This is the preprint of the contribution published as:

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

The publisher’s version is available at:

http://dx.doi.org/10.1016/j.envsoft.2014.01.029

Standardised and transparent model descriptions for agent-based models: current status and prospects

Birgit Müller[a,*], Stefano Balbi[b], Carsten M. Buchmann[c], Luís de Sousa[d], Gunnar Dressler[a], Jürgen Groeneveld[a,e], Christian J. Klassert[f], Quang Bao Le[g], James D.A. Millington[h], Henning Nolzen[a], Dawn C. Parker[i], J. Gary Polhill[j], Maja Schlüter[k], Jule Schulze[a], Nina Schwarz[c], Zhanli Sun[l], Patrick Taillandier[m], Hanna Weise[a]

[a]UFZ, Helmholtz Centre for Environmental Research - UFZ, Department Ecological Modelling, Permoser Str. 15, 04138 Leipzig, Germany, birgit.mueller@ufz.de (BM), gunnar.dressler@ufz.de juergen.groeneveld@ufz.de (JG), henning.nolzen@ufz.de (HN), jule.schulze@ufz.de (JS), hanna.weise@ufz.de (HW)

[b]Basque Centre for Climate Change (BC3) Alameda Urquijo 4, 4º, 48008 Bilbao, Spain, stefano.balbi@bc3research.org

[c]UFZ, Helmholtz Centre for Environmental Research - UFZ, Department Computational Landscape Ecology, Permoser Str. 15, 04138 Leipzig, Germany, carsten.buchmann@ufz.de (CB), nina.schwarz@ufz.de (NS)

[d]Resource Centre for Environmental Technologies, Public Research Centre Henri Tudor, Technoport Schlassgoart, 66 rue de Luxembourg, P.O. BOX 144, L-4002 Esch-sur-Alzette, Luxembourg, luis.a.de.sousa@gmail.com

[e]School of Environment, University of Auckland, Private Bag 92019, Auckland, New Zealand

[f]UFZ - Helmholtz Centre for Environmental Research - UFZ, Department of Bioenergy, Torgauer Str. 116, 04347 Leipzig, Germany, christian.klassert@ufz.de

[g]Natural and Social Science Interface (NSSI), Institute for Environmental Decisions (IED), ETH Zurich, Universitaetstrasse 22, CH-8092 Zurich, Switzerland, quang.le@env.ethz.ch

[h]Department of Geography, King's College London, Strand, London, WC2R 2LS, UK, james.millington@kcl.ac.uk

[i]School of Planning, University of Waterloo, 200 University Avenue West, Waterloo, Ontario, Canada, dcparker@uwaterloo.ca

[j]The James Hutton Institute, Craigiebuckler, Aberdeen. AB15 8QH. United Kingdom, gary.polhill@hutton.ac.uk

[k]Stockholm Resilience Centre, Stockholm University, Kräftriket 2B, 106 91 Stockholm, Sweden, maja.schlueter@stockholmresilience.su.se

[l]Leibniz Institute of Agricultural Development in Central and Eastern Europe (IAMO), Theodor-Lieser-Str. 2, 06120 Halle (Saale), Germany, sun@iamo.de

[m]Department of Geography, UMR IDEES, CNRS, University of Rouen, 7 rue Thomas Becket, 76130 Mont Saint Aignan, France, patrick.taillandier@univ-rouen.fr

***Corresponding author:** Phone +49 341 235 1708 / Fax +49 341 235 1473

Preprint Version of:
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. and H. Weise. Standardized and transparent model descriptions for agent-based models: current status and prospects. Environmental Modelling & Software 55(0) 156-163.
DOI: http://dx.doi.org/10.1016/j.envsoft.2014.01.029

Abstract:

Agent-based models are helpful to investigate complex dynamics in coupled human-natural systems. However, model assessment, model comparison and replication are hampered to a large extent by a lack of transparency and comprehensibility in model descriptions. In this article we address the question of whether an ideal standard for describing models exists. We first suggest a classification for structuring types of model descriptions. Secondly, we differentiate purposes for which model descriptions are important. Thirdly, we review the types of model descriptions and evaluate each on their utility for the purposes. Our evaluation finds that the choice of the appropriate model description type is purpose-dependent and that no single description type alone can fulfil all requirements simultaneously. However, we suggest a minimum standard of model description for good modelling practice, namely the provision of source code and an accessible natural language description, and argue for the development of a common standard.

Keywords:

Agent-based modelling; domain specific languages; graphical representations; model communication; model comparison; model development; model design; model replication; standardised protocols

1. Introduction

Agent-based models are argued to be helpful to investigate complex dynamics in coupled human-natural systems (Hare and Deadman, 2004; Liu et al., 2007; Balbi and Giupponi, 2010, Filatova et al., 2013). However, the production of research using agent-based modelling has not been as efficient as it could be up to now. Reasons include that model assessment, replication, and comparison are hampered to a large extent by a lack of transparency in model descriptions. Further, code developed for one project is rarely reused for other projects, even for closely related research. To overcome these problems, standardised model description protocols, ontologies and graphical representations have been created. The various model description types have been developed to achieve different purposes, including facilitation of in-depth model comprehension, assessment, replication, design and communication.

In this contribution we address the question of whether an ideal standard for describing agent-based models exists. We first present a classification of the prevalent types of model descriptions and give an overview of their different purposes. We then review available model description types, evaluating each on its utility for the different purposes. Finally, we discuss advantages of combining these different types, suggest a minimum standard of model description for good modelling practice and discuss future challenges. Note that we set the focus on providing an adequate description of the model itself and not on the description of model results. Appropriate documentation of the model results is beyond the scope of this paper (but see "Transparent and comprehensive ecological modeling (TRACE) documentation" in Schmolke et al. (2010), pp. 482 which suggests a standard for all parts of the modelling process).

The idea for this article came about at a workshop at the 6th International Congress on Environmental Modelling and Software (iEMSs) 2012 in Leipzig, Germany, and the article reflects the perspectives of the participants, who are members of the integrated social and environmental modelling communities.

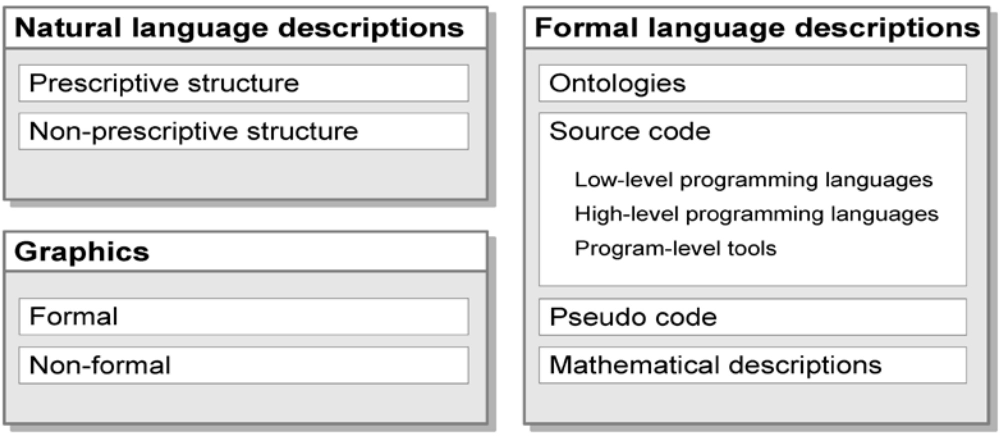

Figure 1

2. Current state of the art: different types of model descriptions in use

We classify the prevalent types of model descriptions in three categories: natural language descriptions, formal language descriptions and graphics (cf. Figure 1 for an overview). In the following paragraphs the different description types are briefly outlined:

Natural language descriptions present models in everyday language with or without a prescriptive structure. The **prescriptive approach** divides the model description into categories, each explaining a particular part of the model. One example of such an approach is the ODD protocol (cf. Grimm et al., 2010 and its extension to include a description of human decisions in ABMs, ODD+D in Müller et al., 2013). ODD describes the model in a hierarchical way using three main categories: *Overview*, *Design concepts* and *Details* that are themselves subdivided into several subcategories such as (in the case of design concepts) *sensing* or *interaction*. ODD is being widely used for the description of ABMs (for examples see Balbi et al., 2013; Caillault et al., 2013; Marohn et al., 2013; Smajgl and Bohensky, 2013). In contrast, a **non-prescriptive natural language description** puts no constraints regarding content and form of the model description on the author (see exemplary model descriptions in Becu et al., 2003 and Deadman et al., 2004). Furthermore, non-prescriptive descriptions can also be used to present the source code in a more intuitive way. Examples are literate programming (cf. Knuth, 1984), documentation generators such as Doxygen or Javadoc that assemble source code comments into a structured document, or, in principle, any form of source code documentation that uses natural language.

Formal languages describe models in an abstract and self-consistent way with formal syntax and semantics that avoid ambiguity. Model descriptions written in formal languages may therefore be used to describe important aspects of a model specifically. Formal languages that we consider here include ontologies, source code, pseudo code and mathematical descriptions.

An **ontology** can be defined as "an explicit specification of a conceptualization" (Gruber, 1993, p. 199) that describes entities and their structural interrelationships, often using a hierarchical categorisation. They specifically allow logical inferences to be drawn. Various formal languages are available for writing ontologies – OWL (Web Ontology Language) being currently the most popularly used (Horrocks et al., 2003; Grau et al., 2008). OWL has been argued to improve the transparency of formal descriptions of model structure in comparison with source code, since the latter is focused on programmer and compilation convenience rather than using logics to reflect common-sense perceptions (Polhill and Gotts, 2009). One example of ontologies applied to agent-based modelling is that of Christley et al. (2004). A second example is the MR POTATOHEAD ontology developed by Parker et al. (2008), which describes the components that appear in agent-based models of land use/cover change. It identifies key model elements and their alternative instantiations, based on a broad review of models. MR POTATOHEAD has an OWL implementation which facilitates evaluating conceptual completeness.

Providing **source code** is another formal way to communicate models. The following subcategories are listed according to their readability, from cryptic to simple-to-read. **Low-level programming languages** (e.g. assembly language) are characterized by their strong linkage to the computer's hardware and are often platform-dependent. Though unlikely to be used for an entire ABM implementation, these can be useful for computationally intensive functions where bespoke code improves on compiler optimisation. Assembly language is necessary where higher-level programming language libraries are not available for specialised hardware operations. For example, it is common in Linux distributions not to provide C libraries for accessing floating point arithmetic utilities stipulated by the IEEE 754 (1985) standard (IEEE, 1985). Polhill and Izquierdo, 2005 , footnote 2) note that implementation of these utilities using assembly language is necessary in a Cygwin environment.[1]

High-level programming languages in their basic form are platform-independent (especially where governed by standards) and improve the readability for the user by providing algorithmic constructs such as loops or conditional statements. Popular examples of high-level programming languages are Java and C++. In addition, **program-level tools** extend the functionality of high-level programming languages by "providing useful software libraries for building specific classes of models" (de Sousa and da Silva, 2011, p. 170) and can further improve the readability of the source code. Usually they are tailored to specific fields of modelling. They can be further distinguished into tools that provide a domain-specific language (e.g. NetLogo (Tisue and Wilensky, 2004) or GAMA (Taillandier et al., 2012) for agent-based modelling) and tools where the user has to write the application in a high-level programming language (e.g. Repast (North et al., 2007) or Mason (Luke et al., 2005)). In any case, provision of source code has been argued to be a necessary condition of maintaining good scientific practice in the publication of simulation models (e.g. Polhill and Edmonds, 2007).

Pseudo code is a structured description of the model combining natural language elements with formal language constructs (e.g. loops). Gilberg and Forouzan (2004), p. xii, define pseudo code as "[natural language]-like presentation of the steps needed to solve a problem. It is written with a relaxed syntax ... that hides the detail [allowing the reader] to concentrate on the problem requirements." While such definitions can be given, to the knowledge of the authors no common standard on formulating pseudo code exists. Indeed, the provision of such a standard has been argued against on the basis that it would then become another programming language (Wikipedia, 2013), though there are stylistic conventions (e.g. Smed and Hakonen, 2006), especially for operators and control statements. The advantage of pseudo code is that it is independent of the programming language and therefore the knowledge of a specific programing language is not required to read and understand the code (for examples cf. Roy, 2006; Perez and Dragicevic, 2010, p.227; Robinson et al., 2013, p. 134).

Mathematical descriptions provide an exact way to depict model processes and states, usually with formulated equations composed of strings of symbols. While being suited to describe quantitative properties of the model, they are not able to communicate specific model concepts, such as underlying theories or process scheduling. Mathematical descriptions can range from general descriptions of model states (see equation (1) for the calculation of the willingness to pay (WTP)) to specific equations (see equation (2), cf. Filatova et al., 2009, section 3.12).

$$WTP = f(utility, income) \tag{1}$$

$$WTP = \frac{utility \cdot income}{b^2 \cdot utility^2}, \text{with } b = \text{constant factor} \tag{2}$$

[1] The utilities they implemented for this purpose are now available at https://github.com/garypolhill/ieeefp

Graphics use particular visualization techniques to illustrate processes, structures, relationships, program flows, etc. They particularly support the understanding of qualitative properties of the model, such as its structure. Graphics can be either **formal**—strictly adhering to pre-defined rules or protocols, such as UML (Unified Modeling Language, cf. Object Management Group, 2011 and examples for class diagrams, activity diagrams and sequence diagrams in Polhill et al., 2013), **or non-formal**—following loosely-defined principles or conventions, such as flow charts (cf. International Organization for Standardization (ISO), 1985, examples in van Oel et al., 2010; Zhang et al., 2011), Bayesian belief networks (Jensen, 2001; Aalders, 2008; Sun and Müller, 2013), decision trees (Quinlan, 1986), cognitive maps (Eden, 1988; Kitchin, 1994) or causal loop diagrams (Maruyama, 1992). Non-formal graphics are often used to get a first impression of the model concept (for examples see Haase et al., 2010; Rebaudo and Dangles, 2013).

It should be noted that the different description types outlined above are not necessarily mutually exclusive. For instance UML, as part of the graphics category, is also an ontology. At the same time, the MR POTATOHEAD ontology can be visualized using various graphical approaches. However from our point of view a separate category "graphics" underpins the importance of visualisation approaches for instance for the communication with stakeholders or scientists from other disciplines.

3. Does an ideal standard for describing models exist?

3.1. Different purposes of model descriptions

Model descriptions can enable their users to meet various different but related purposes as the models themselves (Kelly et al., 2013). We propose the following as the most important purposes: model communication, in-depth model comprehension, model-assessment, -development, -replication, -comparison, theory building and code generation (note some overlap occurs between purposes.). Here we briefly describe each of these purposes to facilitate our review of model descriptions and the discussion of an appropriate standard (the order chosen reflects the level of generality, i.e. from general and universal to rather specific purposes):

Communication of the model: Model descriptions serve as a means of communication of the model to an audience that can consist of scientists as well as stakeholders or people from outside the research domain. These groups may need different information (e.g. methodological details for specialists versus basic information on the model's entities and processes for stakeholders). They also need to be approached in different ways, which poses a particular communication challenge. One example is the interaction with stakeholders which may benefit particularly from the use of easily understandable visual tools. Apart from that: The communication between model designer of the conceptual model and programmer necessary for model implementation is a special but important case of model communication.

In-depth model comprehension: This is the prime motivation for model descriptions. The challenge is to allow a profound and complete understanding of the model's entities and processes. Ideally, the reader can also relate real-world concepts to the model. Model understanding is the precondition for most of the other purposes. A detailed and thorough description of all model components is thus essential. We emphasise here the degree of detailed model understanding required by other scientists, experts and reviewers (in contrast to the purpose of communication which does not necessarily imply overarching comprehension).

Model assessment: Assessment here means an evaluation of the model with respect to its suitability to answer the research question, the consistency between model design and sub-models, as well as chosen spatial and temporal scales. This purpose is particularly relevant for reviewers, but also researchers or policy makers who want to use model results. To fulfil this purpose the respective information (on research question(s), model purpose, design and scales) must be given together with a clear statement of the underlying model assumptions.

Model development: Following a prescriptive model description process can improve model development ('model design'), particularly when the model description is elaborated in parallel to the model design. Ideally, describing the model helps the modeller to adopt another (external) point of view of the model and can act as a check list for completeness. To achieve this, a model description should follow a concise and strict structure which obliges the author to describe all relevant aspects. A structured model description can also facilitate and give guidance to the development of models jointly with stakeholders ('collaborative model development'). In the case where the model description is expected to assist model development up to the final implementation, a rigorous protocol taking into account software and implementation related issues is particularly helpful.

Model replication: Adequate model descriptions can enable model replication. However, different levels of replication (see Wilensky and Rand, 2007) may pose different requirements for the model description. Exact quantitative replication of the results (exactly the same numbers) requires much more detailed information than statistical or qualitative replication. One extreme example for such detailed information is the random seed used if the model includes stochasticity. We use the term 'qualitative' to refer to replication that produces similar behaviour to the original model and is robust to implementation details like random number generators or the hardware used. Different metrics in model performance assessment (see Bennett et al., 2013) can be used for assessing behaviour similarity among the outputs of model replication. This usage is akin to the concept of 'distributional equivalence' in Wilensky and Rand (2007).

Model comparison: Model descriptions can facilitate model comparisons with respect to concepts and techniques (for example fitting or optimization algorithms). This can allow the reader to also evaluate which model(s) are more or less appropriate for investigating certain questions. To achieve this, a strict and complete set of criteria (e.g. aim, scales, and processes) for comparison should be part of the model descriptions, preferably in a standardised structure and format.

Theory building: By communicating the ideas behind a model, a description can also aid in theory building. The challenge and prerequisite here is to embed the respective model (concept) into the existing pool of theories and theoretical concepts. This can be attained most easily when the description standard obliges the author to summarize questions, applied theories, concepts, principles and hypotheses. Such an evaluation of the described model can reveal a lack of theoretical foundation for the model. Thereby, model descriptions can support models in their role of challenging existing theory (cf. Epstein, 2008). This, in turn, together with evaluation of model results, can facilitate the creation and assessment of new theoretical concepts and even new theory.

Code generation: Formal languages and graphic-based model descriptions in particular can support model implementation through (automated) source code generation. In case of model reimplementation, code regeneration here implies a higher level of formality, compared to most levels of model replication (see above). Code (re)generation can be achieved if a description standard is formal, complete, and exact (not allowing any ambiguities), while containing accurate information on entities and processes plus their translation into code characteristics like classes and methods. In this case, specialised software can then directly generate the basic (code) structure from the model description alone.

3.2. Matching purposes and types

In the following, we assess how well the purposes are met by the different description types. For the assessment, we focus on the potential of the description type rather than on how it is realised in practice in our experience. The various description types fulfil the purposes mentioned in the previous section to different degrees (see Table 1 for an overview and Table A1 in the online appendix for further details).

Communication of the model needs to be differentiated for different target groups. (a) Communication for peers is achieved with most of the description types, given the knowledge of the respective description approach/programming language. (b) Communication for education purposes is improved by e.g. natural language descriptions, OWL, usage of program-level tools, pseudo code and mathematical descriptions as well as non-formal graphics. (c) Communication to stakeholders should not be too technical, thus the suitable description types are limited to natural language descriptions, OWL, and non-formal graphics. Non-formal graphics are the only description type that can foster communication for all target groups, due to their large flexibility.

In-depth model comprehension that includes profound understanding of model entities and processes is fostered by natural language descriptions, particularly with prescriptive structure (such as the ODD protocol), but also by formal language descriptions (i.e. OWL ontology, source code in a high-level programming language and of program-level tools) as well as different types of graphics. While being suited to describe quantitative properties of the model, mathematical descriptions alone are not able to communicate specific model concepts, such as underlying theories or process scheduling.

Model assessment in general is facilitated by structured natural language descriptions, ontologies and all types of graphics, i.e. UML, and non-formal graphics such as cognitive maps, decision trees and the like. Some specific types of model assessment - such as checking the consistency between model design and sub-models - can more easily be carried out with ontologies or formal graphics.

Model development: (a) Model design by modellers and programmers is aided with prescriptive natural language descriptions such as ODD, ontologies and usage of program-level tools, pseudo code or mathematical descriptions; both formal and non-formal graphics are also helpful. However, usage of non-formal graphics and also program-level tools as a check list for the model design process is limited, depending on the specific tool chosen. (b) Collaborative development together with stakeholders on the contrary is eased by non-prescriptive natural language descriptions and formal and non-formal graphics.

(Quantitatively exact) **model replication** is difficult based on mathematical descriptions without the provision of source code in program-level tools—equations alone do not guarantee replications (as discussed in section 3.1). Usually detailed information of the specific analytical or numerical procedures needs to be provided in form of script similar to high-level source code. However, we want to highlight that although the provision of source code technically facilitates model replication, it may circumvent the consistency check between the conceptual model and its implementation (one purpose of model replication) by encouraging "replicators" to simply copy the source code. In addition, for independent model replication it is necessary to know the intention of the modeller and therefore, we suggest that for independent model replications, both source code and a natural language description are provided. Qualitative model replication may be achieved also with other model descriptions, such as MR POTATOHEAD, ODD, UML and non-formal graphics.

Model comparison is made easier with prescriptive natural language descriptions, ontologies, usage of program-level tools and the provision of mathematical descriptions, if parameters are also provided. Again, the specific focus of a model comparison will indicate descriptions that are most suitable from this list (e.g. comparison of the conceptual basis might be easier using prescriptive natural language descriptions or ontologies rather than mathematical descriptions).

Theory building is not well facilitated by model descriptions, as most model description types do not ask for the theoretical background of the model, hypotheses to be tested, etc. One exception is the prescriptive structure of the ODD+D protocol (Müller et al., 2013), which asks for the theories underlying the model; while theories are up to now not explicitly listed in the "basic principles" section of the ODD protocol (Grimm et al., 2010). Another example is presented in Schlüter et al., in press. They develop a procedure to document the theoretical background, the hypotheses and the assumptions on which a model conceptualization is based in a structured way. Furthermore, non-formal graphics are able to convey information relevant for theory building and can thus facilitate theory building without the constraints of formalised graphics.

Code generation in the sense of automatically generating code is enabled by formal graphics, such as UML (cf. Bersini, 2012) or program-level tools facilitating generation of system models such as SIMULINK or STELLA as well as OWL program-level tools. Apart from that, pseudo-code is often used to generate the structure of the programme (e.g. Roy, 2006).

Table 1: Assessment on how well the purposes are met by the different description types (light grey -limited ability, middle grey - medium ability, dark grey - high ability, x not applicable)

	Natural language		Formal language							Graphics	
	Prescriptive structure	Non-prescr. structure	Ontologies		Source code			Pseudo code	Mathematical descriptions	Formal	Non-formal
	ODD	Verbal descript.	OWL	MR POTA-TOEHEAD	Low-level progr. lan.	High-level progr. lan.	Program-level tools			UML	Exampl.[2]
Commun. to peers	high	medium	medium	high	limited	high	high	medium	high	high	high
Commun. for education	medium	high	high	medium	limited	medium	high	high	high	medium	high
Commun. for stakeholders	limited	high	high	medium	limited	limited	limited	medium	medium	medium	high
In-depth comprehen.	high	medium	medium	high	limited	high	high	medium	medium	high	high
Assessment	high	medium	high	medium	limited	medium	medium	medium	medium	high	high
Developm.: design	high	limited	high	high	limited	medium	high	high	high	high	high
Developm.: collaborative	medium	high	high	medium	limited	limited	medium	medium	medium	high	high
Replication	medium	limited	limited	medium	medium	medium	high	limited	high	medium	medium
Comparison	high	medium	high	high	limited	medium	high	medium	high	medium	medium
Theory building	medium	high	medium	medium	limited	limited	limited	medium	medium	medium	high
Code generation	limited	limited	high	limited	x	x	high	limited	limited	high	limited

[2] Non-formal graphics examples: Cognitive maps, Bayesian belief networks, decision trees, flow charts, causal loop-diagrams

3.3. One size does not fit all

Our main conclusion from the analysis conducted above is that the choice of a model description standard is purpose-dependent and that no single model description type alone can fulfil all requirements simultaneously. We have identified conflicting objectives: a) to achieve a detailed model description that enables model replication and b) to provide a concise and easy to communicate model description. Furthermore, one should avoid making the recommendations for model description more demanding than necessary for its purpose. This is important for making the recommendations useful to a wide range of authors, which seems to be a precondition for the establishment of a common standard of model description.

Although the provision of information (especially source code) is sometimes hindered by legal or other institutional reasons (Ince et al., 2012; Polhill and Edmonds, 2007), we consider it important to make the **source code** of a model available for three reasons: firstly, it is the definitive implementation of the model, not subject to ambiguities, omissions or inaccuracies associated with verbal descriptions, secondly, because it provides the most direct means of replicating model experiments, and thirdly, because it is necessary to allow others to identify shortcomings in the implementation chosen by the author. The website www.openabm.org, for example, provides an archive where model files can be uploaded to share source code and/or model implementations.

Natural language description, especially when formalized in **standard protocols, such as ODD** (Grimm et al., 2010 or its extension focusing on human decision-making, ODD+D, Müller et al., 2013), helps to make a connection between verbal descriptions of the real world system underlying the model and the model itself. These standard protocols can inform the scientific community whether and how the model itself meets minimum scientific standards, and what additional aspects or capabilities the model requires to meet its specific purposes. In our view, the elements which should be documented in the protocol in order to meet a minimum standard of model description fall into two main categories: those that are always needed to describe any system models (e.g. goal statement, context/boundary setting conditions, unit and scale of analysis), and those that are specific for an ABM, such as minimal characteristics of the encoded agents (e.g. 'heterogeneous', 'autonomous', 'interactive', 'reactive' and 'adaptive', cf. Benenson and Torrens, 2004), or system properties that ABMs are usually designed to explain (e.g. 'emergence' and 'adaptation', cf. Holland, 1995; Bonabeau, 2002).

Standard protocols tell authors the information they need to include in their model description, and they prime readers' expectations regarding what information they will find where. For readers, this can facilitate the understanding of the assumptions made in the implementation of a model. It further requires authors to more fully open the "black box" of their model, potentially revealing its weak areas and better contributing to scientific progress. Therefore, we consider the use of a standard description important.

In addition, graphical representations, such as UML diagrams, can facilitate various purposes of model communication and understanding, and informal graphical representations are especially beneficial for educational purposes and when working with stakeholders. The close links between certain graphical representations and ontologies present a strong argument for the use of ontologies in the model design phase (cf. Livet et al. (2010) for the potential of ontologies for model building). However, the particular mode of graphical representation to use is sensitive to the model in question and to the intended audience. Hence, we do not recommend it as a minimum standard for model description in journal articles, but as an optional augmentation to the text and source code.

4. Ways ahead

We recommend that researchers build on current examples of good model descriptions, not only to describe their models transparently, but also to strive for common standards in describing ABMs, in order to contribute to comparability, model assessment and replication, and theory development. However, many open questions and challenges also remain, which need to be addressed to improve model descriptions in the future.

Firstly, the standardisation of model descriptions is impeded by the fact that it is extremely difficult to find a consistent terminology across the many disciplines to which agent-based modelling is applied (cf. Balbi and Giupponi, 2010). Researchers on multi- and inter-disciplinary projects often report that differences in terminology and vocabulary are an impediment to mutual understanding (McConnell et al., 2011). A standardised description has the potential to promote the use of a common terminology, through suggesting and defining terminology such as agent and emergence by a standardised description protocol, examples and guiding questions therein.

Secondly, while there is a lot of common understanding about the purposes of model descriptions, there are some aspects on which we have found different perspectives. A major issue is whether natural language description standards need to be detailed enough to allow for replication, or whether such standards should only facilitate understanding and communication, leaving replication to the availability of the source code. On the one hand, if we emphasize the view that natural language descriptions are necessary to assess the consistency between the model and the real world, then this might be an argument to make standardised descriptions comprehensive enough to allow for replication. On the other hand, the question arises of whether such comprehensive descriptions might come at the cost of losing the readability of the documentation when models are very large and/or complex. Solutions to this might come from hierarchical natural language description such the ODD (starting from an overview and providing details later) or distinguishing between different levels of replication (numerical, statistical, and qualitative) and from developing large models over the course of several articles (Polhill et al., 2008; Grimm et al., 2010). However, the debate has not yet come to a conclusion.

In addition, although we have focused on model descriptions in this article, there are similar challenges for the description of model results. One attempt to address this issue is the use of narrative approaches which, for a working model, can be useful to illustrate characteristic (and specific) interactions between model agents and explain how these interactions produce system-level dynamics (Millington et al., 2012). Information about model outputs may be relevant for theory building; for example, documentation about hypotheses tested by the model and their results, or the results of global sensitivity analysis.

Thirdly, there are institutional and cultural issues surrounding the adoption and spread of standards. Should authorities promote standards (e.g. by journals making them a publication requirement; Polhill, 2010), or should they spread in an emergent process? Another aspect is that the pressure for providing transparent model descriptions might be greater if replicating a model to assess the reliability of its results were a more common practice in the ABM scientific arena. However, several institutional and cultural factors impede such a development: journals do not insist on licenses that enable software reuse, employers have an interest in protecting intellectual property rights, there are no standard libraries for ABM, and replication is not seen by everyone as innovative research. Further, model replication is a resource-intensive undertaking, and in an era of shrinking research budgets and university funding, it may not be practical.

Finally, there have been several attempts to ease communication between modellers and facilitate reuse of models and model components. Such reuse is seen as potentially decreasing start-up costs and reducing barriers to entry to modelling, thus increasing efficiency and speed of scientific progress in the field (Alessa et al., 2006). Common platforms for ABMs and model-level tools have been developed with these goals in mind. Contrary to program-level tools, model-level tools allow "the usage of … simulation models without requiring programming. These are pre-programmed models, designed for specific application fields that can be parameterized by the user." (de Sousa and da Silva, 2011, p. 170). Model-level tools can greatly facilitate communication since each model can be described simply by its set of parameters and inputs, using a single standard implementation. On the negative side this sort of tools tends to be highly specialised, filling narrow market niches; thus they are usually commercial tools whose internal implementation may not be open to independent scrutiny.

Although some platforms and program-level tools such as Repast have become popular, there are still a plethora of different platforms being used by the ABM developer community. The issue of how to increase research efficiency in the field by helping to facilitate code reuse remains an important one, but the large number of platforms impedes developments in this direction to some extent, and there seems to be no tendency towards agreeing on one common platform that is used by everyone. Therefore, we suggest it is worth working on a platform-independent standard for model description, especially as such a standard should remain relevant and useful even if common code bases are adopted in the future.

The process of establishing such a platform-independent standard for model descriptions could be inspired by successful attempts to establish standards in other domains: Model-Driven Development (Selic, 2003) is an emerging approach proposing the creation of domain specific lexica allowing for the simultaneous development and documentation of models. It has proved successful in domains parallel to ABM, notably with the SysML and ModelicaML languages, thus pointing to a further avenue for standardisation. To mention a second example, the Object Management Group (OMG) is a non-for-profit organisation of the software industry that has developed several widely adopted standards such as UML (Object Management Group, 2011), while over the years evolving a rather intricate internal structure with a multi-tier hierarchy and multiple ad-hoc boards at top level. A third example is the Open Geospatial Consortium (OGC), which involves the public sector, academia and industry to develop standards for spatial data publication and sharing (Castronova et al., 2013). These success stories highlight two points: First, some sort of formal and well-defined organisation is needed to drive the process of specification and later on the diffusion of the standard. Second, well-defined standards that address objective problems tend to be swiftly adopted by software developers and the industry in general. In the context of ABM, which is a domain still somewhat restricted to academia, an organisation like the OGC seems more suitable. This kind of organisation may be simpler for a small number of volunteers to start working on a draft standard, drawing on platforms such as www.openabm.org.

5. Conclusion

We have identified eight main purposes of model descriptions and summarized our expert knowledge in an assessment of the suitability of description types for different purposes. We conclude that no single model description type alone can fulfil all purposes simultaneously. Hence, we suggest a minimum standard consisting of a structured natural language description plus the provision of source code. Such description frame is particularly important for academic purposes, favouring in-depth model comprehension and model assessment. This echoes comments by other authors (e.g. Ince et al., 2012; Polhill and Edmonds, 2007) that good modelling practice entails both the provision of source code and an accessible natural language description, ideally following a formalized standard such as ODD (Grimm et al., 2006; Grimm et al., 2010). However, other description types can strengthen model description substantially in regard to specific purposes or target groups. For instance, graphics are appropriate to facilitate the model communication, while ontologies can foster model comparison and mathematics can improve the possibilities of replication.

Every author should therefore tailor the usage and weight of one or more description types according to the characteristics of the model and the purpose of the publication, in the view of meeting the above mentioned minimum standard. A joint effort of the ABM community towards transparent and comprehensible model descriptions through the use of standards would lead to a significant advancement of the field by enhancing exchange of information between peers and improving communication with model end-users. Therewith, the potential of agent-based modelling to support problem-oriented analysis and governance of human-natural systems would strongly increase.

Acknowledgments

We thank all participants of the workshop "Human decisions in agent-based models (ABM) for natural resource use - need for protocols" at the 6th International Congress on Environmental Modelling and Software (iEMSs) 2012 in Leipzig for their valuable input to the discussions and Volker Grimm and three anonymous reviewers for helpful comments on an earlier version of this manuscript.

Financial support from the DFG (German Research Foundation) in the framework of the Collaborative Research Centre 586 is gratefully acknowledged by BM. JGP's contribution was funded by The Scottish Government Rural Affairs and the Environment Portfolio Strategic Research Theme 1 (Ecosystem Services). JM's contribution was funded by a Leverhulme Early Career Fellowship. MS acknowledges funding from the European Research Council under the European Union's Seventh Framework Programme (FP/2007-2013) /ERC grant agreement no 283950 SES-LINK. ZS's contribution was supported by I-REDD+ project funded by the European Community's Seventh Framework Research Programme. DP's contribution was supported by the Canadian Social Science and Humanities Research Council grant 410-2011-1340.

References

Aalders, I., 2008. Modeling Land-Use Decision Behavior with Bayesian Belief Networks. Ecology and Society 13.

Alessa, L.N.i., Laituri, M., Barton, M., 2006. An "all hands" call to the social science community: establishing a community framework for complexity modeling using agent based models and cyberinfrastructure. Journal of Artificial Societies and Social Simulation 9 (4), 14.

Balbi, S., Giupponi, C., 2010. Agent-Based Modelling of Socio-Ecosystems: A Methodology for the Analysis of Adaptation to Climate Change. International Journal of Agent Technologies and Systems (IJATS) 2 (4), 17-38.

Balbi, S., Giupponi, C., Perez, P., Alberti, M., 2013. A spatial agent-based model for assessing strategies of adaptation to climate and tourism demand changes in an alpine tourism destination. Environmental Modelling & Software 45 (0), 29-51.

Becu, N., Perez, P., Walker, A., Barreteau, O., Le Page, C., 2003. Agent based simulation of a small catchment water management in northern Thailand description of the CATCHSCAPE model. Ecological Modelling 170 (2-3), 319-331.

Benenson, I., Torrens, P.M. 2004. Geosimulation: Automata-based Modeling of Urban Phenomena. Wiley, New York, USA.

Bennett, N.D., Croke, B.F.W., Guariso, G., Guillaume, J.H.A., Hamilton, S.H., Jakeman, A.J., Marsili-Libelli, S., Newham, L.T.H., Norton, J.P., Perrin, C., Pierce, S.A., Robson, B., Seppelt, R., Voinov, A.A., Fath, B.D., Andreassian, V., 2013. Characterising performance of environmental models. Environmental Modelling & Software 40 (0), 1-20.

Bersini, H., 2012. UML for ABM. Journal of Artificial Societies and Social Simulation 15 (1), 9.

Bonabeau, E., 2002. Agent-based modeling: methods and techniques for simulating human systems. Proceedings of the National Academy of Sciences of the United States of America 99 (Suppl. 3), 7280-7287.

Caillault, S., Mialhe, F., Vannier, C., Delmotte, S., Kêdowidé, C., Amblard, F., Etienne, M., Bécu, N., Gautreau, P., Houet, T., 2013. Influence of incentive networks on landscape changes: A simple agent-based simulation approach. Environmental Modelling & Software 45 (0), 64-73.

Castronova, A.M., Goodall, J.L., Elag, M.M., 2013. Models as web services using the Open Geospatial Consortium (OGC) Web Processing Service (WPS) standard. Environmental Modelling & Software 41 (0), 72-83.

Christley, S., Xiang, X., Madey, G., 2004. An ontology for agent-based modeling and simulation, Agent 2004 Conference on Social Dynamics: Interaction, Reflexivity and Emergence. Citeseer.

de Sousa, L., da Silva, A., 2011. Review of Spatial Simulation Tools for Geographic Information Systems, SIMUL 2011, The Third International Conference on Advances in System Simulation, pp. 169-174.

Deadman, P., Robinson, D., Moran, E., Brondizio, E., 2004. Colonist household decisionmaking and land-use change in the Amazon Rainforest: an agent-based simulation. Environment and Planning B-Planning & Design 31 (5), 693-709.

Eden, C., 1988. Cognitive Mapping. European Journal of Operational Research 36 (1), 1-13.

Epstein, J.M., 2008. Why Model? Journal of Artificial Societies and Social Simulation 11 (4), 12.

Filatova, T., Parker, D., van der Veen, A., 2009. Agent-Based Urban Land Markets: Agent's Pricing Behavior, Land Prices and Urban Land Use Change. Jasss-the Journal of Artificial Societies and Social Simulation 12 (1).

Filatova, T., Verburg, P.H., Parker, D.C., Stannard, C.A., 2013. Spatial agent-based models for socio-ecological systems: Challenges and prospects. Environmental Modelling & Software 45 (0), 1-7.

Gaube, V., Remesch, A., 2013. Impact of urban planning on household's residential decisions: An agent-based simulation model for Vienna. Environmental Modelling & Software 45 (0), 92-

103.
Gilberg, R.F., Forouzan, B.A. 2004. Data Structures: A Pseudocode Approach With C. Thomson.
Grau, B.C., Horrocks, I., Motik, B., Parsia, B., Patel-Schneider, P., Sattler, U., 2008. OWL 2: The next step for OWL. Web semantics: science, services and agents on the World Wide Web 6 (4), 309-322.
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. Ecological Modelling 198 (1-2), 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. Ecological Modelling 221 (23), 2760-2768.
Gruber, T.R., 1993. A translation approach to portable ontology specifications. Knowledge Acquisition 5 (2), 199-220.
Haase, D., Lautenbach, S., Seppelt, R., 2010. Modeling and simulating residential mobility in a shrinking city using an agent-based approach. Environmental Modelling & Software 25 (10), 1225-1240.
Hare, M., Deadman, P., 2004. Further towards a taxonomy of agent-based simulation models in environmental management. Mathematics and Computers in Simulation 64 (1), 25-40.
Holland, J.H. 1995. Hidden order: how adaptation builds complexity. Addison-Wesley, Reading, MA.
Horrocks, I., Patel-Schneider, P.F., Van Harmelen, F., 2003. From SHIQ and RDF to OWL: The making of a web ontology language. Web semantics: science, services and agents on the World Wide Web 1 (1), 7-26.
IEEE, 1985. IEEE Standard For Binary Floating-Point Arithmetic. IEEE 754-1985, New York, NY: Institute of Electrical and Electronics Engineers.
Ince, D.C., Hatton, L., Graham-Cumming, J., 2012. The case for open computer programs. Nature 482 (7386), 485-488.
International Organization for Standardization (ISO), 1985. Information processing -- Documentation symbols and conventions for data, program and system flowcharts, program network charts and system resources charts.
Jensen, F.V., 2001. Bayesian networks and decision graphs. Statistics for engineering and information science. Springer 32.
Kelly, R.A., Jakeman, A.J., Barreteau, O., Borsuk, M.E., ElSawah, S., Hamilton, S.H., Henriksen, H.J., Kuikka, S., Maier, H.R., Rizzoli, A.E., van Delden, H., Voinov, A.A., 2013. Selecting among five common modelling approaches for integrated environmental assessment and management. Environmental Modelling & Software 47 (0), 159-181.
Kitchin, R.M., 1994. Cognitive Maps - What Are They and Why Study Them. Journal of Environmental Psychology 14 (1), 1-19.
Knuth, D.E., 1984. Literate Programming. Computer Journal 27 (2), 97-111.
Liu, J.G., Dietz, T., Carpenter, S.R., Folke, C., Alberti, M., Redman, C.L., Schneider, S.H., Ostrom, E., Pell, A.N., Lubchenco, J., Taylor, W.W., Ouyang, Z.Y., Deadman, P., Kratz, T., Provencher, W., 2007. Coupled human and natural systems. Ambio 36 (8), 639-649.
Livet, P., Müller, J.-P., Phan, D., Sanders, L., 2010. Ontology, a Mediator for Agent-Based Modeling in Social Science. Journal of Artificial Societies and Social Simulation 13 (1), 3.
Luke, S., Cioffi-Revilla, C., Panait, L., Sullivan, K., Balan, G., 2005. MASON: A multiagent simulation environment. Simulation 81 (7), 517-527.
Marohn, C., Schreinemachers, P., Quang, D.V., Berger, T., Siripalangkanont, P., Nguyen, T.T., Cadisch, G., 2013. A software coupling approach to assess low-cost soil conservation strategies for highland agriculture in Vietnam. Environmental Modelling & Software 45 (0), 116-128.
Maruyama, M., 1992. A quickly understandable notation system for causal loops for strategic decision makers. Technological Forecasting and Social Change 42 (4), 409-412.
McConnell, W.J., Millington, J.D.A., Reo, N.J., Alberti, M., Asbjornsen, H., Baker, L.A., Brozovic, N.,

Drinkwater, L.E., Drzyzga, S.A., Fragoso, J., Holland, D.S., Jantz, C.A., Kohler, T.A., Maschner, H.D.G., Monticino, M., Podestá, G., Pontius, R.G., Redman, C.L., Sailor, D., Urquhart, G., Liu, J., 2011. Research on Coupled Human and Natural Systems (CHANS): Approach, Challenges, and Strategies. Bulletin of the Ecological Society of America 92 (2), 218-228.

Millington, J.D.A., O'Sullivan, D., Perry, G.L.W., 2012. Model histories: Narrative explanation in generative simulation modelling. Geoforum 43 (6), 1025-1034.

Müller, B., Bohn, F., Dreßler, G., Groeneveld, J., Klassert, C., Martin, R., Schlüter, M., Schulze, J., Weise, H., Schwarz, N., 2013. Describing human decisions in agent-based models – ODD + D, an extension of the ODD protocol. Environmental Modelling & Software 48 (0), 37-48.

North, M., Howe, T., Collier, N., Vos, J., 2007. A declarative model assembly infrastructure for verification and validation, Advancing social simulation: The first world congress. Springer, pp. 129-140.

Object Management Group, 2011. UML Infrastructure Specification, v2.4.1.

Parker, D.C., Brown, D.G., Polhill, J.G., Deadman, P.J., Manson, S.M., 2008. Illustrating a new 'conceptual design pattern' for agent-based models and land use via five case studies: the MR POTATOHEAD framework, In: Paredes, A.L., Iglesias, C.H. (Eds.), Agent-based modelling in natural resource management. Universidad de Valladolid: Valladolid, Spain, pp. 23-51.

Perez, L., Dragicevic, S., 2010. Modeling mountain pine beetle infestation with an agent-based approach at two spatial scales. Environmental Modelling & Software 25 (2), 223-236.

Polhill, J.G., Parker, D., Brown, D. and Grimm, V., 2008. Using the ODD Protocol for Describing Three Agent-Based Social Simulation Models of Land-Use Change. Journal of Artificial Societies and Social Simulation 11 (2), 3.

Polhill, J.G., 2010. ODD Updated. Journal of Artificial Societies and Social Simulation 13 (4), 9.

Polhill, J.G., Edmonds, B., 2007. Open Access for Social Simulation. Jasss-the Journal of Artificial Societies and Social Simulation 10 (3).

Polhill, J.G., Gimona, A., Gotts, N.M., 2013. Nonlinearities in biodiversity incentive schemes: A study using an integrated agent-based and metacommunity model. Environmental Modelling & Software 45 (0), 74-91.

Polhill, J.G., Gotts, N., Sánchez-Maroño, N., Pignotti, E., Fontenla-Romero, Ó., Rodríguez-García, M., Alonso-Betanzos, A., Edwards, P., Craig, T., 2012. An ontology-based design for modelling case studies of everyday proenvironmental behaviour in the workplace. in: Proceedings of the iEMSs Sixth Biennial Meeting: International Congress on Environmental Modelling and Software (iEMSs 2012). International Environmental Modelling and Software Society, Leipzig, Germany, July 2012, pages 10.

Polhill, J.G., Gotts, N.M., 2009. Ontologies for transparent integrated human-natural system modelling. Landscape Ecology 24 (9), 1255-1267.

Polhill, J.G., Izquierdo, L., 2005. Lessons Learned from Converting the Artificial Stock Market to Interval Arithmetic Journal of Artificial Societies and Social Simulation 8 (2).

Quinlan, J.R., 1986. Induction of decision trees. Machine learning 1 (1), 81-106.

Rebaudo, F., Dangles, O., 2013. An agent-based modeling framework for integrated pest management dissemination programs. Environmental Modelling & Software 45 (0), 141-149.

Robinson, D.T., Sun, S., Hutchins, M., Riolo, R.L., Brown, D.G., Parker, D.C., Filatova, T., Currie, W.S., Kiger, S., 2013. Effects of land markets and land management on ecosystem function: A framework for modelling exurban land-change. Environmental Modelling & Software 45 (0), 129-140.

Roy, G.G., 2006. Designing and explaining programs with a literate pseudocode. J. Educ. Resour. Comput. 6 (1), 1.

Schlüter, M., Hinkel, J., Bots, P.W.G., Arlinghaus, R., in press. Application of a SES framework for model-based analysis of the dynamics of social-ecological systems. . Ecology & Society.

Schmolke, A., Thorbek, P., DeAngelis, D.L., Grimm, V., 2010. Ecological modelling supporting environmental decision making: a strategy for the future. Trends in Ecology & Evolution 25 (8), 479-486.

Schreinemachers, P., Berger, T., 2011. An agent-based simulation model of human–environment interactions in agricultural systems. Environmental Modelling & Software 26 (7), 845-859.
Selic, B., 2003. The pragmatics of model-driven development. IEEE Software 20 (5).
Simon, C., Etienne, M., 2010. A companion modelling approach applied to forest management planning. Environmental Modelling & Software 25 (11), 1371-1384.
Smajgl, A., Bohensky, E., 2013. Behaviour and space in agent-based modelling: Poverty patterns in East Kalimantan, Indonesia. Environmental Modelling & Software 45 (0), 8-14.
Smed, J., Hakonen, H. 2006. Algorithms and Networking for Computer Games. John Wiley & Sons.
Sun, Z., Müller, D., 2013. A framework for modeling payments for ecosystem services with agent-based models, Bayesian belief networks and opinion dynamics models. Environmental Modelling & Software 45 (0), 15-28.
Taillandier, P., Vo, D.-A., Amouroux, E., Drogoul, A., 2012. GAMA: a simulation platform that integrates geographical information data, agent-based modeling and multi-scale control. Principles and Practice of Multi-Agent Systems, 242-258.
Tisue, S., Wilensky, U., 2004. Netlogo: A simple environment for modeling complexity, International Conference on Complex Systems, pp. 16-21.
Touza, J., Drechsler, M., Smart, J.C.R., Termansen, M., 2013. Emergence of cooperative behaviours in the management of mobile ecological resources. Environmental Modelling & Software 45 (0), 52-63.
van Oel, P.R., Krol, M.S., Hoekstra, A.Y., Taddei, R.R., 2010. Feedback mechanisms between water availability and water use in a semi-arid river basin: A spatially explicit multi-agent simulation approach. Environmental Modelling & Software 25 (4), 433-443.
Wikipedia, 2013. Wikipedia: WikiProject Computer science/Manual of style - Algorithms, http://en.wikipedia.org/wiki/MOS:PSEUDOCODE#Algorithms (last accessed: 12.03.2013).
Wilensky, U., Rand, W., 2007. Making Models Match: Replicating an Agent-Based Model. Jasss-the Journal of Artificial Societies and Social Simulation 10 (4).
Yu, C., MacEachren, A.M., Peuquet, D.J., Yarnal, B., 2009. Integrating scientific modeling and supporting dynamic hazard management with a GeoAgent-based representation of human–environment interactions: A drought example in Central Pennsylvania, USA. Environmental Modelling & Software 24 (12), 1501-1512.
Zhang, B., Zhang, Y., Bi, J., 2011. An adaptive agent-based modeling approach for analyzing the influence of transaction costs on emissions trading markets. Environmental Modelling & Software 26 (4), 482-491.

Figure captions:

Figure 1: Classification for structuring the prevalent types of model descriptions