Chapter 8
Documenting Social Simulation Models: The ODD Protocol as a Standard

Volker Grimm, Gary Polhill, and Julia Touza

AU1

Abstract The clear documentation of simulations is important for their communication, replication, and comprehension. It is thus helpful for such documentation to follow minimum standards. The "Overview, Design concepts and Details" document protocol (ODD) is specifically designed to guide the description of individual- and agent-based simulation models (ABMs) in journal articles. Popular among ecologists, it is also increasingly used in the social simulation community. Here, we describe the protocol and give an annotated example of its use, with a view to facilitating its wider adoption and encouraging higher standards in simulation description.

Why Read This Chapter? To learn about the importance of documenting your simulation model and discover a lightweight and appropriate framework to guide you in doing this.

8.1 Introduction and History

A description protocol is a framework for guiding the description of something, in this case a social simulation model. It can be thought of as a check-list of things that need to be covered and rules that should be followed when specifying the details of a simulation (in a scholarly communication). Following such a protocol means that readers can become familiar with its form and that key elements are less likely to be forgotten. This chapter describes a particular documentation protocol, the ODD (pronounced: "odd", or "oh dee dee") protocol.

V. Grimm (✉)
UFZ, Helmholtz Centre of Environmental Research, UFZ, Department of Ecological Modelling, Permoserstr. 15, 04318 Leipzig, Germany

G. Polhill
The James Hutton Institute, Craigiebuckler, Aberdeen AB15 8QH, United Kingdom

J. Touza
Applied Economics Department, University of Vigo, Vigo C.P. 36.310, Spain

B. Edmonds and R. Meyer (eds.), *Simulating Social Complexity*,
Understanding Complex Systems, DOI 10.1007/978-3-540-93813-2_8,

The ODD protocol (Grimm et al. 2006, 2010; Polhill et al. 2008; Polhill 2010) is a standard layout for describing individual- and agent-based simulation models (ABMs), especially for journal articles, conference papers, and other academic literature. It consists of seven elements which can be grouped into three blocks: Overview, Design concepts, Details (hence, "ODD"; see Table 8.1). The purpose of ODD is to facilitate writing and reading of model descriptions, to better enable replication of model-based research, and to establish a set of design concepts that should be taken into account while developing an ABM. It does this in a relatively lightweight way, avoiding over-formal approaches whilst ensuring that the essentials of a simulation are explicitly described in a flexible yet appropriate manner.

Originally, ODD was formulated by ecologists, where the proportion of ABMs described using ODD is increasing fast and might cross the 50 % margin in the near future. In social simulation, the acceptance of ODD has been slower. A first test, in which three existing descriptions of land-use models were re-formulated according to ODD, demonstrated the benefits of using ODD but also revealed that some refinements were needed to make it more suitable for social simulation (Polhill et al. 2008). In 2010, an update of ODD was released (Grimm et al. 2010), which is based on users' feedback and a review of more than 50 ODD-based model descriptions in the literature. In this update, ODD itself was only slightly modified but the explanation of its elements completely rewritten, with the specific intention of making it more suitable for social simulation.

Currently in social simulation, interest in ODD is also increasing (Polhill 2010). An indicator for this is the inclusion of ODD chapters in recent reference books (this volume; Heppenstall et al. 2012). Moreover, a recent textbook of agent-based modelling uses ODD consistently (Railsback and Grimm 2012), so that the next generation of agent-based modellers is more likely to be familiar with ODD, and hence to use it themselves.

8.2 The Purpose of ODD

Why is ODD (or a protocol very much like it) needed? There are a number of endeavours in agent-based social simulation that are facilitated through having a common approach to describing the models that is aimed at being readable and complete[1]:

- *Communication* is the most basic aim of anyone trying to publish their results. For agent-based modellers, this can pose a particular challenge, as our models can be complicated, with many components and submodels. As a critical mass of

[1] Many of these endeavours have been covered in submissions to the "model-to-model" series of workshops, organised by members of the social simulation community (Hales et al. 2003; Rouchier et al. 2008. The second workshop was held as a parallel session of the ESSA 2004 conference: see http://www.insisoc.org/ESSA04/M2M2.htm).

t1.1 **Table 8.1** The seven elements of the ODD protocol. Descriptions of ABMs are compiled by answering the questions linked to each element

	***O*verview**	1. Purpose		What is the purpose of the model?
t1.2		2. Entities, state variables, scales		What kind of entities are in the model? Do they represent managers, voters, landowners, firms or something else? By what state variables, or attributes, are these entities characterized? What are the temporal and spatial resolutions and extents of the model?
t1.3		3. Process overview, scheduling		What entity does what, in what order? Is the order imposed or dynamic? When are state variables updated? How is time modelled: as discrete steps or as a continuum over which both continuous processes and discrete events can occur?
t1.4	***D*esign concepts**	4. Design concepts	Basic principles	Which general concepts, theories or hypotheses ~~approaches~~ are included in the model's design? How were they taken into account? Are they used at the level of submodels or at the system level?
t1.5			Emergence	What key results are emerging from the adaptive traits, or behaviours of individuals? What results vary in complex/unpredictable ways when particular characteristics change? Are there other results that are more tightly imposed by model rules and hence less dependent on what individuals do?
t1.6			Adaptation	What adaptive traits do the individuals have? What rules do they have for making decisions or changing behaviour in response to changes in themselves or their environment? Do agents seek to increase some measure of success or do they reproduce observed behaviours that they perceive as successful?
t1.7			Objectives	If agents (or groups) are explicitly programmed to meet some objective, what exactly is that and how is it measured? When individuals make decisions by ranking alternatives, what criteria do they use? Note that the objective of such agents as group members may not refer to themselves but the group
t1.8			Learning	May individuals change their adaptive traits over time as a consequence of their experience? If so, how?
t1.9			Prediction	Prediction can be part of decision-making; if an agent's learning procedures are based on estimating future consequences of decisions, how they do this? What internal models do agents use to estimate future conditions or consequences? What 'tacit' predictions are implied in these internal model's assumptions?
t1.10			Sensing	What aspects are individuals assumed to sense and consider? What aspects of which other entities can an individual perceive (e.g. displayed 'signals')? Is sensing local, through networks or global? Is the structure of networks imposed or emergent? Are the mechanisms by which agents obtain information modelled explicitly in a process or is it simply 'known'?

(continued)

t1.12 **Table 8.1** (continued)

t1.11			Interaction	What kinds of interactions among agents are assumed? Are there direct interactions where individuals encounter and affect others, or are interactions indirect, e.g. via competition for a mediating resource? If the interactions involve communication, how are such communications represented?
t1.12			Stochasticity	What processes are modelled by assuming they are random or partly random? Is stochasticity used, for example, to reproduce variability in processes for which it is unimportant to model the actual causes of the variability, or to cause model events or behaviours to occur with a specified frequency?
t1.13			Collectives	Do the individuals form or belong to aggregations that affect, and are affected by, the individuals? Such collectives can be an important intermediate level of organization. How are collectives represented – as emergent properties of the individuals or as a separate kind of entity with its own state variables and traits?
t1.14			Observation	What data are collected from the ABM for testing, understanding, and analyzing it, and how are they collected? Are all output data freely used, or are only certain data sampled and used, to imitate what can be observed in an empirical study?
t1.15	***Details***	5. Initialization		What is the initial state of the model world, i.e., at time $t = 0$? How many entities of what type are there initially, and what are the values of their state variables (or how were they set)? Is initialization always the same, or is it varied? Are the initial values chosen arbitrarily or based on available data?
t1.16		6. Input data		Does the model use input from external sources such as data files or other models to represent processes that change over time?
t1.17		7. Submodels		What are the submodels that represent the processes listed in 'Process overview and scheduling'? What are the model parameters, their dimensions, and reference values? How were submodels designed or chosen, tested, and parameterized?

papers using ODD develops, so readers of agent-based modelling papers will find themselves sufficiently more familiar with papers structured using ODD than those using an arbitrary layout devised by the authors that they will find the former easier to read and understand than the latter.

- *Replication*, as we discuss later in this chapter, is a pillar of the scientific endeavour. If our model descriptions are inadequate, our results are not repeatable, and the scientific value of our work commensurately reduced. ODD helps to encourage the adequacy of descriptions by saving authors having to 'reinvent the wheel' each time they describe a model, by providing a standard layout designed to ensure that all aspects of a model needed to replicate it are included in the account.
- *Comparing models* is likely to become increasingly important as work in agent-based modelling continues. If two or more research teams produce similar models with different outcomes, comparing the models will be essential to identifying the cause of the variance in behaviour. Such comparisons will be much easier if all teams have used the same protocol to describe the models. At a conceptual level, the design concepts also enable comparison of models with greater differences and application domains.
- *Dialogue among disciplines* can be encouraged through a standard that is used by both the ecological and social simulation communities. This is especially useful for those developing coupled socio-ecosystem models (Polhill et al. 2008), which is a rapidly growing area of research (Polhill et al. 2011).

In the following, we briefly describe the rationale of ODD and how it is used, provide an example model description, and finally discuss benefits of ODD, current challenges, and its potential future development.

8.3 The ODD Protocol

A core principle of ODD is that first an 'Overview' of a model's purpose, structure and processes should be provided, *before* 'Details' are presented. This allows readers to quickly get a comprehensive overview of what the model is, what it does, and for what purpose it was developed. This follows the journalistic 'inverted pyramid' style of writing, where a summary is provided in the first one or two paragraphs, and progressively further detail is added on the story the further on you read (see, e.g. Wheeler 2005). It allows the reader to easily access the information they are interested in at the level of detail they need. For experienced modellers, this overview part is sufficient to understand what the model is for, to relate it to other models in the field, and to assess the overall design and complexity.

Before presenting the 'Details', ODD requires a discussion of whether, and how, ten design concepts were taken into account while designing the model. This 'Design concepts' part of ODD does not describe the model itself but the principles and rationale underlying its design. 'Design concepts' is thus not needed for model replication but for making sure that important design decisions were made consciously and that readers are fully aware of these decisions. For example, it is important to be clear about what model output is designed to emerge from the behaviour the model's entities and their interactions, and what, in contrast, is imposed by fixed rules and parameters. Ideally, key behaviours in a model emerge, whereas other elements might be imposed. If modellers are not fully aware of this difference, which is surprisingly often the case, they might impose too much so that model output is more or less hard-wired into its design, or they might get lost in a too complex model because too much emergence makes it hard to understand anything. Likewise, the design concept 'stochasticity' requires that modellers explicitly say what model processes include a stochastic component, why stochasticity was used, and how it was implemented. Note that, in contrast to the seven elements of ODD, the sequence in which design concepts are described can be changed, if needed, and design concepts that are not relevant for the model can be omitted.

The 'Details' part of ODD includes all details that are needed to re-implement the model. This includes information about the values of all model entities' state variables and attributes at the begin of a simulation ('Initialisation'), the external models or data files that are possibly used as 'Input data' describing the dynamics of one or more driving contextual or environmental variables (e.g., rainfall, market price, disturbance events), and 'Details' where the submodels representing the processes listed in 'Process overview and scheduling' are presented. Here, it is recommended for every submodel to start with the factual description of what the submodel is and then explain its rationale.

Model parameters should be presented in a table, referred to in the 'Submodels' section of ODD, including parameter name, symbol, reference value, and – if the model refers to real systems – unit, range, and references or sources for choosing parameter values. Note that the simulation experiments that were carried out to analyse the model, characterized by parameter settings, number of repeated runs, the set of observation variables used, and the statistical analyses of model output, is not part of ODD but ideally should be presented in a section 'Simulation experiments' directly following the ODD-based model description.

8.4 How to Use ODD

To describe an ABM using ODD, the questions listed in Table 8.1 have to be answered. The identifiers of the three blocks of ODD elements – Overview, Design concepts, Details – are not used themselves in ODD descriptions (except for 'Design concepts', which is the only element of the corresponding block). Rather, the seven elements are used as headlines in ODD-based model descriptions. For experienced ODD users, the questions in Table 8.1 are sufficient. For beginners, however, it is recommended to read the more detailed description of ODD in

Grimm et al. (2010) and to use the template, which provides additional questions and examples, and which is available via download.[2]

8.5 An Example

In the supplementary material of Grimm et al. (2010), publications are listed which use ODD in a clear, comprehensive, and recommendable way. Many further examples are provided in the textbook by Railsback and Grimm (2012). In Grimm and Railsback (2012), Schelling's segregation model, as implemented in the model library of the software platform NetLogo (Wilensky 1999), is used as an example. Here, we demonstrate the process of model documentation using ODD by describing a model developed by Deffuant et al. (2002), which explores the emergence of extreme opinions in a population. We choose this model because it is simple but interesting and opinion dynamics models are quite well-known in the social simulation community. It is also one of the introductory examples in Gilbert (2007). The ODD for the Deffuant et al. model is interspersed with comments on the information included, with a view to providing some guidelines for those applying ODD to their own model. Clearly this is a very simple example and many models would require more extensive description. The parts of ODD are set in italics and indented to distinguish them from comments. Normally the ODD description would simply form part of the text in the main body of a paper or in an appendix.[3]

8.5.1 Purpose

> *The model's purpose is to study the evolution of the distribution of opinions in a population of interacting individuals, which is under the influence of extremists' views. Specifically, it aims to answer how marginal extreme opinions can manage to become the norm in large parts of a population. The central idea of the model is that people who have more extreme opinions are more confident than people with moderate views. More confident people are, however, assumed to more easily affect the opinion of others, who are less confident.*

Comments: The purpose section is deliberately brief. Even for more sophisticated models than this, we would not expect to see much more text here. This would otherwise repeat information in the rest of the paper. However, since the ODD, to some extent, needs to stand alone and be comprehensive, the summary of the purpose is included as here.

[2] E.g. http://www.ufz.de/index.php?de=10466.

[3] It is often the case that a substantial description needs to be included in the main text so readers can get an idea of what is being discussed, but maybe a more complete description might be added in an appendix.

8.5.2 Entities, State Variables, and Scales

The model includes only one type of entity: individuals. They are characterised by two continuous state variables, opinion x and uncertainty u. Opinions ranges from −1 to 1. Individuals with an opinion very close to x = −1 or +1 are referred to as "extremists", all other individuals are "moderates". Uncertainty u defines an interval around an individuals' opinion and determines whether two individuals interact and, if they do, on the relative agreement of those two individuals which then determines how much opinion and uncertainty change in the interaction. One time step of the model represents the time in which all individuals have randomly chosen another individual and possibly interacted with it. Simulations run until the distribution of opinions becomes stationary.

Comments: For larger models, this section has the potential to get quite long if written in the same style as this example, which has only one type of entity, with two state variables. Other articles have taken the approach of using tables to express this information; one table per entity, with one row per state variable associated with that entity (see, e.g. Polhill et al. 2008). Other articles have used UML class diagrams (e.g., Bithel et al. 2009), as suggested in the original ODD article (Grimm et al. 2006); however, these do not provide a means for giving any description, however brief, of each state variable. Simply listing the entities and the data types of the state variables does not provide all the information that this element of ODD should provide. This, together with the fact that UML is focused on Object-Oriented Design (which is used to implement the majority of ABMs, but by no means all: NetLogo, for example, is not an object-oriented language, and many, particularly in agent-based social simulation, use declarative programming languages), meant that the recommendation to use UML was retracted in the recent ODD update (Grimm et al. 2010).

In declarative programming languages, the entities and their state variables may not be so explicitly represented in the program code as they are in object-oriented languages. For example, this information may be implicit in the arguments to rules. However, many declarative programs have a database of knowledge that the rules operate on. This database could be used to suggest entities and state variables. For example, a Prolog program might have a database containing the assertions person (volker) and nationality(volker, german). This suggests that 'person' is an entity, and 'nationality' a state variable. (It might be reasonable to suggest in general that assertions with one argument suggest entities, and those with two, state variables.)

8.5.3 Process Overview and Scheduling

In each time step each individual chooses randomly one other individual to interact with, then the relative agreement between these two agents is evaluated, and the focal individual's opinion and uncertainty are immediately updated as a result of this opinion interaction. *Updating of state variables is thus asynchronous. After all individuals have interacted, a* convergence index *is* calculated *which captures the level of convergence in the* *opinions of the population; additionally,* ~~*and*~~ is *output updated* (e.g.: *draw histogram of the population's opinions; write each individual's opinion to a file*.)

Comments: This section briefly outlines the processes (or submodels) that the model runs through in every time step (ignoring initialisation), and in what order. Notice how each process is given an emphasized label, which corresponds to subsection headings in the Submodels section. Whilst the ODD protocol does not make such precise stipulations as to formatting, there should be a clear one-to-one correspondence between the brief outlines of processes here, and the details provided on each in the Submodels section.

In describing larger models than Deffuant et al.'s, it may be appropriate to simply present the process overview as a list. Many models have a simple schedule structure consisting of a repeated sequence of actions; such a list would clearly show this schedule. However, others use more complicated scheduling arrangements (e.g. dynamic scheduling). In such cases, the rules determining when new events are added to the schedule would need to be described, as well as an (unordered) list of event types, each corresponding to a subsection of 'Submodels'.

The 'schedule' in a declarative model may be even less clear, as it will depend on how the inference engine decides which rules to fire. However, declarative programs are at least asked a query to start the model, and this section would be an appropriate place to mention that. Some declarative programs also have an implied ordering to rule firing. For example, in Prolog, the rule a :- x, y, z. will, in the event that the inference engine tries to prove a, try to prove x, then y, then z. Suppose the model is started with the query ?- a. In describing the model here, it might suffice simply to summarise how x, y and z change the state of the model. Any subrules called by the inference engine trying to prove these could be given attention in the Details section.

The declarative programmer may also use language elements (such as cuts in Prolog) to manage the order of execution. In deciding which rules to describe here, a declarative modeller might focus on those changing the value of a state variable over time. The key point is that the program will do *something* to change the values of state variables over time in the course of its execution. Insofar as that can be described in a brief overview, it belongs here.

8.5.4 Design Concepts

***Basic principles**. – This model extends earlier stylised models on opinion dynamics, which either used only binary opinions instead of a continuous range of opinions, or where interactions only depended on whether opinion segments overlapped, but not on relative agreement (for references, see Deffuant* et al. 2002*).*

***Emergence**. – The distribution of opinions in the population emerges from interactions among the individuals.*

***Sensing**. – Individuals have complete information of their interaction partner's opinion and uncertainty.*

***Interaction**. – Pairs of individuals interact if their opinion segments, $[x - u, x + u]$, overlap.*

Stochasticity. *– The interaction between individuals is a stochastic process because interaction partners are chosen randomly.*
Observation. *– Two plots are used for observation: the histogram of opinions, and the trajectories of each individual's opinion. Additionally, a convergence index is calculated.*

Comments: Note that the design concepts are only briefly addressed. This would be expected in larger models too. Note also that several design concepts have been omitted because they are not appropriate to the model. Specifically, adaptation, objectives, learning, prediction, and collectives have been left out here: individuals change their opinion after interaction, but this change is not adaptive since it is not linked to any objective; there also no collectives since all individuals act on their own. Nevertheless, most models should be able to relate to some basic principles, emergence, interactions, and observation, and most often also stochasticity. Small models might use the option of concatenating the design concepts into a single paragraph to save space.

8.5.5 Initialization

Simulations are run with 1,000 individuals, of which a specified initial proportion, p_e, are extremists; p_+ denotes the proportion of 'positive' extremists, and p_- are the proportion of 'negative' extremists. Each moderate individual's initial opinion is drawn from a random uniform distribution between −1 and +1. Extremists have on opinion of either −1 or +1. Initially, individuals have a uniform uncertainty, which is larger for moderates than for extremists.

Comments: This explains how the simulation is set up before the main schedule starts. In other models, this might include empirical data of various kinds from, for example, surveys. The key question to ask here, particularly given the potential for confusion with the next section ('input data'), is whether the data are used *only* to provide a value for a state variable before the schedule runs.

8.5.6 Input Data

The model does not include any input of external data.

Comments: These are time-series data used to 'drive' the model. Some of these data *may* specify values for variables at time 0 (i.e. during initialisation); however, if a data series specifies values for any time step other than during initialisation, then it is input data rather than initialisation. It is also important not to confuse 'Input data' with parameter values.

8.5.7 Submodels

All model parameters are listed in the following table.

Parameter	Description
N	Number of individuals in population
U	Initial uncertainty of moderate individuals
μ	Speed of opinion dynamics
p_e	Initial proportion of extremists
p_+	Initial proportion of positive extremists
p_-	Initial proportion of negative extremists
u_e	Initial uncertainty of extremists

***Opinion interaction**. – This is run for an agent j, whose 'opinion segment' s_j is defined in terms of its opinion x_j and uncertainty u_j as:*

$$s_j = [x_j - u_j, x_j + u_j]$$

The length of the opinion segment is $2u_j$ and characterizes an individual's overall uncertainty.

In opinion interaction, agent j (the influenced, focal, or 'calling' individual) is paired with a randomly chosen agent, i, the influencing individual. The 'overlap' of their opinion segments, h_{ij}, is then computed as:

$$h_{ij} = min(x_i + u_i, x_j + u_j) - max(x_i - u_i, x_j - u_j)$$

This overlap determines whether in opinion interaction will take place or not: Agent j will change its opinion if $h_{ij} > u_i$, which means that overlap of opinions is higher than the uncertainty of the influencing agent (see Fig. 8.1).

For opinion interactions, the relative agreement of the two agents' opinions, RA, is calculated by dividing the overlap of their opinion segments (h_{ij}) minus the length of the non-overlapping part of influencing individual's opinion segment, ($2u_i$ – h_{ij}), and this difference divided by agent i's opinion segment length, $2u_i$ (Fig.8.1 depicts these terms graphically):

$$RA = (h_{ij} - (2u_i - h_{ij}))/2u_i = 2(h_{ij} - u_i)/2u_i = (h_{ij}/u_i) - 1$$

The opinion and uncertainty of agent j are then updated as follows:

$$x_j = x_j + \mu RA(x_i + x_j)$$

$$u_j = u_j + \mu RA(u_i + u_j)$$

Thus, the new values are determined by the old values and the sum of the old values of both interacting individuals multiplied by the relative agreement, RA, and by parameter μ, which determines how fast opinions change.

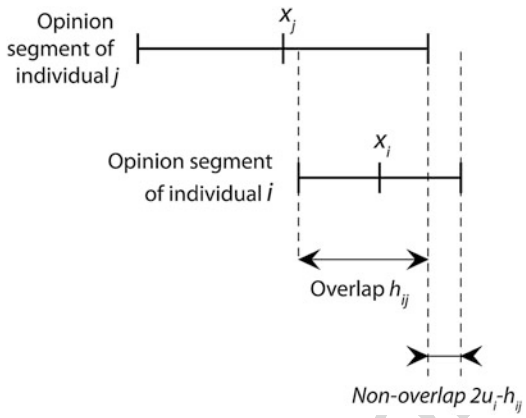

Fig. 8.1 Visualisation of the individual's opinions, uncertainties, and overlap in opinions in the model of (Deffuant et al.)

The main features of this interaction model are, according to Deffuant et al. *(2002):*

- *Individuals not only influence each other's opinions but also each other's uncertainties.*
- *Confident agents, who have low uncertainty, are more influential. This reflects the common observation that confident people more easily convince more uncertain people than the other way round – under the conditions that their opinions are not too different at the beginning.*

Calculate convergence index*. – This index, y, is used as a summary model output for sensitivity analysis and an exploration of the model's parameter space. It is defined as:*

$$y = q_{+} + q_{-}$$

where q_+ and q_- are the proportions of initially moderate agents which become extremists in the positive extreme or negative extreme, respectively. If after reaching the steady state none of the initially moderate agents became extremist the index would take a value of zero. If half of them become positive extremists and the other half becomes negative extremists, the index would be 0.5. Finally, if all the initially moderate agents converge to only one extreme, the index would be one. Note that for calculating y, "positive" or "negative" extreme has to be defined via *an interval close the extreme, with a width of, for example, 0.15.*

Comments: Here, details on the two processes described in Sect. 8.3 are provided, in sufficient depth to enable replication, i.e. *opinion interaction* and *calculate convergence index*. Note how these names match with those used in the process overview in Sect. 8.3.

Authors describing larger models may find journal editors protesting at the length of the ODD if all submodels are described in the detail required. There are various ways such constraints can be handled. One is to include the submodels in an appendix or supplementary material to the paper. Another is to provide them as a technical report accessible separately (e.g. on a website), and referred to in the text. If space is not too limited, a summary of each submodel could be provided in the main text, longer than the brief description in the process overview, but shorter than the full detail; the latter being provided separately. For very large models, or where space is highly constrained, there may be little room for much more than the three Overview sections in the journal article; again, making the full ODD available separately is a possible solution. Nevertheless, excluding the 'Submodels' element entirely from the main text should be avoided because this would mean to ask readers to accept, in the main text of the article, the model as a black box. Description of the most important processes should therefore be included also in the main text.

8.6 Discussion

Since the example model by Deffuant et al. (2002) is very simple, using ODD here comes with the cost of making the model description longer than the original one, through requiring the ODD labels. The original model is actually relatively clear and easy to replicate (which might partly explain this model's success). However, easy replication is much more the exception than the rule (Hales et al. 2003; Rouchier et al. 2008), and the more complex an ABM, the higher the risk that not all information is provided for unambiguous replication.

ODD facilitates writing comprehensive and clear documentations of ABMs. This does not only facilitate replication, it also makes writing and reading model documentations easier. Modellers no longer have to come up with their own format for describing their model, and readers know, once they are familiar with the structure of ODD, *exactly* where to look for what kind of information.

Whether or not to use ODD as a standard format for model descriptions might look like a rather technical question, but it has fundamental consequences, which go far beyond the issue of replication. Once ODD is used as a standard, it will be become much easier to compare different models addressing similar questions. Even now, ODD can be used to review models in a certain field, by rewriting existing model descriptions according to ODD (Grimm et al. 2010). Building blocks of existing models, in particular specific submodels, which seem to be useful in general, will be much easier to identify and re-use in new models.

Most importantly, however, using ODD affects the way we design and formulate ABMs in the first place. After having used ODD for documenting two or three models, you start formulating ABMs by answering the ODD questions: What 'things', or entities, do I need to represent in my model? What state variables and behavioural attributes do I need to characterize these entities? What processes do I want to represent explicitly, and how should they be scheduled? What are the spatial and temporal extent and resolution of my model, and why? What do I want to impose, and what to let emerge? What kind of interactions does the model include? For what purposes should I include stochasticity? How should the model world be initialized, what kinds of input data do I need, and how should I, in detail, formulate my submodels?

These questions do not impose any specific structure on simulation models, but they provide a clear checklist for both model developers and users. This helps avoiding "ad hoc-ery" in model design (Heine et al. 2005). Modellers can also more easily adopt designs of existing models and don't have to start from scratch all the time, as in most current social simulation models.

Criticisms of ODD include Amouroux et al. (2010), who, acknowledging its merits, find the protocol ambiguous and insufficiently specified to enable replication. This article pertained to the Grimm et al. (2006) first description of ODD. The update in Grimm et al. (2010) endeavoured to address issues such as these. However, the success of the latter article in so doing, and indeed any future revisions of ODD, can only be measured by comparing replication efforts based on ODD descriptions with those not conforming to any protocol – the norm prior to 2006 when ODD was first published. As suggested above, the record for articles not using ODD has not been particularly good: Rouchier et al. (2008) observe in their editorial to a special section of JASSS on the third Model-2-Model workshop that several researchers attempting replications have to approach the authors of the original articles to disambiguate model specifications. If the models were adequately described in the original articles, this should not be necessary.

Polhill et al. (2008) also observed that those used to object-oriented designs for modelling will find the separation of what will for them effectively amount to instance variables and methods (state variables and processes respectively) counter-intuitive, if indeed not utterly opposed to encapsulation: one of the key principles of object orientation. For ODD, however, it is the reader who is important rather than programming principles intended to facilitate modularity and code reuse. It is also important that, as a documentation protocol, ODD does not tie itself to any particular ABM implementation environment. From the perspective of the human reader, it is illogical (to us at least) to discuss processes before being informed what it is the processes are operating on. Encapsulation is about hiding information; ODD has quite the opposite intention.

The main issue with ODD in social simulation circles as opposed to ecology, from which it originally grew, pertains to its use with declarative modelling environments. This matter has been raised in Polhill et al. (2008), and acknowledged in Grimm et al. (2010). Here we have tried to go further towards illustrating how a declarative modeller might prepare a description of their model that conforms to ODD. However, until researchers using declarative environments attempt to use ODD when writing an article, and feedback on their findings, this matter cannot be properly addressed.

Certainly, ODD is not the silver bullet regarding standards for documenting ABMs. Nevertheless, even at the current stage its benefits by far outweigh its limitations, and using it more widely is an important condition for further developments. Still, since ODD is a verbal format, not all ambiguities can be prevented. Whilst a more formal approach using, for example XML or UML (e.g. Triebig and Klügl 2010, and for ABMs of land use/cover change, the MRPOTATOHEAD framework – Livermore 2010; Parker et al. 2008) might address such ambiguities, we consider it important that written, natural language formulations of ABMs exist (Grimm and Railsback 2005). This is the only way to make modelling, as a scientific activity, independent of technical aspects of mark-up or programming languages and operating systems. Further, verbal descriptions force us to *think* about a model, to try to understand what it is, what it does, and why it was designed in that way and not another (J. Everaars, *pers. comm.*). We doubt that a 'technical' standard for documenting ABMs – one that can be read by compilers or interpreters, would ever initiate and require this critical thinking about a model.

Nevertheless, it is already straightforward to translate ODD model description to NetLogo programs because much of the way models are written in NetLogo corresponds to the structure of ODD: the declaration of 'Entities, state variables, and scales' is done via NetLogo's globals, turtles-own, and patches-own primitives, 'Initialization' is done via the setup procedure, 'Process overview and scheduling' corresponds to the go procedure, 'Details' are implemented as NetLogo procedures, and 'Design concepts' can be included, (as indeed can the entire ODD model description), on the 'Information' tab of NetLogo's user interface.

8.7 Conclusion

Clearly describing simulations well, so that other researchers can understand a simulation is important for the scientific development and use of complex simulations. It can help in: the assessment and comprehension of simulation results by readers; replicating simulations for checking and analysis by other researchers; transferring knowledge embedded within simulations from one domain to another; and allowing simulations to be better compared. It is thus an important factor for making the use of simulations more rigorous and useful. A protocol such as ODD is useful in standardising such descriptions and encouraging minimum standards. As the field of social simulation matures it is highly likely that the use of a protocol such as ODD will become standard practice.

The investment in learning and using ODD is minimal but the benefits, both for its user and the scientific community, can be huge. We therefore recommend learning and testing ODD by re-writing the model description of an existing, moderately complex ABM, and, in particular, using ODD to formulate and document the next ABM you are going to develop.

Acknowledgements We are grateful to Bruce Edmonds for inviting us to contribute this chapter, and for his helpful comments and suggested amendments to earlier drafts. Gary Polhill's contribution was funded by the Scottish Government.

Further Reading

Railsback and Grimm (2012) is a textbook which introduces agent-based modelling with examples described using ODD. The OpenABM website (http://openabm.org) is a portal specifically designed to facilitate the dissemination of simulation code and descriptions of these using the ODD protocol. The original reference document for ODD is (Grimm et al. 2006) with the most recent update being (Grimm et al. 2010). Polhill (2010) is an overview of the 2010 update of ODD written specifically with the social simulation community in mind.

References

Amouroux E, Gaudou B, Desvaux S, Drogoul A (2010) O.D.D.: a promising but incomplete formalism for individual-based model specification. In: Ho TB, Zuckerman DN, Kuonen P, Demaille A, Kutsche R-D (eds) 2010 IEEE-RIVF international conference on computing and communication technologies: research, innovation and vision for the future, Vietnam National University, Hanoi, 1–4 Nov 2010

Bithell M, Brasington J (2009) Coupling agent-based models of subsistence farming with individual-based forest models and dynamic models of water distribution. Environ Model Software 24:173–190

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

Gilbert N (2007) Agent-based models. Sage, London

Grimm V, Railsback SF (2005) Individual-based modeling and ecology. Princeton University Press, Princeton

Grimm V, Railsback SF (2012) Designing, formulating, and communicating agent-based models. In: Heppenstall A, Crooks A, See LM, Batty M (eds) Agent-based models of geographical systems. Springer, Berlin, pp 361–377

Grimm V et al (2006) A standard protocol for describing individual-based and agent-based models. Ecol Model 198:115–126

Grimm V et al (2010) The ODD protocol: a review and first update. Ecol Model 221:2760–2768

Hales D, Rouchier J, Edmonds B (2003) Model-to-model analysis. J Artif Soc Soc Simulat 6(4). http://jasss.soc.surrey.ac.uk/6/4/5.html

Heine B-O, Meyer M, Strangfeld O (2005) Stylised facts and the contribution of simulation to the economic analysis of budgeting. J Artif Soc Soc Simulat 8(4). http://jasss.soc.surrey.ac.uk/8/4/4.html

Heppenstall A, Crooks A, See LM, Batty M (eds) (2012) Agent-based models of geographical systems. Springer, Berlin

Livermore M (2010) MR POTATOHEAD framework: a software tool for collaborative land-use change modeling. In: Swayne DA, Yang W, Voinov AA, Rizzoli A, Filatova T (eds) International Environmental Modelling And Software Society (iEMSs) 2010 international congress on environmental modelling and software: modelling for environment's Sake, Fifth Biennial Meeting, Ottawa. http://www.iemss.org/iemss2010/index.php?n=Main,Proceedings

Parker DC et al (2008) Case studies, cross-site comparisons, and the challenge of generalization: comparing agent-based models of land-use change in frontier regions. J Land Use Sci 3(1):41–72

Polhill JG (2010) ODD updated. J Artif Soc Soc Simulat 13(4). http://jasss.soc.surrey.ac.uk/13/4/9.html

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. J Artif Soc Soc Simulat 11(2). http://jasss.soc.surrey.ac.uk/11/2/3.html

Polhill JG, Gimona A, Aspinall RJ (2011) Agent-based modelling of land use effects on ecosystem processes and services. J Land Use Sci 6(2–3):75–81

Railsback SF, Grimm V (2012) Agent-based and individual-based modeling: a practical introduction. Princeton University Press, Princeton

Rouchier J, Cioffi-Revilla C, Polhill JG, Takadama K (2008) Progress in model-to-model analysis. J Artif Soc Soc Simulat 11(2). http://jasss.soc.surrey.ac.uk/11/2/8.html

Triebig C, Klügl F (2010) Elements of a documentation framework for agent-based simulation. Cybern Syst 40(5):441–474

Wheeler S (2005) Beyond the inverted pyramid: developing news-writing skills. In: Keeble R (ed) Print journalism: a critical introduction. Routledge, Abingdon, pp 84–93

Wilensky U (1999) NetLogo. http://ccl.northwestern.edu/netlogo