CoSMoS simulation experiment reproducibility and the ODD protocol

Susan Stepney[1]

[1]Department of Computer Science, University of York, UK

Abstract. CoSMoS is a defined approach for designing, building, and arguing fit for purpose, a scientifically rigorous simulation of a complex real world system. The ODD (Overview, Design concepts, Details) protocol is a standardised way of describing simulation models, defined to support reproducibility. This paper demonstrates that use of the CoSMoS approach to build a simulation supports the presentation of the simulation results using the ODD protocol. It does so by presenting a mapping from the various ODD components, to where the required information for those components is located in a CoSMoS project's documentation. All the information is present, but is distributed through the project documentation. Moreover, a project developed using the CoSMoS approach documents additional information, which is necessary not merely for reproducibility, but for understanding of and confidence in the simulations.

1 Introduction

Computer-based simulation is a key tool in many fields of scientific research. In silico experiments can be used to explore and understand complex processes, to guide and complement in vitro and in vivo experiments, to suggest new hypotheses to investigate, and to predict results where experiments are infeasible. Simulation is an attractive, accessible tool: producing new simulations of simple systems is relatively easy. But it is also a dangerous tool: simulations are often complex, buggy, and difficult to relate to the real-world system.

A simulation needs to be both scientifically useful to the researcher, and scientifically credible to third parties; it needs to have the properties of a well-designed *scientific instrument* [4]. The CoSMoS (Complex Systems Modelling and Simulation infrastructure) project has established an approach to simulation of complex systems that supports principled development of simulations as scientific instruments. The CoSMoS approach [2, 23] is generic: it does not mandate a particular modelling technique, or particular implementation language. What it does mandate is the careful and structured use of models and arguments, to ensure that the simulation is both well-engineered, and seen to be well-engineered. In order to help developers through this careful and structured approach, we have developed a pattern language [22, 23] to help guide development, promote good simulation engineering practice, and warn of potential pitfalls.

Since CoSMoS is generic, it can work in harmony with various other approaches that are used for building scientific simulations. For example, there are well-established third party simulation implementation libraries and frameworks. NetLogo [26] is a multi-agent programmable modelling environment used for prototyping multi-agent simulations. It provides a programming language and user-interface widgets to support rapid prototyping of relatively simple agent-based simulations. It has a large user-base and a large library of example simulations. MASON [12], from George Mason University, is a discrete-event multi-agent simulation library, which can be used as the foundation for Java simulations. FLAME (Flexible Large-scale Agent Modelling Environment) [6], from the University of Sheffield, is an agent-based modelling system. From an extended finite state machine model, FLAME generates a complete agent-based application, which can be targetted to computing systems ranging from laptops to super computers. Libraries and frameworks such as these can be exploited as part of a simulation project being developed with the CoSMoS approach. They can be used as the implementation basis of the CoSMoS Simulation Platform (see later) or of a Prototype. Their integration into the overall project can be argued fit-for-purpose using the CoSMoS Argumentation approach [15–17]. Thus a CoSMoS user can benefit from the existing significant investment in such implementation platforms.

In addition to specific implementation technologies, researchers have also developed approaches to making the resulting simulations more scientifically acceptable. One essential requirement of a scientific result is that it be *reproducible* [18, §22]. This requires sufficient description of the experimental details that third parties can replicate the setup, and reproduce the result. If simulation is to be used as a scientific instrument, credible to the scientific community, then it is necessary for individual simulation experiments to be described in sufficient detail for their reproducibility. Grimm and co-workers have devised the ODD (Overview, Design concepts, Details) protocol [7, 8] as a standard way of describing simulations using Agent Based Models (ABMs), with the explicit aim of making them reproducible: "ODD is expected to lead to more complete model descriptions, making ABMs easier to replicate and hence less easily dismissed as unscientific" [8].

The Open ABM Consortium [13] provides support for the ODD protocol. It has a discussion forum "to allow the ABM community to collaboratively develop this protocol into a widely useable communication standard for describing ABMs in the social sciences" [11]. As part of its support, it defines CoMSES (Computational Modeling for SocioEcological Science) Modeling Standards, which include (among other things) the requirement to be "fully documented using the ODD standard for model documentation, or an equivalent documentation protocol" [14].

In the same way that existing libraries and frameworks can be exploited by the CoSMoS user for implementation, the existing ODD protocol can be followed for presenting CoSMoS simulation experiments and results. This provides the CoSMoS user with a well-established and recognised means to ensure third party

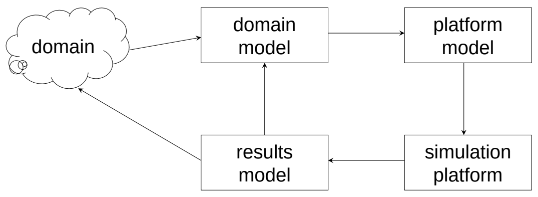

Fig. 1. Relationship between simulation components; arrows represent flows of information. These are all framed by the Research Context.

reproducibility of their scientific simulation results. Contrariwise, it provides the ODD-compliant simulation developer with the broader CoSMoS approach within which to develop a scientifically credible simulator.

In order to support such reuse of the ODD protocol within CoSMoS, this paper presents a mapping from the various ODD components, to where the required information is located in CoSMoS components.

2 CoSMoS in a nutshell

The CoSMoS approach is documented through a pattern language [22,23]. These patterns are based on a collection of models (figure 1) [3] and other components. The patterns referenced in this paper are:

Research Context: the statement of the overall scientific context, goals and scope of the simulation-based research being conducted, including the Simulation Purpose, resources, constraints, assumptions, and success criteria.

Domain: the part of the real world that is the system of study, that the simulation project is "about".

Domain Model: a model encapsulating the scientific understanding of appropriate aspects of the Domain. It provides the agreed scientific basis and assumptions for the development of a Simulation Platform; simulation implementation details are not considered in this model.

Platform Model: a model providing the high level specification of the Simulation Platform, comprising design and implementation details, incorporating relevant Domain Model scientific concepts, Research Context experimental requirements, and implementation constraints and assumptions.

Simulation Platform: the encoding of the Platform Model into a Calibrated software and hardware platform with which various Simulation Experiments can be performed.

Results Model: a model that encapsulates the understanding of outputs and results from Simulation Experiments, in Domain terms, enabling comparison with results from domain experiments.

Simulation Purpose: a component of the Research Context. It documents the scientific purpose for which the Simulation Platform is being built and used.

The *purpose* of a simulation exercise is the single most important concept. Without an agreed purpose, it is impossible to scope the research context, or to arrive at a consensus over fitness for purpose. The purpose of the simulation constrains the appropriate levels of abstraction for modelling, the appropriate implementation languages and platform, and the appropriate analysis and interpretation of results. A simulator that is designed for one purpose may or may not be modifiable for another purpose.

Modelling Approach: a component of the Domain Model and Platform Model. The choice of an appropriate modelling approach and notation that can capture the relevant aspects of the Domain and Simulation Purpose in a form that can be implemented in the Simulation Platform.

Argumentation: the demonstration that the simulation is fit for purpose (as defined in the Simulation Purpose). The fitness-for-purpose argument exposes the rationale for scientific and engineering credibility of the simulator, the assumptions made, the limitations in the purpose and scope of the simulator, and in the potential use of its results. While CoSMoS's rigorous model-based approach ensures that the simulator *is* fit for purpose as a scientific instrument, the accompanying argumentation ensures that the simulator *is seen to be* fit for purpose.

Simulation Experiment: a specific experiment executed on the Simulation Platform. This should be designed and documented by analogy to a domain experiment, including the hypothesis to be tested, initial conditions and parameter values, experimental process, and results analysis.

Data Dictionary: a component of the Domain Model, Platform Model, Results Model, and Simulation Experiment. It provides a common definition of the modelling data (parameter values such as sizes, scales, rates) used to build the Simulation Platform, calibration data, and the experimental data (initial values and results) both from Domain experiments and the corresponding Simulation Experiments.

Calibration: tuning the Simulation Platform parameter values so that simulation results match the experimental calibration data provided in the Data Dictionary [1, 20].

There are many more patterns that make up the CoSMoS approach. The ones summarised here are those that are relevant to mapping the ODD protocol.

3 ODD and CoSMoS

The ODD (Overview, Design concepts, Details) protocol was introduced in [7] and refined in [8] as a standard way of describing simulation models (particularly in the ecological domain), specifically to aid reproducibility of implementation of such models. By 2010 the original formulation had been used in more that 50 publications [8]; experience gained from this use led to its update.

ODD is more narrowly focussed than CoSMoS: it is concerned with reproducibility of simulation models from their given descriptions (although a noted side-effect of its use is "a more rigorous formulation of models" [8]). Hence the ODD protocol information is only part of the entire CoSMoS model: it is mostly contained in the Platform Model, and some of the Research Context. ODD explicitly does not cover any Results Model features of experimentation and sensitivity analysis (it considers these to be the "methods" part of a description; the ODD protocol covers only the "materials" part of a standard scientific article [7]). Also, it is more concerned with the implemented model, so it has only a little that correspond to Domain Model elements.

In addition to the explicit material noted in the summary below, the ODD protocol also recommends making the source code available (that is, the Simulation Platform and Simulation Experiment), and using code comments to highlight the various parts of the protocol information [7].

The CoSMoS approach has places for all the information needed to give an ODD protocol description. This section describes the various components of the updated ODD protocol (as defined in [8, §3]), and explains where the corresponding material can be found in a CoSMoS-based simulation.

ODD quotations in this section are taken from [8].

3.1 ODD: purpose

ODD component: "a concise summary of the overall objectives(s) for which the model was developed"

CoSMoS location: this maps naturally onto the Simulation Purpose.

Simulation purpose is closely tied to criticality (for example, will the results be applied directly, or be used to generate hypotheses that can be tested by other means) and impact (for example, will the results affect small-scale research, or large scale policy decisions) [15]. If the purpose is to provide critical evidence for high-impact research, then the approach to simulation development should access state-of-the-art software engineering methods, argumentation, and documentation approaches; the simulator and its fitness-for-purpose must be capable of international expert scrutiny. However, if the purpose is to provide a test-bed for hunches and a generator of hypotheses, all of which will then be subject to conventional laboratory analysis and confirmation before publication, then the development and argumentation need to be just good enough: internal consensus and internal documentation are sufficient.

3.2 ODD: entities, state variables and scales

This part of the ODD protocol specifies the structure of the modelled world in terms of its components. Like CoSMoS, ODD claims not to be wedded to any one implementation approach: "Although the protocol was designed for [Agent Based Model]s, it can help with documenting any large, complex model" [8]. Nevertheless, the terminology in this part of the ODD protocol is ABM-specific. The Modelling Approach is assumed to be Agent Based from now on; UML is a notation that may be used to capture aspects of ABMs [5].

Entities

ODD component: "An entity is a distinct or separate object or actor that behaves as a unit and may interact with other entities or be affected by external environmental factors."

[8] distinguishes the following types of entities: (i) agents/individuals; (ii) spatial units (grid cells); (iii) environment; (iv) collectives (groups of agents that can have their own group-level identity and behaviours)

CoSMoS location: the implemented entities are captured as part of the Platform Model; if UML is the notation used, then the entities could be partially captured as the classes in a class diagram.

Most Platform Model entities are derived from, but may differ from, those in the Domain Model. For example, simplifications may be made; platform entities may be surrogates for more complex domain entities. In particular, the Domain Model may include emergent entities that are not explicitly included in the Platform Model, since determining their (hypothesised) emergence may be part of the Simulation Purpose. Non-implemented emergent entities are distinct from ODD "collectives", which are explicitly modelled and implemented hierarchical entities. It is important to distinguish these cases in order to ensure that the desired emergent answer is not hard-coded into the simulation. This is one reason why CoSMoS makes a careful distinction between the Domain Model and Platform Model [3].

There may be additional entities in the Platform Model, such as implementation-specific entities (such as instrumentation and interface entities needed to support Simulation Experiments, or proxy entities needed to implement a large-scale distributed simulator). Again, this is a reason to separate the Domain Model and Platform Model: the scientific Domain Model is not polluted with implementation details, and may be an appropriate basis for diverse implementations.

State variables

ODD component: "A state variable or attribute is a variable that distinguishes an entity from other entities of the same type or category, or traces how the entity changes over time. ... [they] can contain both numerical variables and references to behavioural strategies. ... If state variables have units, they should be provided."

CoSMoS location: the implemented state variables are captured as part of the Platform Model. If UML is the notation used, these would be captured, for example, as the instance variables (for "numerical variables") and methods (for "behavioural strategies") in a class diagram. The CoSMoS Data Dictionary component pattern contains some of the detail, including the units.

Scales

ODD component: "spatial and temporal scales and extents (the amount of space and time represented in a simulation), . . . what the model's units represent in reality."

CoSMoS location: this information is documented in the Data Dictionary, both the Domain Model part for physical scales, and the Platform Model part for any surrogates and abstractions of these scales. Tuning these and other experimental parameter values, particularly where experimental values refer to Domain entities and the corresponding Platform Model entities are surrogates, is part of Calibration. The validation of the mapping from real world units to simulation units, and of the appropriateness of the spatial and temporal discretisation (including spatial grid size and time-step size) is part of the Argumentation process.

3.3 ODD: process overview and scheduling

ODD component: this covers the dynamical behaviour of the model: what the agents do, in what order (the order in which state variables are updated, synchronous or asynchronous, concurrency); the behaviour of any controller object; how time is modelled (discrete, continuous, or hybrid).

"one should use pseudo-code to describe the schedule in every detail, so that the model can be re-implemented from this code."

CoSMoS location: the Platform Model contains this information, at various levels of detail. If UML is the notation used, the highest level model might be expressed as activity diagrams and state diagrams. Lower level models, developed on the way to implementation, might be expressed in pseudo-code.

3.4 ODD: design concepts

The final part of the ODD protocol is a list of specific concepts that need to be defined for reproducibility. See also [9, ch.5], [19] for further discussion. These concepts are particularly important for ecological models where the agents have complex behaviours, where the agents may change their behaviours, and where emergent properties are prominent. Not all these concepts are important in every CoSMoS simulation, but when they are, there is a place for them to be captured in the approach, listed below.

ODD: basic principles

ODD component: the general concepts, theories, hypotheses and modelling approaches underlying the model's design.

CoSMoS location: these are variously captured in the Research Context and the Domain (general concepts), the Domain Model (theories and hypotheses), and the Modelling Approach. The various components may be brought together during Argumentation.

ODD: emergence

ODD component: the system level phenomena that emerge from the individual behaviours, rather than being built in to the model

CoSMoS location: emergent properties are captured in the Domain Model, and removed from Platform Model. In general, given a hypothesis under consideration, components in the Domain Model that are *outcomes* of hypothesised mechanisms, whether emergent or not, should not appear in the Platform Model, to ensure that the answer is not be explicitly coded into the Simulation Platform [3]. The Results Model is used to capture emergent properties of Simulation Experiments; the simulated properties captured in the results model can be compared to the real-world properties in the Domain Model.

ODD: adaptation, objectives, learning, prediction

ODD component: the rules the entities have for making decisions and changing behaviour in response to changes in themselves or the environment; measures of an entity's adaptive success (such as fitness, utility), and the criteria used to measure this success; how entities change their adaptive traits; how an entity predicts future consequences in order to make decisions.

CoSMoS location: the real world versions of these can all be captured in the Domain Model as a specific kind of entity behaviour or property. The appropriate abstractions and surrogates are then all captured in the Platform Model. If UML is the notation used, then a stereotype can be used to highlight the specific kinds of behaviours and properties of interest.

ODD: sensing, interaction

ODD component: what state variable values (of itself, and of the environment) an entity has access to, that it can exploit to guide or influence its behaviour; any direct and indirect interactions between agents; representation of communications

CoSMoS location: setting the scope of entity interactions, in terms of sensing capabilities and levels of detail, is part of the Domain Model. The information access across entities required for simulation is captured in the Platform Model. If UML is the notation used, then the relationships between entities (including the environment entity) a class diagram can document the information they can sense about each other.

ODD: stochasticity

ODD component: how and where randomness is used to model real world variability that is unimportant to model in detail

CoSMoS location: the Domain Model captures the real world variability at some level of abstraction; the Platform Model implements this as randomness; the Argumentation of the appropriateness of this particular form of implementation demonstrates that the approximation is fit for purpose.

ODD: collectives

ODD component: collections of agents that behave as entities in their own right; which are emergent, and which are explicitly modelled

CoSMoS location: the Platform Model captures explicitly-modelled collectives; emergent collective entities are in the Domain Model but removed from the Platform Model (see the discussion under the emergence heading earlier in this section).

ODD: observation

ODD component: a definition of the data samples collected from the ABM for testing, understanding, and analysing it.

CoSMoS location: the Platform Model includes specification of the instrumentation used to produce the output data; the Data Dictionary (Results Model component) includes the specification of the output data and its analysis; the Simulation Experiment includes experiment-specific details.

3.5 ODD: initialisation

ODD component: the initial state of simulation, the number and state of agents and the environment, at time $t = 0$.

CoSMoS location: state variable values, and other parameter values, for specific Simulation Experiments are held in the Data Dictionary.

3.6 ODD: input data

ODD component: the data that drives environmental variables (such as rainfall or harvesting regimes) [7], imported from external files or models.

CoSMoS location: raw data for specific Simulation Experiments is held in the Data Dictionary. If the data is produced on the fly by an external model, that model and its interfaces will be captured, at some level of abstraction, in the Platform Model. Initialisation data for that model is also held in the Data Dictionary.

3.7 ODD: submodels

ODD component: the detailed sub-models, mathematical equations, rules, and parameters that define the processes (§3.3).

CoSMoS location: the Platform Model contains this information, at various levels of detail. For components such as mathematical equations, the Platform Model may contain a reference to where the equation appears in the Domain Model, plus the extra definitions necessary to implement the equation under the prevailing model assumptions such as the implementation of time and space. Parameters are stored in the Data Dictionary.

4 Summary and Conclusions

As demonstrated, the documentation resulting from a simulation project developed under the CoSMoS approach contains places for all the information needed to document an ABM using the ODD protocol. Hence CoSMoS can be used to develop ODD-compliant simulations. However, the material is scattered through several CoSMoS artefacts: the Platform Model and its Data Dictionary; the Research Context and Simulation Purpose; the Domain Model; the Argumentation; and more. If the ODD protocol is to be used to present the results of a CoSMoS-developed Simulation Experiment, then it would be sensible to devise some project standards that specifically tag the ODD-relevant information in each of the CoSMoS artefacts.

A CoSMoS project's artefacts contain much more than the information required by the ODD protocol. Specifically, much of the ODD information is found in the CoSMoS Platform Model, yet much of the information in the Platform Model is itself derived from the Domain Model. Additionally, there is information in the Domain Model explicitly not carried over to the Platform Model: the emergent properties and other hypothesised outputs of the Simulation Experiments. This extra, crucial, information has a formal home in the CoSMoS approach.

Furthermore, a key part of the CoSMoS approach is Argumentation: the explicit demonstration that the simulation is fit for purpose [15–17]. This exposes a difference in the goals of ODD and CoSMoS: use of the ODD protocol makes simulation experiments more *reproducible*; use of the CoSMoS approach additionally helps to ensure that the simulated world has a clear link to the real world (the experiments *make domain sense*), and that the simulator as a scientific instrument is fit for purpose (the experimental results are *credible*).

The developers of ODD are not trying to capture everything in their protocol: in particular, ODD covers only the "materials" part of a standard scientific article, and not the "methods" part [7]. The ODD researchers are developing TRACE (Transparent and Comprehensive Ecological Documentation) [10, 21] for more fully documenting models and their analyses. The OpenABM CoMSES Modeling Standards requires that a model "Correctly simulates the processes it claims to simulate" [14]. Future work for the CoSMoS approach is to demonstrate how it is compliant with the requirements of TRACE and CoMSES.

Acknowledgements

This work is part of the Complex Systems Modelling and Simulation (CoSMoS) project, partially funded by EPSRC grants EP/E053505/1 and EP/E049419/1. I would like to thank my CoSMoS project colleagues, and pattern book [23] co-authors, for providing much of the raw material on which this paper is based. Particular thanks go to Paul Andrews, Jim Bown and Fiona Polack, for comments on earlier versions of this paper. Thanks also to Teodor Ghetiu for bringing the ODD protocol to the CoSMoS team's attention.

CoSMoS project documentation is available from the CoSMoS project website www.cosmos-research.org.

References

1. Kieran Alden, Mark Read, Jon Timmis, Paul S. Andrews, Henrique Veiga-Fernandes, and Mark Coles. *Spartan*: A comprehensive tool for understanding uncertainty in simulations of biological systems. *PLoS Comput Biol*, 9(2):e1002916, 2013.
2. Paul S. Andrews, Fiona A. C. Polack, Adam T. Sampson, Susan Stepney, and Jon Timmis. The CoSMoS process, version 0.1: A process for the modelling and simulation of complex systems. Technical Report YCS-2010-453, Department of Computer Science, University of York, March 2010.
3. Paul S. Andrews, Susan Stepney, Tim Hoverd, Fiona A. C. Polack, Adam T. Sampson, and Jon Timmis. CoSMoS process, models, and metamodels. In Stepney et al. [25], pages 1–13.
4. Paul S. Andrews, Susan Stepney, and Jon Timmis. Simulation as a scientific instrument. In Stepney et al. [24], pages 1–10.
5. Bernhard Bauer and James Odell. UML 2.0 and agents: How to build agent-based systems with the new UML. *Journal of Engineering Applications of Artificial Intelligence*, 18:141–157, 2005.
6. FLAME website: www.flame.ac.uk.
7. Volker Grimm, Uta Berger, Finn Bastiansen, Sigrunn Eliassen, Vincent Ginot, Jarl Giske, John Goss-Custard, Tamara Grand, Simone K. Heinz, Geir Huse, Andreas Huth, Jane U. Jepsen, Christian Jørgensen, Wolf M. Mooij, Birgit Müller, Guy Pe'er, Cyril Piou, Steven F. Railsback, Andrew M. Robbins, Martha M. Robbins, Eva Rossmanith, Nadja Rüger, Espen Strand, Sami Souissi, Richard A. Stillman, Rune Vabø, Ute Visser, and Donald L. DeAngelis. A standard protocol for describing individual-based and agent-based models. *Ecological Modelling*, 198(1-2):115–126, 2006.
8. Volker Grimm, Uta Berger, Donald L. DeAngelis, J. Gary Polhill, Jarl Giske, and Steven F. Railsback. The ODD protocol: A review and first update. *Ecological Modelling*, 221(23):2760–2768, 2010.
9. Volker Grimm and Steven F. Railsback. *Individual-based Modeling and Ecology*. Princeton University Press, 2005.
10. Volker Grimm and Amelie Schmolke. How to read and write TRACE documentations, 1st draft. Technical report, Helmholtz Centre for Environmental Research, Leipzig, Germany, 2011.

11. Marco A. Janssen, Lilian Na'ia Alessa, Michael Barton, Sean Bergin, and Allen Lee. Towards a community framework for agent-based modelling. *Journal of Artificial Societies and Social Simulation*, 11(2):6, 2008.
12. Sean Luke, Claudio Cioffi-Revilla, Liviu Panait, Keith Sullivan, and Gabriel Balan. MASON: A multi-agent simulation environment. *Simulation: Transactions of the society for Modeling and Simulation International*, 82(7):517–527, 2005. Website: cs.gmu.edu/~eclab/projects/mason.
13. OpenABM website: www.openabm.org.
14. OpenABM model certification: www.openabm.org/faq/what-model-certification-and-how-does-it-work.
15. Fiona A. C. Polack. Arguing validation of simulations in science. In Susan Stepney, Peter H. Welch, Paul S. Andrews, and Adam T. Sampson, editors, *2010 CoSMoS workshop*, pages 51–74. Luniver Press, 2010.
16. Fiona A. C. Polack, Paul S. Andrews, Teodor Ghetiu, Mark Read, Susan Stepney, Jon Timmis, and Adam T. Sampson. Reflections on the simulation of complex systems for science. In *ICECCS 2010*, pages 276–285. IEEE Press, 2010.
17. Fiona A. C. Polack, Alastair Droop, Philip Garnett, Teodor Ghetiu, and Susan Stepney. Simulation validation: exploring the suitability of a simulation of cell division and differentiation in the prostate. In Stepney et al. [25], pages 113–133.
18. Karl Popper. *The Logic of Scientific Discovery.* Hutchinson, 1959.
19. Steven F. Railsback. Concepts from complex adaptive systems as a framework for individual-based modelling. *Ecological Modelling*, 139(1):47–62, 2001.
20. Mark Read, Paul S. Andrews, Jon Timmis, and Vipin Kumar. Techniques for grounding Agent-Based Simulations in the real domain: a case study in Experimental Autoimmune Encephalomyelitis. *Mathematical and Computer Modelling of Dynamical Systems (MCMDS)*, 18(1):67–86, 2012.
21. Amelie Schmolke, Pernille Thorbek, Donald L. DeAngelis, and Volker Grimm. Ecological models supporting environmental decision making: a strategy for the future. *Trends in Ecology & Evolution*, 25(8):479–486, 2010.
22. Susan Stepney. A pattern language for scientific simulations. In Stepney et al. [24], pages 77–103.
23. Susan Stepney, Kieran Alden, Paul S. Andrews, James L. Bown, Alastair Droop, Teodor Ghetiu, Tim Hoverd, Fiona A. C. Polack, Mark Read, Carl G. Ritson, Adam T. Sampson, Jon Timmis, Peter H. Welch, and Alan F. T. Winfield. *Engineering Simulations as Scientific Instruments.* Springer, 2013. in preparation.
24. Susan Stepney, Paul S. Andrews, and Mark Read, editors. *Proceedings of the 2012 Workshop on Complex Systems Modelling and Simulation, Orleans, France, September 2012.* Luniver Press, 2012.
25. Susan Stepney, Peter Welch, Paul S. Andrews, and Carl G. Ritson, editors. *Proceedings of the 2011 Workshop on Complex Systems Modelling and Simulation, Paris, France, August 2011.* Luniver Press, 2011.
26. Uri Wilensky. NetLogo. Website: ccl.northwestern.edu/netlogo.