Theory development via replicated simulations and the added value of standards. J. Artif. Societ. Soc. Simul. 23.

Jackson, A.L., Ruxton, G.D., Houston, D.C., 2008. The effect of social facilitation on foraging success in vultures: a modelling study. Biol. Lett. 4, 311–313.

Jakeman, A.J., Elsawah, S., Wang, H.-H., Hamilton, S.H., Melsen, L., Grimm, V., 2024. Towards normalizing good practice across the whole modeling cycle: its instrumentation and future research topics. Socio-Environ. Syst. Modell. 6, 18755. -18755.

Kerr, B., Riley, M.A., Feldman, M.W., Bohannan, B.J.M., 2002. Local dispersal promotes biodiversity in a real-life game of rock-paper-scissors. Nature 418, 171–174.

Müller, B., Balbi, S., Buchmann, C.M., de Sousa, L., Dressler, G., Groeneveld, J., Klassert, C.J., Le, Q.B., Millington, J.D.A., Nolzen, H., Parker, D.C., Polhill, J.G., Schlüter, M., Schulze, J., Schwarz, N., Sun, Z., Taillandier, P., Weise, H., 2014. Standardised and transparent model descriptions for agent-based models: current status and prospects. Environ. Modell. Softw. 55, 156–163.

North, M.J., Collier, N.T., Ozik, J., Tatara, E.R., Macal, C.M., Bragen, M., Sydelko, P., 2013. Complex adaptive systems modeling with Repast Simphony. Complex Adapt. Syst. Model. 1, 1–26.

O'Keefe, K.J., 2005. The evolution of virulence in pathogens with frequency-dependent transmission. J. Theor. Biol. 233, 55–64.

Papert, S., 1985. Different visions of logo. Comput. Schools 2, 3–8.

Parker, T.H., Forstmeier, W., Koricheva, J., Fidler, F., Hadfield, J.D., Chee, Y.E., Kelly, C. D., Gurevitch, J., Nakagawa, S., 2016. Transparency in ecology and evolution: real problems, real solutions. Trends Ecol. Evol. (Amst.) 31, 711–719.

Polhill, J.G., Brown, D.G., Grimm, V., 2008. Using the ODD protocol for describing three agent-based social simulation models of land use change. JASSS-J. Artif. Societ. Soc. Simul. 11. https://jasss.soc.surrey.ac.uk/11/12/13.html.

Railsback, S.F., Ayll¢n, D., Berger, U., Grimm, V., Lytinen, S.L., Sheppard, C.J.R., Thiele, J.C., 2017. Improving execution speed of models implemented in NetLogo. J. Artif. Societ. Soc. Simul. 20. https://doi.org/10.18564/jasss.3282.

Railsback, S.F., Grimm, V., 2019. Agent-based and Individual-based Modeling: a Practical Introduction, 2nd edition. Princeton University Press. Princeton and, Oxford, p. 339.

Railsback, S.F., Harvey, B.C., 2002. Analysis of habitat-selection rules using an individual-based model. Ecology 83, 1817–1830.

Ratz, A., 1995. Long-term spatial patterns created by fire: a model oriented towards boreal forests. Int. J. Wildland. Fire 5, 25–34.

Schmolke, A., Thorbek, P., DeAngelis, D.L., Grimm, V., 2010. Ecological models supporting environmental decision making: a strategy for the future. Trends Ecol. Evol 25, 479–486.

Silvertown, J., Holtier, S., Johnson, J., Dale, P., 1992. Cellular automaton models of interspecific competition for space–the effect of pattern on process. J. Ecol. 527–533.

Thiele, J.C., Grimm, V., 2015. Replicating and breaking models: good for you and good for ecology. Oikos. 124, 691–696.

Thiery, J.M., D'Herbes, J.M., Valentin, C., 1995. A model simulating the genesis of banded vegetation patterns in Niger. J. Ecol 83, 497–507.

Vincenot, C.E., 2018. How new concepts become universal scientific approaches: insights from citation network analysis of agent-based complex systems science. Proceed. Royal Soc. B-Biolog. Sci. 285.

Warszawski, L., Frieler, K., Huber, V., Piontek, F., Serdeczny, O., Schewe, J., 2014. The inter-sectoral impact model intercomparison project (ISI–MIP): project framework. Proceed. Nation. Acad. Sci. 111, 3228–3232.

Weiner, J., Stoll, P., Muller-Landau, H., Jasentuliyana, A., 2001. The effects of density, spatial pattern, and competitive symmetry on size variation in simulated plant populations. Am. Naturalist 158, 438–450.

Wijermans, N., Scholz, G., Chappin, É., Heppenstall, A., Filatova, T., Polhill, J.G., Semeniuk, C., Stöppler, F., 2023. Agent decision-making: the Elephant in the Room-Enabling the justification of decision model fit in social-ecological models. Environ. Modell. Softw. 170, 105850.

Wilensky, U., 1999. NetLogo. Northwestern University, Center For Connected Learning and Computer-Based Modeling, Evanston.

Wilensky, U., Rand, W., 2015. An Introduction to Agent-Based modeling: Modeling natural, social, and Engineered Complex Systems With NetLogo. MIT press.

Wilkinson, M.D., Dumontier, M., Aalbersberg, I.J., Appleton, G., Axton, M., Baak, A., Blomberg, N., Boiten, J.-W., da Silva Santos, L.B., Bourne, P.E., 2016. The FAIR Guiding Principles for scientific data management and stewardship. Sci. Data 3, 1–9.

Wimmler, M.-C., Bathmann, J., Vollhüter, J., Berger, U., 2024. pyMANGA: a modular, open and extendable software platform for modeling of forest and vegetation dynamics. Environ. Modell. Softw., 105973

Wissel, C., 1992. Modelling the mosaic cycle of a middle European beech forest. Ecol. Modell. 63, 29–43.

Zhang, J., Robinson, D.T., 2021. Replication of an agent-based model using the replication standard. Environ. Modell. Softw. 139, 105016.

Zhong, W., Kim, Y., 2010. Using model replication to improve the reliability of agent-based models, Advances in Social Computing: Third International Conference on Social Computing, Behavioral Modeling, and Prediction, SBP 2010, Bethesda, MD, USA, March 30-31, 2010. Proceedings 3. Springer, pp. 118–127.