Proceedings of the 2012 Winter Simulation Conference
C. Laroque, J. Himmelspach, R. Pasupathy, O. Rose, and A. M. Uhrmacher, eds.

SEVEN PITFALLS IN MODELING AND SIMULATION RESEARCH

Adelinde M. Uhrmacher

Albert Einstein Str. 22
University of Rostock
18059 Rostock, GERMANY

ABSTRACT

Modeling and simulation is applied in many disciplines. While its multidisciplinarity is part of its fascination, its ubiquity also holds some dangers. Being not aware about these dangers might imply that resources are wasted, (PhD-) projects fail, and the overall progress of modeling and simulation is needlessly slowed down. Seven of these pitfalls are identified and tentative recommendations are made on how these pitfalls can be avoided.

1 INTRODUCTION

Since many years efforts are dedicated towards establishing modeling and simulation as an own discipline and along with it a methodology comprising specific phases, tasks, and methods. These efforts are hampered by a salient feature of modeling and simulation, i.e., its interdisciplinarity. Modeling and simulation is applied in many different disciplines with their specific scientific cultures. The analyzed systems require tailored solutions as do the questions and the manner how those are asked and answered. In addition, solutions are based on methodological developments in mathematics, computer science, and the application field. Thus, any research in modeling and simulation is interdisciplinary in nature. Hardly anybody will pursue a PhD, similarly weighting simulation study, mathematics, and computer science. Neither is this advisable, as one might end up knowing a bit but just too little about everything (see also the discussion on “From short and fat, to tall and thin, to T-shaped people” about selecting a research PhD topic (Fujimoto 2011)). However, independently what is called one’s scientific home, whether one’s research is more application-oriented or more method-oriented in nature, some pitfalls are to be circumvented by all modeling and simulation researchers alike.

This is not the first paper on pitfalls in modeling and simulation. E.g., Law (2003), Banks and Chwif (2011), Barth et al. (2012) describe pitfalls of simulation studies, when collecting and interpreting data, building a model, verifying and validating a model, executing a model, interpreting the results, and making decisions based on them. The pitfalls I will be referring to are of a more general nature but not less dangerous for the success of a PhD project in modeling and simulation.

2 PITFALL 1: DON’T KNOW WHETHER IT IS MODELING OR SIMULATION

It is rather easy to get stuck in the terminological jungle of modeling and simulation. Plenty of definitions for modeling and simulation exist (Oren 2011). I would like to cite two definitions that I found helpful to capture the notion of modeling and simulation and in keeping things apart. One definition has been attributed to Marvin Minsky, “A model of a system S and an experiment E is anything to which E can be applied to answer questions about S” (Cellier 1991, p.5). It emphasizes that a model is constructed and validated with certain questions and experiments in mind, which also explains why even validated models might be utterly useless for answering certain questions. Questions and experiments, which a model has been generated for, also determine largely in which context it can be reused (Overstreet et al. 2002). “Simulation is an experiment performed on a model” - this definition is attributed to Granino Korn and

978-1-4673-4782-2/12/$31.00 ©2012 IEEE

John Wait (Cellier 1991, p.6). It is a quite commonly agreed upon interpretation of simulation and clarifies the relation between model and simulation.

The importance of a clear distinction between model and simulation I will illustrate shortly. In the last years several papers have announced the need to combine different approaches, e.g., discrete event and agent-based simulation, or discrete event and systems dynamics simulation (Johansson et al. 2010; Jain et al. 2011). Discrete event simulation is about experiments with models that describe discrete event systems. So discrete event refers to what we want to describe, i.e., the semantics of a model. The syntax, i.e. how we describe such a discrete event system, might vary significantly (see Figure 1). E.g., we can describe discrete event systems as stochastic Petri Nets, DEVS (Page 1994, p.32), or one of the proprietary formats that discrete event modeling and simulation tools offer. One of these modeling approaches might even emphasize an agent-based perception on the system of interest (Himmelspach et al. 2010; Uhrmacher and Weyns 2009).

So it is difficult to guess that what is really meant by combining discrete event and agent-based simulation, is a combination of a particular discrete event modeling approach (focusing on processes, flow, and control) and an agent-based modeling approach (focusing on individuals and their interactions) (Siebers et al. 2010). Let us take a look at the announced combination of systems dynamics and discrete event simulation. Systems dynamics refers to a particular approach in modeling, which however is typically associated with a continuous semantics. Note that older versions of some systems dynamics tools use a forward Euler integrator with fixed step-size to execute models. This might lead to drastically wrong (referring to a continuous systems semantics) results. One could assume that combining systems dynamics with discrete event simulation is about introducing events into the continuous world of systems dynamics. However this is not what is meant, again the work refers typically to a particular discrete event modeling approach and the implied specific view on the system of interest which is combined with a systems dynamics approach (see e.g., Brailsford et al. (2010)).

Figure 1 shows model fragments in different discrete event modeling approaches that describe a gene regulatory process. A repressor is activated by Tryptophan (Trp) binding to it. In this activated state the repressor can dock to the operator region of the Trp operon. The Trp can also be released and the repressor becomes inactive again. The repressor binding to the operator region of the Trp operon prevents that RNA polymerase can bind to the promoter region of Trp operon and thus, hampers the transcription of genes that are responsible for synthesizing Trp. After a certain time the repressor might undock from the operon again enabling the transcription of genes. Neither transcription of genes nor the synthesis are part of the shown model fragments. The sojourn times are exponentially distributed.

In DEVS only the repressor model is shown with its different phases, i.e. inactive, active, and docking. DEVS supports a reactive systems metaphor, systems react to incoming events, e.g., the announcement of docking, and change their state after a certain time, and communicate via events with their environment. The stochastic pi model describes the system as a set of concurrently active processes that are communicating synchronously over channels. The rule-based model, describes the dynamics in terms of reactions.

All of these are discrete event models. The compactness of modeling languages might vary significantly as does the view they offer, the expressiveness, and the "intuitiveness" for different groups of users, see also Faeder (2011).

Distinguishing between modeling and simulation clarifies the number and nature of problems that need to be solved (see, e.g., also Davis and Tolk (2007)). If a discrete event formalism and an agent-based approach with a discrete event semantics are combined, the work to be done refers to the syntax: how can the user describe such a model in a concise and compact manner? If systems dynamics and a discrete event modeling formalism are combined, work has to be done referring to the syntax and, in addition, referring to semantics (e.g., do I only allow linear continuous dynamics in coupling continuous and discrete dynamics?), and referring to the execution algorithms (e.g., do I use bi-section methods to identify the threshold crossing where the continuous change shall trigger an event?). Looking at the problem in this

```
Repressor :=
 X := { "trp", myName }, Y := { "trp", myName }

 S := { (phase, releaseT, undockingT) | phase ∈ { inactive, active, docking }, releaseT, undockingT ∈ R }

 ta (phase, ...) :=
       if phase == active then exponentialRandom(releaseT)
       if phase == docking then exponentialRandom(undockingT)
       else infinity

 δ_int (phase, ...) :=
       if phase == active then (inactive, ...) else if phase == docking then (active, ...)

 δ_ext ((phase, ...), elapsedT, inp) :=
       if phase == inactive ∧ inp == "trp" then (active, ...)
       else if phase == active ∧ inp == myName then (docking, ...)

 λ (phase, ...) :=  if phase == active then "trp" else if phase == docking then myName
```

```
Repressor ::= (tryptophanRepressor (release)) .  ActiveRepressor (release) .

ActiveRepressor (release) ::=  ((release(),releaseT). Repressor  +
   ((ν docking) (repressorOperon(docking),dockingT).(docking(),undockingT) .
   ActiveRepressor (release).


Tryptophan ::=
   ((ν release) (trptophanRepressor(release),reactionT) . (release()) . Tryptophan

Operon ::= (repressorOperon(docking)) . (docking()) . Operon.
```

```
Repressor(inactive) + Tryptophan → Repressor(active) @ reactionTime
Repressor(active) → Repressor(inactive) + Tryptophan @ releaseTime
Repressor(active) + Operon(free) → Operon(dockedRepressor) @ dockingTime
Operon(dockedRepressor) → Operon(free) + Repressor(active) @ undockingTime
```

Figure 1: A fragment of a Tryptophan regulatory model in three different discrete event modeling approaches, repressor in DEVS, operon, tryptophan, and repressor in stochastic PI, and in a rule-based language.

manner will also facilitate to draw on earlier solutions. The terminological conciseness is a pre-requisite for another ingredient of successful research in modeling and simulation, i.e., separation of concerns.

3 PITFALL 2: NO SEPARATION OF CONCERNS

Separation of concerns is an important software design principle and is assumed to go back to Edsger Dijkstra (Dijkstra 1982): “Let me try to explain to you, what to my taste is characteristic for all intelligent thinking. It is, that one is willing to study in depth an aspect of one’s subject matter in isolation for the sake of its own consistency, all the time knowing that one is occupying oneself only with one of the aspects. We know that a program must be correct and we can study it from that viewpoint only; we also know that it should be efficient and we can study its efficiency on another day, so to speak. In another mood we may ask ourselves whether, and if so: why, the program is desirable. But nothing is gained –on the contrary!– by tackling these various aspects simultaneously.”. Since the 70s, this design principle has also been promoted in the modeling and simulation community by Bernard Zeigler, by clearly distinguishing between modeling formalism, its intended semantics (specified as an abstract simulator), and the experimental settings, treating the later as a first class object, i.e., the experimental frame (Zeigler

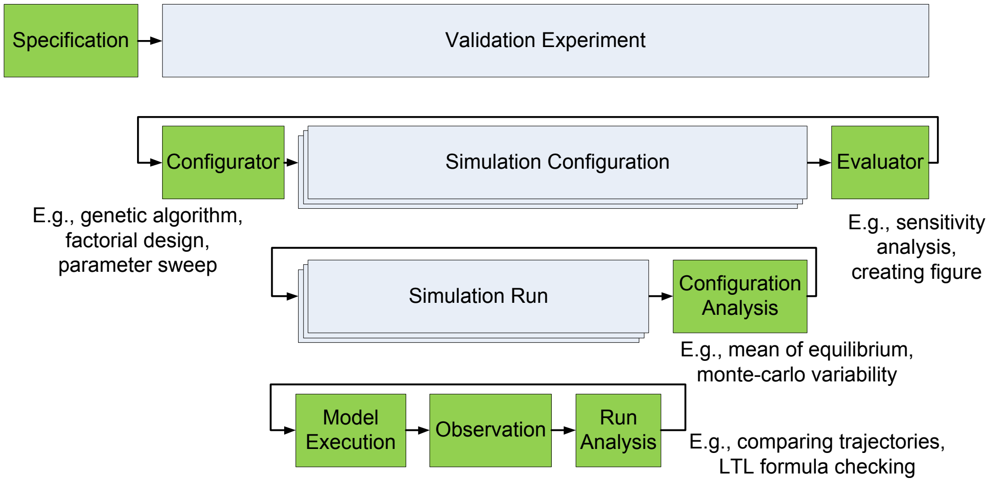

Specification
Validation Experiment
Configurator
Simulation Configuration
Evaluator
E.g., genetic algorithm, factorial design, parameter sweep
E.g., sensitivity analysis, creating figure
Simulation Run
Configuration Analysis
E.g., mean of equilibrium, monte-carlo variability
Model Execution
Observation
Run Analysis
E.g., comparing trajectories, LTL formula checking

Figure 2: Validating a model comprises several steps and requires the application of different methods.

1984) (see Pitfall 1). Therefore, a model specified in a modeling formalism, can be executed by different algorithms, e.g., a reaction network can be executed by numerical integration or by stochastic simulation algorithms that might be approximative or exact (Jeschke and Ewald 2008). Also one and the same execution algorithm can be used in different settings, e.g, in performance studies or application oriented simulation studies. One and the same experimental setting can be used to test different alternative models and, thus, to explore the implications of a family of models. If simulation is an experiment performed with a model, is a single execution of a model already a simulation, are multiple replications a simulation, or an optimization? Simulation can refer to all of those, depending on the question we would like to answer. However, only in very rare occasions, the typical toy duck approach to simulation will suffice: “Wind it up and let it run” (Robinson 2002; Wang and Lehmann 2010) (see also Pitfall 6). Almost always more intricate experiments will be required. We could apply our separation of concerns to the analysis of configurations, replications, or single executions (Figure 2 (Rybacki et al. 2012)).

At each of those steps we find specific methods designed for specific purposes. To contribute to or to apply these methods, it is crucial to locate them in the overall picture of modeling and simulation, and to know about the methods’ respective constraints.

So separation of concerns facilitates to focus your research, to make a contribution to a particular field, and to use the work of others. There is a huge portfolio of methods already available. Without separation of concerns, it is very likely to redo what others have already done, and even to do this poorly.

4 PITFALL 3: NO CLEAR SCIENTIFIC QUESTION

When I read Ernie Page’s PhD thesis in the midth 90s, I stumbled over a citation by Douglas Adam’s with which Ernie started the fourth chapter on formal modeling approaches to discrete event simulation: “It is very easy to be blinded to the essential uselessness of them by the sense of achievement you get from getting them to work at all.”(Page 1994, p.32). Since then I have been reminded of this citation quite often. Many publications in modeling and simulation show that something can be done (sometimes with a lot of effort) but why this has been done appears to have been lost (if it had ever been known) while doing it. So first one has to define what the scientific problem one wishes to solve is and, ideally, from there everything else should follow.

If you apply a formal approach, the question is what benefit lies in this. A declarative description of a model in a modeling formalism or language with a clearly defined semantics typically facilitates documenting, storing, manipulating, discussing, reusing, and analyzing models also by different simulation software. However, not all formalisms come with the same set of free gifts. Some modeling formalisms are so compact that even larger models can be fully described in a paper, e.g., the stochastic π-calculus (Priami et al. 2001), whereas other modeling formalisms are comparatively verbose, e.g., DEVS (Zeigler 1984), which might make a comprehensive presentation of realistic models difficult (see also Figure 1). A modular compositional design with clearly defined interfaces helps reducing complexity (by divide and conquer), and is particularly useful if different variants of a model are needed or models shall be reused. Some formal modeling formalisms come with a suite of different analysis tools, e.g., (stochastic) Petri nets – however again this is not true for all modeling formalisms. Also a modeling formalism or language might facilitate modeling in a particular application domain, however to provide support for this argument implies significant efforts (see Pitfall 5). Do not take this the wrong way, there are plenty of good reasons to take a formal approach towards modeling and simulation (actually more reasons than not to, see also Pitfall 6), however, you need to be clear which of those matter to you, and why the selected approach serves your needs best.

Independently whether you are developing or applying modeling and simulation methods, it is important to clarify the reasons you selected a particular formalism or method to work with. The question "why" should always loom large in the background of your research and whether expected benefits have been achieved deserves special attention (see Pitfall 5).

5 PITFALL 4: IMPLEMENTING EVERYTHING FROM SCRATCH

"... ERROR ... why scientific programming does not compute" (Merali 2010). Now it is out in the open: scientific programs (including simulation software) are of a poor quality and results cannot be trusted because programs are neither sufficiently tested nor documented, and as only few software products are released, the claimed results cannot be verified (see also Pitfall 5). The effort in developing a modeling and simulation tool is often underestimated. In addition to providing support for modeling, e.g., providing a model editor and implementing an execution algorithm, means to observe, store, and analyze data needs to be realized or integrated. A user interface is needed. More complex experiments might require a parallel execution, optimization algorithms etc. Before one notices one is busy implementing a lot of things that have little to do with the primary research. So at best, this development from scratch leads to wasting time. However, as typically neither time nor background for a thorough testing of all implemented functionalities exist, and the results depend on each of those functionalities working in a correct and efficient manner, the entire simulation results might be in jeopardy (Himmelspach and Uhrmacher 2009).

Now one might argue, if a new modeling language shall be designed, a new modeling and simulation tool needs to be implemented for this. However, an alternative might be to translate the newly developed language into an existing language and reuse the tools associated with this language (in this case only the translation between new and old language needs to be tested), another solution is to use one of the modeling and simulation frameworks around that are aimed at facilitating the development of modeling and simulation tools by reuse, e.g., Himmelspach and Uhrmacher (2009).

E.g., in JAMES II, to design a modeling and simulation tool that supports a new modeling formalism only two puzzle pieces have to be replaced: one to support the model (formalism) and one to support the execution of the model (simulator) (see Figure: 3). Everything else (with the obvious exception of the concrete model), incl. functionalities for parallelization, observation, data sinks and workflow documentation can be reused (Himmelspach and Uhrmacher 2007). Using such a plugin-based modeling and simulation framework also helps separating the different concerns in modeling and simulation (Pitfall 2). They can offer you a rich portfolio of methods for your research, e.g., more than 100 plug-in types and more than 700 plug-ins are currently available in James II of which about 50% are released as open source.

In addition, a suite of supportive sub-algorithms like event queues and random number generators is at the users disposal, which is of value, not only if the impact of sub-algorithms on the performance of different algorithms shall be tested (Jeschke and Ewald 2008). The time needed for implementation and

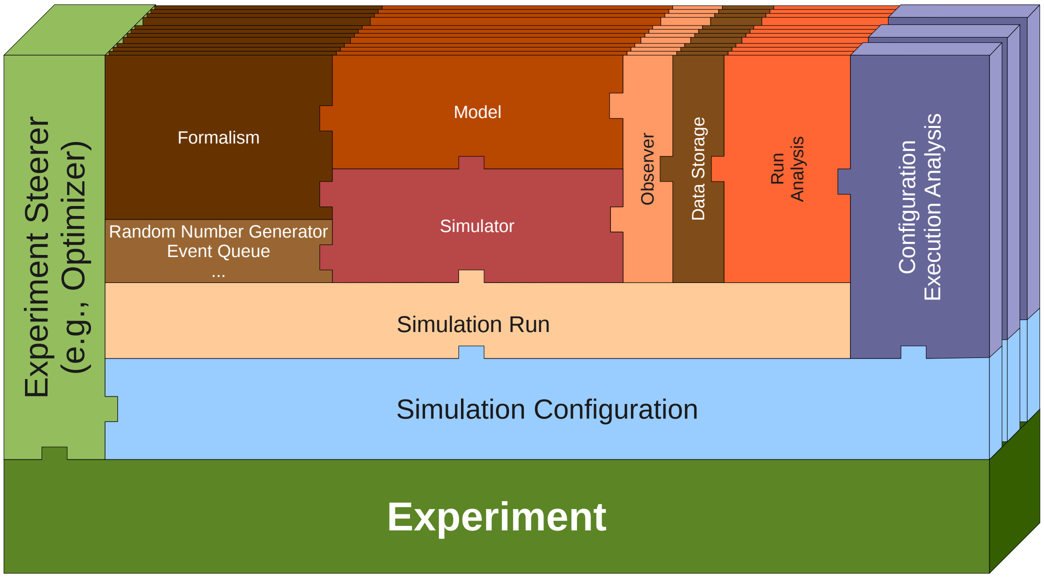


Figure 3: Plugin-based design of the modeling and simulation framework James II.

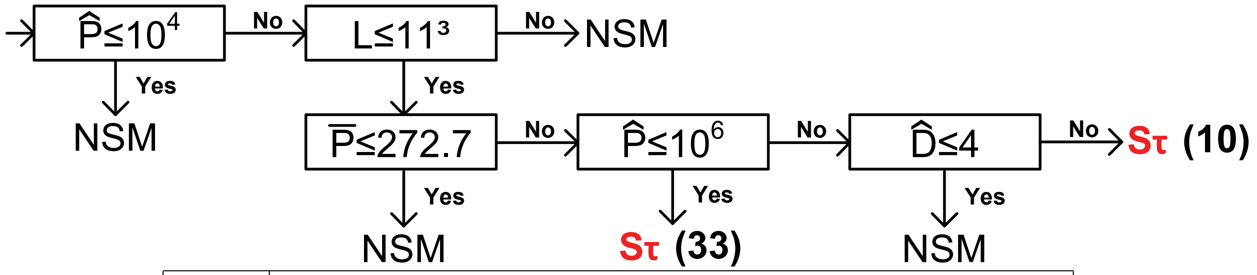
the potential sources for errors can significantly be reduced. In addition, reuse facilitates to relate one's own work to the work of others and will lead to a higher visibility (and future reuse) of the research done.

6 PITFALL 5: UNSUPPORTED CLAIMS

There is a plethora of claims that people like to make when it comes to modeling and simulation and often those appear to be hardly justified, (see also the discussion about overselling results in Wilson (2002)). The vagueness of claims tend to increase the less concrete the scientific question is (Pitfall 3) and the less strict a clear separation of concerns has been realized (Pitfall 2).

Lets first take a look at modeling approaches. One of the claims which should ring alarm clocks is if an author states that the developed or used modeling approach is more intuitive. Intuitive for whom is here the question: for the person who invented it? - this would be hardly surprising. If it shall be intuitive for others, how has this been shown? Do elaborate user studies with a representative set of users and different bench mark models exist to underpin that the modeling language is really intuitive in comparison to others? How has the cognitive load and the learning effort induced due to the modeling language be assessed? If a language allows a more compact description, how has this been demonstrated? Does a language allow to answer new questions in an application field?

To assess the efficiency of execution algorithms is also a non trivial matter. If one compares one execution algorithm with another, have possible sources for bias be identified and dealt with, as there are, implementations in different languages, algorithms implemented with different care, are the benchmark studies sufficiently representative and fair for all algorithms (or have they been rigged to serve the needs of one approach), has the influence of data structures (e.g., event queues) been considered etc. If the performance is not compared to other algorithms, what have been the benchmark tests, are those representative for the intended application field, and are those described in sufficient depth? If non-exact algorithms are used, how has the introduced approximation error been measured?



Symbol	Feature Description
$\hat{P}$	Maximum initial number of particles per sub-volume: $\max_{i\in[1,L]}\{\sum_{j=1}^{N} X_j^i(0)\}$
L	Number of sub-volumes.
$\bar{P}$	Average initial number of particles: $\left(\sum_{i=1}^{L}\sum_{j=1}^{N} X_j^i(0)\right)/L$
$\hat{D}$	Maximum diffusion rate: $\max_{i\in[1,N]} D_i$

Figure 4: A decision tree summarizing the results of a performance experiment.

A prerequisite for any meaningful claim (and thus for circumventing Pitfall 5) is to have a clearly defined scientific question (see Pitfall 3). For most modeling as well as simulation methods, the statement of Fujimoto (1990) (when assessing parallel discrete event execution algorithms at the beginning of the 90s) is still true and likely to remain so: “a general solution has proven to be elusive”, or to put it in different terms, no silver bullet does exist. So be very precise for which fraction of the problem space your solution is superior and provide sufficient justification for this. Although we like to state that our newly developed approach outperforms the other ones, typically the situation is more complicated.

E.g., a new approximative spatial algorithm for executing reaction diffusion models has been developed (Jeschke, Ewald, and Uhrmacher 2011). Species (particles) are located at a grid, diffuse from one sub-volume to another with a certain rate, or react with other species (particles) residing in the same sub-volume with a certain rate. To assess the new simulation algorithm, it has been compared with other exact and approximative methods in a performance study comprising several benchmark models. One result of the extensive performance study is depicted in the decision tree (Figure 4 from Jeschke et al. (2011)). It shows for which models the newly developed spatial τ-leaping method should outperform the exact Next Subvolume Method (NSM).

7 PITFALL 6: TOY DUCK APPROACH

Simulation is an experiment performed with a model. There exist plenty of literature about the many things you can do wrong with respect to experimentation (Schmeiser 2001; Edmonds and Hales 2003). Whereas, in some disciplines like biology or physics it is self-evident that to be of any value an experiment needs to be carefully designed and well documented (also to be reproducible), other disciplines lack this experimental background (Tichy 1998). Deficiencies in validity referring to model, experiments, and software (see also Pitfall 4) will endanger the credibility of the simulation study. The lack of valid experiments has been identified as a reason for the credibility crisis of simulation in the area of performance evaluation (Pawlikowski et al. 2002; Kurkowski et al. 2005). In an atmosphere where the credibility of modeling and simulation is at risk (and this is not limited to performance evaluation) it is of utmost importance not only to validate and verify models thoroughly (Galan and Izquierdo 2005) but also to provide readers with the resources to see for themselves what has been done.

Each of the steps of the modeling and simulation life cycle implies selecting (possibly adapting or even newly developing) and applying diverse methods, each of which influences the results of the modeling and simulation study and, thus, needs to be documented carefully for reproducing the experiment. A clear documentation of model, question of interest, experiment (both used for validating the model and for predicting and analyzing the behavior of the system based on this model), and software used needs to be provided in order to assess the results achieved and to enable a reproduction of these results (Figure 5). Thereby, it is important to note that those steps are typically non-monolithic. E.g., running experiments implies typically executing and analyzing single runs, replications, and different configurations. For executing replications of a stochastic model, the information required includes the model, the model initialization, the version of the simulation algorithm, the random number generator, the seed, what methods have been used to determine the length of the simulation, and the number of replications.

Some communities have started defining first guidelines how simulation experiments should be described (Waltemath et al. 2011). State of the art modeling and simulation software have started to provide support for automatic documentation and reproduction of experiments, e.g., by exploiting the potential of workflows (Rybacki et al. 2012). Often, it has been argued that the model and experiment description would not fit into a publication. However, this is no longer an excuse due to the possibility to publish supplementary material on-line. Thereby, all necessary information including the software can be made accessible.

In many experimental sciences, without sufficient information on how to repeat experiments, research results will not be published. If we want our discipline to mature and be no longer associated with something everybody can do, i.e., “to wind it up and let it run”, our standards need to change and it is always better being on the forefront of a development than slumbering behind.

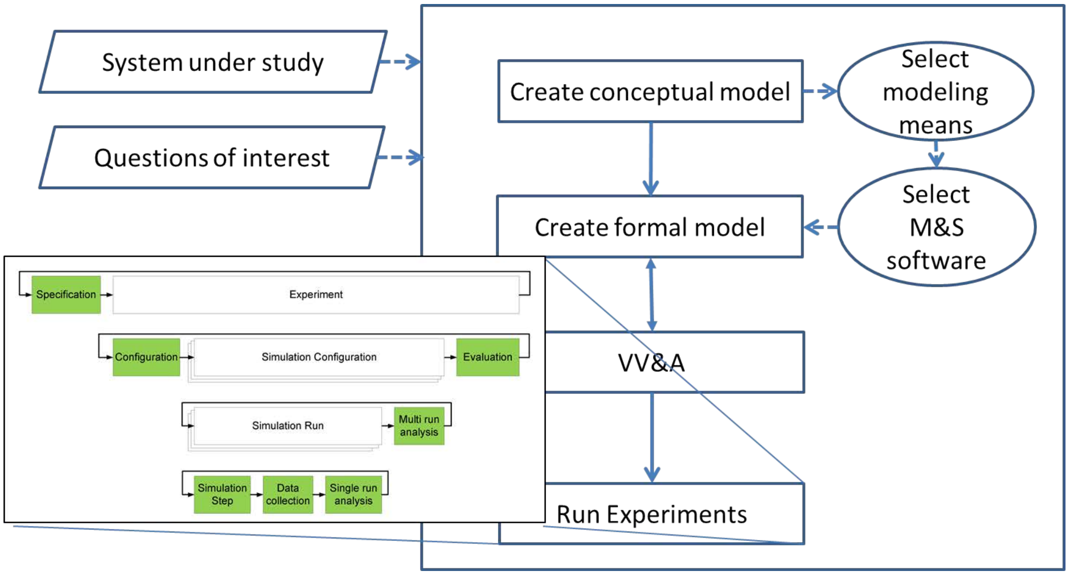


Figure 5: Diverse tasks in "in-silico experiments" that require documentation.

8 PITFALL 7: THE TUNNEL VIEW

Many authors in modeling and simulation appear mortally afraid to take a closer look at the work of others. The interdisciplinary nature of modeling and simulation should help looking across the boundaries of his or her research, however interestingly it seems to have the opposite effect. The different vocabulary used in different disciplines makes it harder to locate related work. Even if one has found it, it requires some effort to place what one did into the context of another research field, as it is easy to get the semantics of the terms wrong. Therefore, many authors rather stay in their own niche and clique, and are content with publishing “from the clique for the clique” (Parnas 2007).

The overall impact on modeling and simulation as a scientific discipline is dramatic. We see mostly conferences where people come to talk rather than to listen. Rigorous and detailed reviewing procedures of modeling and simulation conferences are discouraged with the convenient excuse of the involved different scientific cultures. However, our journals and our conferences are evaluated according to their impact and to the applied reviewing procedures. The top journals in modeling and simulation have in comparison to other disciplines a significantly lower impact factor. In addition, the entire modeling and simulation realm appears void of highly ranked conferences. However unreliable these measures and rankings are, they give an indication of the current state of our field. Our relation to rigorous reviewing, to publishing from the clique for the clique, and to related work needs to change to move modeling and simulation towards science and, pragmatically, to survive the next budget cut at our institutions. So if a conference does not give you detailed constructive reviews (independently of the acceptance rate), let the steering committee know about this and submit your work rather to another conference (constructive feedback is what conference submissions is mostly about), contact other authors directly if you assume a relation between your and their work, discuss differences with the work of others in detail, give credit to the work of others (this does not diminish your work, on the contrary, it makes it easier to appreciate your work), leave your clique once in a while (it can be a refreshing exercise), and be not afraid to challenge flimsy excuses.

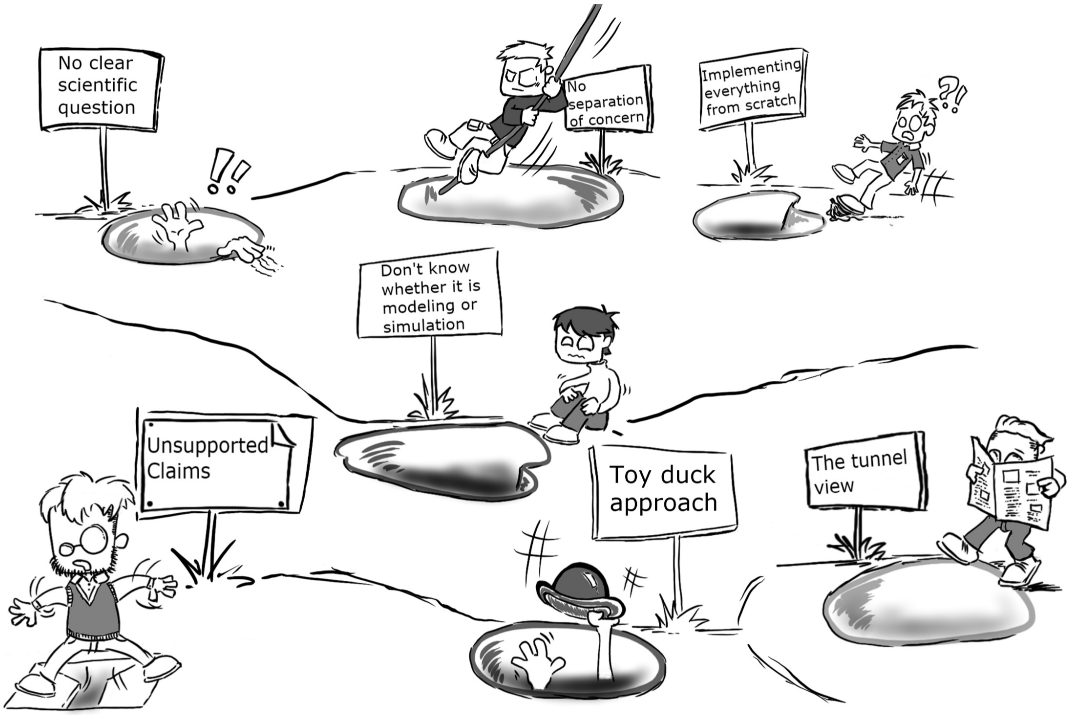

No clear scientific question
!!
No separation of concern
Implementing everything from scratch
?!
Don't know whether it is modeling or simulation
Unsupported Claims
Toy duck approach
The tunnel view

Figure 6: Pitfalls of modeling and simulation research (Illustration by Michael Wagner).

9 CONCLUSION

It might take a bit longer to reach your respective research goals by circumventing the above pitfalls (Figure 6). Other students might appear faster on the road - but in any case it should grant you (as a scientist) and your research results a longer life, and modeling and simulation a brighter future.

10 ACKNOWLEDGEMENT

I feel very lucky to have achieved a position at a University where I can freely pursue my research interests. I am also thankful for the support of the German Research Foundation (DFG): without it, a lot of my ideas would never have materialized. I have had the pleasure to know and to work with exceptional scientists during my career who have influenced my view on modeling and simulation. However, the most gratifying activity during my career has been to work with my students. I would like to thank my PhD students for their relentless enthusiasm for modeling and simulation, their creativity, and their dedication to get to the bottom of things and to get them done: Alke Martens, Jan Himmelspach, Mathias Röhl, Florian Marquardt, Mathias John, Matthias Jeschke, Roland Ewald, Orianne Mazemondet, Sabine Zinn, and currently Carsten Maus, Fiete Haack, Stefan Leye, Alexander Steiniger, Stefan Rybacki, and Arne Bittig. I learned from every single one of them. Also I would like to thank Michael Wagner, currently a bachelor student in my group, for his illustration of the 7 pitfalls (Figure 6).

REFERENCES

Banks, J., and L. Chwif. 2011. “Warnings about simulation”. *Journal of Simulation* 5:279–291.

Barth, R., M. Meyer, and J. Spitzner. 2012. “Typical Pitfalls of Simulation Modeling - Lessons Learned from Armed Forces and Business”. *Journal of Artificial Societies and Social Simulation* 15 (2): 5.

Brailsford, S. C., S. M. Desai, and J. Viana. 2010, December. “Towards the holy grail: Combining system dynamics and discrete-event simulation in healthcare”. In *Proceedings of the 2010 Winter Simulation Conference*, edited by B. Johansson, S. Jain, J. Montoya-Torres, J. Hugan, and E. Yücesan, 2293–2303. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Cellier, F. 1991. *Continuous systems modeling*. Springer.

Davis, P. K., and A. Tolk. 2007, December. “Observations on new developments in composability and multi-resolution modeling”. In *Proceedings of the 2007 Winter Simulation Conference*, edited by S. G. Henderson, B. Biller, M.-H. Hsieh, J. Shortle, J. D. Tew, and R. R. Barton, 859–870. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Dijkstra, E. W. 1982. “On the role of scientific thought”. In *Selected writings on Computing: A Personal Perspective*, edited by E. W. Dijkstra, 60–66. Springer-Verlag New York.

Edmonds, B., and D. Hales. 2003. “Replication, Replication and Replication: Some hard lessons from model alignment.”. *J. Artificial Societies and Social Simulation* 6 (4).

Faeder, J. R. 2011. “Toward a comprehensive language for biological systems”. *BMC Biology* 9 (68).

Fujimoto, R. M. 1990. “Parallel discrete event simulation”. *Commun. ACM* 33 (10): 30–53.

Fujimoto, R. M. 2011, December. “On innovation, and building and sustaining a successful career in research”. In *Proceedings of the 2011 Winter Simulation Conference*, edited by S. Jain, R. R. Creasey, J. Himmelspach, K. P. White, and M. Fu, 1–11. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Galan, J. M., and L. R. Izquierdo. 2005. “Appearances Can Be Deceiving: Lessons Learned Re-Implementing Axelrod’s ’Evolutionary Approach to Norms’”. *Journal of Artificial Societies and Social Simulation* 8 (3): 2.

Himmelspach, J., M. Röhl, and A. M. Uhrmacher. 2010. “Component-based models and simulation experiments for supporting valid multi-agent system simulations”. *Journal of Applied Artificial Intelligence* 24 (5): 414–442.

Himmelspach, J., and A. M. Uhrmacher. 2007, March. “Plug’n simulate”. In *ANSS ’07: Proceedings of the 40th Annual Simulation Symposium*, edited by H. Karatza and T. F. Znati, 137–143. Washington, DC, USA: The Institute of Electrical and Electronics Engineers, Inc.

Himmelspach, J., and A. M. Uhrmacher. 2009. “What contributes to the quality of simulation results?”. In *Proceedings of the 2009 INFORMS Simulation Society Research Workshop*, edited by L. H. Lee, M. E. Kuhl, J. W. Fowler, and S. Robinson, 125–129. University of Warwick, Coventry, U.K.: INFORMS Simulation Society.

Jain, S., R. Creasey, J. Himmelspach, and M. White, K.P.and Fu. (Eds.) 2011. *Winter Simulation Conference (WSC)*. IEEE Computer Science, ACM, Informs.

Jeschke, M., and R. Ewald. 2008. “Large-Scale Design Space Exploration of SSA”. In *Computational Methods in Systems Biology, International Conference, CMSB 2008, Rostock, Germany, October*

12-15, 2008, Proceedings, Volume 5307 of *Lecture Notes in Computer Science*, 211–230. Berlin / Heidelberg: Springer.

Jeschke, M., R. Ewald, and A. M. Uhrmacher. 2011. “Exploring the performance of spatial stochastic simulation algorithms”. *Journal of Computational Physics* 230 (7): 2562–2574.

Johansson, B., S. Jain, J. Montoya-Torres, J. Hugan, and E. Yücesan. (Eds.) 2010. *Winter Simulation Conference (WSC)*. IEEE Computer Science, ACM, Informs.

Kurkowski, S., T. Camp, and M. Colagrosso. 2005. “MANET simulation studies: the incredibles”. *SIGMOBILE Mob. Comput. Commun. Rev.* 9 (4): 50–61.

Law, A. M. 2003, December. “Designing a simulation study: how to conduct a successful simulation study”. In *Proceedings of the 2003 Winter Simulation Conference*, edited by S. Chick, P. J. Sánchez, D. Ferrin, and D. J. Morrice, 66–70. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Merali, Z. 2010. “Computational science: ...Error”. *Nature* 467:775–777.

Oren, T. 2011. “A Critical Review of Definitions and About 400 Types of Modeling and Simulation”. *SCS Magazine* 3:142–151.

Overstreet, C. M., R. Nance, and O. Balci. 2002. “Issues in Enhancing Model Reuse”. In *Grand Challenges In Simulation*: SCS.

Page, E. H. 1994. *Simulation Modeling Methodology: Principles and Etiology of Decision Support*. Ph.D. Dissertation, Department of Computer Science, Virginia Tech, Blacksburg, VA.

Parnas, D. L. 2007. “Stop the Numbers Game – Counting papers slows the rate of scientific progress.”. *COMMUNICATIONS OF THE ACM* 59 (11):19–21.

Pawlikowski, K., H.-D. Jeong, and J.-S. Lee. 2002, January. “On credibility of simulation studies of telecommunication networks”. *Communications Magazine, IEEE* 40 (1): 132–139.

Priami, C., A. Regev, E. Shapiro, and W. Silverman. 2001. “Application of a stochastic name-passing calculus to representation and simulation of molecular processes”. *Information Processing Letters* 80 (1): 25–31.

Robinson, S. 2002. “General concepts of quality for discrete-event simulation”. *European Journal of Operational Research* 138 (1): 103 – 117.

Rybacki, S., J.-S. Lee, J. Himmelspach, and A. Uhrmacher. 2012. “Template and Frame based Experiment Workflows in Modeling and Simulation Software with WORMS”. In *International Workshop on Scientific Workflows.*

Schmeiser, B. W. 2001, December. “Output interpretation: some myths and common errors in simulation experiments”. In *Proceedings of the 2001 Winter Simulation Conference*, edited by B. A. Peters, J. S. Smith, D. J. Medeiros, and M. W. Rohrer, 39–46. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Siebers, P., C. Macal, J. Garnett, D. Buxton, and M. Pidd. 2010. “Discrete-Event Simulation is Dead, Long Live Agent-Based Simulation!”. *Journal of Simulation* 4(3):204–210.

Tichy, W. F. 1998. “Should Computer Scientists Experiment More? 16 Reasons to avoid experimentation”. *IEEE Computer* 31 (5): 32–40.

Uhrmacher, A., and D. Weyns. (Eds.) 2009. *Multi-Agent Systems: Simulation and Applications - Methodological Guidelines for Modeling and Developing MAS-Based Simulations*. CRC Press/Taylor & Francis.

Waltemath, D., R. Adams, D. A. Beard, F. T. Bergmann, U. S. Bhalla, R. Britten, V. Chelliah, M. T. Cooling, J. Cooper, E. J. Crampin, A. Garny, S. Hoops, M. Hucka, P. J. Hunter, E. Klipp, C. Laibe, A. K. Miller, I. Moraru, D. P. Nickerson, P. M. F. Nielsen, M. Nikolski, S. Sahle, H. M. Sauro, H. Schmidt, J. L. Snoep, D. Tolle, O. Wolkenhauer, and N. L. Novere. 2011. “Minimum Information About a Simulation Experiment (MIASE)”. *PLoS Computational Biology* 7(4).

Wang, Z., and A. Lehmann. 2010. “Quality Assurance of Models and Simulation Applications”. *International Journal of Modeling, Simulation, and Scientific Computing (IJMSSC)* 1 (1): 27–45.

Wilson, J. 2002. “Responsible Authorship and Peer Review”. *Science and Engineering Ethics* 8 (2): 155–174.

Zeigler, B. 1984. *Multifacetted Modelling and Discrete Event Simulation*. London: Academic Press.

AUTHOR BIOGRAPHY

ADELINDE M. UHRMACHER is professor at the Institute of Computer Science of the University of Rostock in Germany. Her work focuses on developing modeling and simulation methods for discrete event systems and their applications in diverse areas, e.g., demography, computer science, and cell biology. Methods for multi-level, variable structure modeling, parallel and lately also approximative and multi-resolution spatial simulation approaches play a central role. Most of her work has been implemented as part of the plug-in based modeling and simulation framework James II. She has published more than 100 peer-reviewed papers on modeling and simulation. Her scientific visibility is reflected by various roles at the editorial board of simulation related journals, incl. Editor in Chief of SCS Simulation (2000-2006), Department Editor Computer Technologies and Information Systems of the IIE Transactions of Operations Engineering (2001-2003), Associate Editor of LNBI Transactions on Computational Systems Biology - Springer, Associate Editor of BMC Systems Biology, and Area Editor of the ACM Transactions of Modeling and Computer Simulation (TOMACS).