Modelling, Uncertainty, and Complex Systems

Fiona A. C. Polack
f.a.c.polack@keele.ac.uk
Keele Univeristy
Keele, Staffordshire, UK

Richard F. Paige
paigeri@mcmaster.ca
McMaster University
Canada

Jon Timmis
jon.timmis@sunderland.ac.uk
Sunderland University
UK

ABSTRACT

TO ADD

KEYWORDS

Complex system simulation, uncertainty

ACM Reference Format:
Fiona A. C. Polack, Richard F. Paige, and Jon Timmis. 2020. Modelling, Uncertainty, and Complex Systems. In *Models '20: ACM / IEEE International Conference on Model Driven Engineering Languages and Systems October 18–23, 2020, Montreal, Canada.* ACM, New York, NY, USA, 7 pages. https://doi.org/

1 INTRODUCTION

A complex system is a system in which the perceived (high level) behaviour of the system is a consequence of many lower-level behaviours. Complex systems include all natural systems, as well as systems of interacting engineered systems where there is also interaction with environmental sensors or people (cyberphysical systems, systems of systems, large-scale complex IT systems). Uncertainty is inherent in complex systems domains. By contrast, and to emphasise the characteristics of a complex system, an engineered system can be thought of as a complicated system: one that can be decomposed and reassembled with no loss of functionality. In fact, engineered systems are designed to negate the effect of complexity arising from real-world behaviour of the system environment and in the substrate of the system.

The specific area of focus in this paper is agent-based models of complex systems, such as those used in laboratory research or robotic simulation. Uncertainty is inherent in modelling and simulating complex systems, but has been neglected relative to attention paid to mitigating uncertainty in engineered systems. We need to be able to identify (types of) uncertainty, to evaluate the effect of uncertainties, and to develop a language and tools for describing and analysing uncertainties.

In simulating complex systems, we cannot talk about validity as an absolute. We can do conventional software testing (and even, in some cases, proof of properties), which improves our confidence in the software as a product, but it has been widely shown that a simulation can produce similar results to reality from very different concepts and behaviours[27]. A complex systems simulation that is usable as a laboratory research or engineering tool has to be shown to be generating behaviours in an analogous way to reality (i.e. there must be a clear mapping and traceability between real-world and simulation concepts and behaviours). In addition, because complex behaviour is very hard to mimic, the simulation has to be tuned (calibrated) and the sensitivity of its many parameters investigated [5, 30].

Simulation validity is further contingent on the current understanding of the domain, the current representations of the domain, and the computational models and code derived to create a simulation. To avoid the misunderstanding that a valid simulation is "right", we refer to the fit-for-purpose of a simulation, acknowledging and working with uncertainty. Notably, the acts of modelling, simulation and experimentation lead to discovery of new uncertainties. The analysis of simulation results and understanding of how results map to the (real) domain of study must take account of the effects of uncertainty.

This position paper describes how we might provide means to record uncertainties as they are identified, as well as a classification and modelling language to express uncertainties, and ways to analyse and evaluate the effect of uncertainties. Specifically, we address issues in the following areas:

- language design – metamodelling and modelling with uncertainty for a simulation domain, and generalisation of uncertainty language;
- analysis of the effects of uncertainties in complex systems simulation and experimentation.

1.1 Simulation of complex systems

Our focus is on simulation of complex systems to support research or engineering. Our work builds on the CoSMoS Process [37], which is summarised in Figure 1.

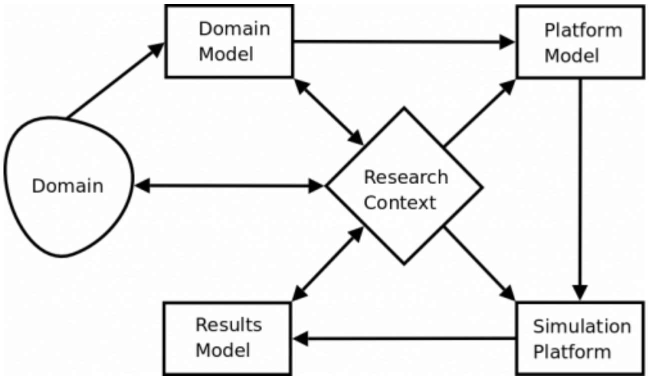

Figure 1: The Cosmos Process, after [37]

Starting from the Domain, CoSMoS is predicated on close collaboration between a domain expert (a specific researcher or laboratory) and a developer (one or more simulation engineers). Knowledge about the domain is always mediated by the domain expert, so that the simulation and its design represents a view that can be recognised and agreed by the domain expert. Development proceeds by capturing what the simulation will do (a purpose or hypothesis, and key concepts and behaviours from the domain implicated in the real-world behaviour) in a domain model – this is an agreed basis for simulation between domain expert and developer. The simulation engineer assumes the lead at this point, devising a platform model from the domain model: this is typically a set of state diagrams that express the behaviour of first real and then computational agents, often complemented by other UML behaviour diagrams such as activity diagrams. The development of the simulation platform is recommended to be as systematic as possible (plans to realise software using model transformation have not yet been fully realised [29]). Simulation experiments are run on the simulation platform and produce simulation results. A simulation result can be compared to the real world, but, crucially, is not a real world observation. Throughout development, a record must be kept of sources, assumptions, interpretations, design decisions, and other collateral information (the research context; this information is used to inform the analysis of simulation results and interpretation back to the real world. Examples of research using CoSMoS, alongside domain experts in a range of immunology areas, can be found in PhD theses and associated papers (e.g. [1, 20, 32, 39] and other research from the former York Computational Immunology Laboratory). These researches also demonstrate the use of argumentation [3, 37] to document the basis of belief for the fitness for purpose of models, simulation, statistical and other analysis methods.

The CoSMoS outline gives us a basis for identifying where uncertainty arises in a domain, modelling and simulation, and a pointer to how different forms of uncertainty manifest and can be handled. Figure 2 extends the key products of the CoSMoS process (from Figure 1, showing all the activities contributing to the demonstration of fitness for purpose, the overarching quality management and argumentation that supports the use of a complex simulation in a research context.

Identification of uncertainties would be incremental, and can occur at any point. We need to incorporate identified uncertainty into models, to evaluate their nature and effect, and to document our evaluation of the uncertainty. During calibration and use of a simulator, we continue to identify, evaluate and document uncertainties. The outputs of our proposed uncertainty activities (parallelograms in Figure 2) comprise uncertainties that are amenable to quantification or description or are mitigatable, as well as residual uncertainties (known and unknown unknowns).

2 EXAMPLES OF UNCERTAINTIES

To illustrate some of the origins and consequences of uncertainty, we present two motivating examples, which are not specific to any one research study but capture a range of our experiences in creating and using complex systems simulations.

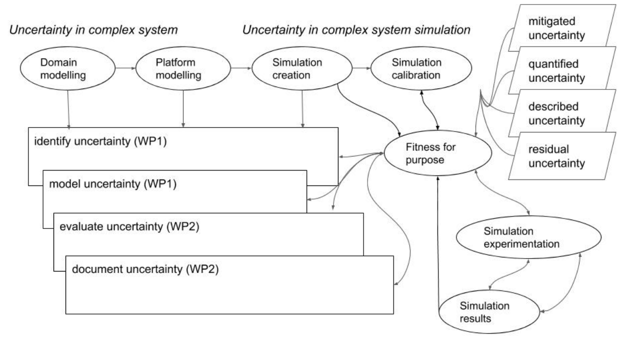

Figure 2: Uncertainty and the Cosmos Process: ovals represent the CoSMoS products from Figure 1, with the additional steps required for demonstrating fitness for purpose and experimental use of the simulation; uncertainty-related activities are rectangles, and their outputs are parallelograms; arrows are indicative not comprehensive.

2.1 Motivating example 1: laboratory research simulation

It is often the case that a biologist needs to understand an *in vivo* process such as an aspect of immune system development, but existing technologies limit their ability to observe the behaviours of the internals of a living organism: ethics limit what can be achieved with live subjects; even non-invasive techniques (e.g. external detection of the location and movement of ingested dyes) modify behaviours, either through chemical changes or through the consequences of stressing or anaesthetising the subject. Simulation offers an ethically sound, repeatable, and extensively modifiable experimental platform, and, once sufficient knowledge (e.g. from *in vivo* and *in vitro* experimentation) exists, can provide a route to confirm the feasibility of theories and a source and test-bed of new hypotheses.

Our example assumes that there is a clear hypothesis, but it can only be tested by perturbing or destroying a living organism. A simulation developer – typically a software engineer – works with the biologist, using a principled approach such as CoSMoS to create a simulator [1, 26]; they agree the design, and the compromises needed to digitise and execute the design. The simulation uses suitable analogies of known cell mechanisms, and they agree that it is fit for its defined research purpose. After calibration, experiments are run *in silico*.

Calibration and experimental results highlight differences between the simulation and the observed biology. The differences could indicate uncertainties in the domain knowledge, but might equally be side-effects of design or implementation decisions – or even statistical aberrations. The biologist and software engineer need considerable effort and expertise to understand and analyse the simulation results, and to relate simulation findings to the biological domain.

Key research questions, relating to uncertainty in research simulations, are:

(1) for each real-world concept, what is the initial assessment of uncertainty of domain knowledge?
(2) for each simulation concept, how do design and implementation decisions diverge from the real world concept, and what additional uncertainty do these divergences introduce?
(3) for each element of uncertainty, can the uncertainty be reduced (e.g. a simulation can test across a range to identify likely parameter values, which can be checked biologically); quantified (e.g. identify that a behaviour is observed with a particular probability; extrapolate from simulation results or reality); or simply described (e.g. we know that simulating a 3D reality in 2D introduces uncertainty but not much more).
(4) can we evaluate and argue the reliability of simulation results? Can we distinguish conditions under which results are likely to be reliable from those where we are less confident of the results (e.g. our PPSim [1, 6] can reliably simulate formation of a Peyers' patch, but not the termination of formation or the distribution of patches).
(5) how, and in what circumstances, can we measure the reliability of simulation results? (e.g. when Simomics uses simulation in drug discovery, can it quantify reliability so that clients can focus their R&D appropriately?)

Addressing these key questions has the potential to provide clear and systematic understanding of sources and effects of uncertainty: (a) clear definition of scope, scale, purpose and simulation hypotheses; (b) guidance on where to focus activities to calibrate and test a simulator; (c) confidence in what the results reveal about the system and how results relate to real biological behaviours.

2.2 Motivating example 2: robotics behavioural simulation

Robotics uses agent-style simulation in development of behaviours of individual robots and, particularly, of robot swarms. For instance, a roboticist creating an evolutionary algorithm works in simulation, eventually implementing either evolved behaviour or a successful evolutionary algorithm in an embodied system. Similarly, robot swarm algorithms are typically created (or evolved) and tested *in silico* before embodiment in swarm robots. A roboticist designs the control software on a publicly-available simulation platform that encodes the mechanics and physics of the style of robot and the environment. When evolving individual or swarm robot algorithms, the evolutionary algorithm are created and run in simulation, and then installed on actual robots.

The behaviour of each real robot always differs both from the simulation and the behaviour of other instances of the same embodied robot. The robotics term for this is the *reality gap*. Divergence arises from the uncertainties of modelling the robot and its environment and the uncertainties in the mechanics and set-up of real robots.

It is futile to try to engineer out uncertainty in robotics: real environments are complex and uncertain; robotic systems exploit the environmental and embodied uncertainties to evolve complex behaviour. Key research questions relating to uncertainty in robotic evolution, are:

(1) can we identify systematic variation between simulation and embodiment (e.g. for one type or robot or one type of environment), and can we quantify or describe the variation?
(2) can we identify and quantify the effects of the different characteristics of uncertainty in the real robots and of noise introduced to a simulated robot or swarm?
(3) can we identify characteristics of uncertainties of the real environment that are used by embodied robots or swarms in ways that are different to the noise introduced to a simulated environment, and can we quantify these effects?
(4) can we identify characteristics of uncertainty in both real and simulated environments that provoke first-order effects (e.g. generalization or noise-robustness) in evolutionary systems?
(5) can we reduce the reality gap such that simulation becomes a more effective design tool?

A clear and systematic understanding of the sources and effects of uncertainty in robotic systems would (a) allow calculation of uncertainty bounds that can be considered in both evolution and embodiment; and (b) provide guidance on how simulations can be tuned to a better representation of a specific context and environment.

2.3 Examples: commentary

In the domains of both motivating examples, complexity and uncertainty are inherent. Biological behaviour is complex on many levels, and complexity is inherent to the context of the behaviours under investigation. The mechanics and environment of an embodied robot or robot swarm are inherently complex, and it is infeasible to express all the complexities of realities in a simulation. In both domains, a systematic way to capture and express uncertainty would be of benefit in simulation design, analysis, use, and comparison with (or embodiment in) the real world.

3 RELATED WORK

Uncertainty has been widely researched, but mainly to eliminate or control it. By contrast, in simulating complex systems, we need to retain and incorporate uncertainty, and to understand the influence of uncertainty on simulation experimentation.

There are many existing ontologies and taxonomies of uncertainty, and some metamodels, which offer a starting point for identification or classification of uncertainty. There are also many approaches for quantification or mitigation of uncertainty.

The process of identifying uncertainty is something of a circularity: to detect uncertainty, we need to know what sort of thing we are seeking. A common indicator of uncertainty is the implicit or explicit assumptions made in modelling.

The most prominent protocol for agent modelling, originating in the ecological community, is the ODD (Overview, Design concepts, Details) Protocol [12, 13]. ODD is recommended by agent-based communities such as CoMSES OpenABM[1], along with data protocols, such as the FAIR principles [38]. ODD originated as a protocol to improve the scientific reproducability of simulation results, and provides a template to document the design and rationale of the simulation. However, there is no explicit recording of assumptions, only the source or rationale of design decisions. The protocol has

[1] https://www.comses.net/

nowhere to express or explore what is not definitively known in the domain, design or use of the simulation. Similarly, the FAIR principles focus on complete and consistent documentation of (meta)data, but do not consider documentation of data uncertainties.

The CoSMoS process [37], by contrast, puts emphasis on identifying and recording assumptions, and on arguing the fitness for purpose – and thus also the limitations – of the simulation. It has been argued that the ODD protocols and CoSMoS are complementary [36], but there has been no attempt create an alignment.

3.1 Identifying and evaluating uncertainty

There are many existing attempts to define uncertainty. Jousselme et al. [14] review many existing hierarchies and ontologies, in their search for a comprehensive set of definitions on which to build an understanding of uncertainty. Basing their contribution on situational analysis, they distinguish various types, ranging from ignorance (a state of mind) and uncertainty (a consequence of limitations in observation).

Padulo and Guenov [22], working on engineering design, also present an extensive review, and crucially note that design decisions are not rationally-justified decisions taken once all facts and options are available, and that the design process itself results in a growth of knowledge – and, presumably, a reduction in uncertainty. Padulo and Guenov provide a useful high-level summary, in which the design problem is seen as separable into uncertainty about and uncertainty within the problem (Figure 3). Uncertainty about the problem includes issues such as assumptions, values and external influences, whilst uncertainty within the problem comes from experts, data, models, design approaches and imposed standards.

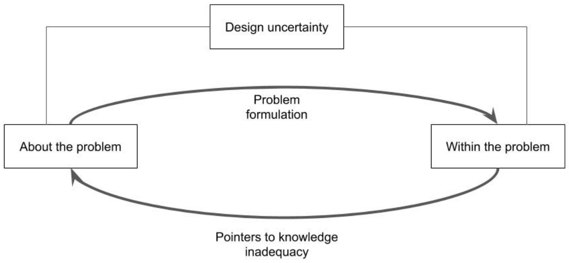

Figure 3: Classes of design problems relating to uncertainty, summarising the scheme of Padulo and Guenov [22]

Both the above works touch on the issue of quantification of uncertainty. Knight [17] separates quantifiable effects, which he labels *risk* and describes as randomness expressed by probability and statistics, from uncertainty, which is any other sort of unknown effect. Lo and Mueller [19] extend the classification of uncertainty to *fully reducible*, *partially reducible* and *irreducible*, noting that what is irreducible in one domain (and with the current state of knowledge) may be (partially) reducible in others.

Many researchers define uncertainty as *systemic* or *epistemic* uncertainties, denoting things that could be known but are not, such as missing parameters, variables, structures, unknown distributions or imprecise algorithms [34]. There is extensive research on approaches that can turn systemic or epistemic uncertainties into *aleatoric* uncertainties – things that are not precisely known but vary in ways that can be expressed in statistics (e.g. Dai et al. [8]) or mathematics (e.g. [21, 24]), in order to quantify and manage reducible uncertainty. For example, Kennedy and O'Hagan [16] present Bayesian approaches to calibration for seven types of uncertainty.

Identification of uncertainty is inherent in engineering disciplines that seek to mitigate or control uncertainty (e.g. MIT Uncertainty Quantification Group; aeronautic uncertainty [23, 33, 41]; sources of uncertainty in self-adaptive systems [9]).

A more colloquial approach is to label uncertainty as known unknowns and unknown unknowns. However, within each of these categories, there are reducible and irreducible uncertainties, quantifiable, qualifiable (or describable) and merely identified uncertainties.

3.2 Modelling uncertainty

There has been growing interest in modelling uncertainty, spearheaded by an OMG call for proposals and the OMG Uncertainty group[2]. The work includes a draft metamodel of uncertainty in cyberphysical systems [40]. The focus is on capturing the (un)certainty of the modeller by expressing beliefs about information. The state of the art is captured clearly by Vallecillo, in his keynote presentation to QUATIC2019[3]. Vallecillo recognises the wide range of classifications of uncertainty, and summarises the origins of uncertainty as comprising: under-specification; lack of knowledge; lack of precision; imperfect, incorrect or missing information; approximation; indeterminacy; and different interpretations of the same information by different people.

3.3 Arguments acknowledging uncertainty

A potential complementary approach to modelling uncertainty through belief is used in critical systems engineering and in the CoSMoS simulation approach. Rather than representing uncertainty directly in engineering models, an argument that acknowledges the weaknesses of the models is created.

Argument use in critical systems engineering is encapsulated by safety case argumentation: for example, ISO61508(general), ISO26262(automotive), and DO-178B(airplane) require documentation of "Safety Cases". A typical example, the safety of aircraft systems, is presented as an argument over hazards and mitigations [15]. The structure of arguments is most commonly presented using a notation such as the Goal Structuring Notation: a claim is asserted, with associated assumptions and elaborations; a strategy for evaluating the claim is chosen; sub-claims and strategies are stated; and eventually evidence supporting sub-claims is presented, such that the combination of all evidence amounts to a sufficient demonstration that the claim is met [35]. The arguments allow manufacturers to support the general assetion that the risk associated with their system is *as low as reasonably practicable*.

[2] http://www.omgwiki.org/uncertainty/oku.php?id=start

[3] Modeling and Evaluating Quality in the Presence of Uncertainty, https://www.slideshare.net/avallecillo/modeling-and-evaluating-quality-in-the-presence-of-uncertainty

The CoSMoS team and York Computational Immunology Group has adapted argument notations to present the rationale for a belief in the fitness for purpose of complex systems simulations [2, 3, 7]. Arguments of fitness for purpose can be created for, for instance, models, implementation (design decisions), analysis techniques (e.g. statistical tests), and experiments. Like a critical systems argument [18], the validity of the argument depends on the current context and use of the simulation: adaptation or reuse requires a revisiting of fitness for purpose.

3.4 Tools

There is a growing range of tools to support the various facets of complex simulation engineering.

It has long been acknowledged that uncertainties of mapping domain models through to a simulation can be partially addressed by domain specific modelling languages and workflows [28, 37]. Early work used metamodelling to define domain specific modelling approaches (e.g. [25]), whilst several initiatives now seek to develop domain specific modelling languages (DSMLs) for simulation [29].

Tools exist to support systematic calibration, sensitivity analysis, and statistical analysis of experimental results. For example, Spartan[4] [4, 5] is an R-based package of statistical techniques specifically designed to explore the relationship between a complex research simulation and the real-world. As such, many of its techniques focus on analsyis and reduction of uncertainty in complex systems simulations. ASPASIA[5] [10] supports analysis of simulated biological behaviour that depends on an intervention, and the concomitant issues of calibration, which introduce uncertainty in the interpretation of experimental results – e.g. in assessing the extent to which an alteration in behaviour can be attributed to the intervention alone. Clarity on the classification and identification of uncertainty can lead to better application of the techniques for calibration, sensitivity analysis and statistical analysis of results supported by these tools.

A range of commercial tools to support documentation, argumentation and analysis of simulators (notably the virutal laboratory and associated tools offered by Simomics Ltd[6]). Such tools already provide means to document uncertainties, but need to be able to apply the classification and evaluation of effects of uncertainties.

Whilst most of the tools identified relate to research simulations (and specifically, research in immunology), some have been adapted for robot simulation and engineering. For example, the York Robotics Lab has developed RoboSpartan[7], to support optimisation of the behaviours of robotic systems and targeted experimentation in hardware, specifically:

- automated parameter value sampling and result analysis for uncertainty and sensitivity analyses;
- generated simulation configuration files;
- generated execution scripts;
- surrogate model generation, for use where a model analysis proves intractable;
- support for evolutionary and Bayesian computation techniques to identify parameter regions giving rise to desired behaviours.

4 DISCUSSION: UNCERTAINTY IN COMPLEX SYSTEM SIMULATION ENGINEERING

To advance the state of the art in complex systems simulation, we need to bring together learning from a wide range of uncertainty research, identifying how each can be used in complex systems simulation engineering, embodied here in the CoSMoS process [37]. The goals are to enable identification, modelling, evaluation and documentation of uncertainties and their effects.

A classification of types of uncertainty that applies across complex systems simulation engineering needs to build on the work of authors such as Padulo and Guenov [22], Jousselme et al. [14] and the OMG Uncertainty Group. However, there is a need to go beyond a belief- or structural-analysis-based approach. In complex systems simulation engineering, there are definitively-known uncertainties, such as parameters whose absolute value is unknown but whose range or distribution of values is not disputed. These uncertainties are distinguishable from uncertainties that are known but not quantifiable, such as the effect on simulation of digitising space and time, and those only identifiable by observation (e.g. a systematic but unexplained divergence between observed and simulated behaviour).

Once a classification of the generic uncertainties of complex simulation engineering is established, we need to understand how the classification applies and adapts in particular complex domains and on specific simulation subjects. There will, for instance, be variations in forms and instances of uncertainty manifested between and within research domains (e.g. uncertainties typical to simulated immunological behaviours will differ from uncertainties in simulations of collective human behaviour), whilst engineering simulations (here, for swarm robots and evolutionary robotic design) will manifest different variants of uncertainty to research simulations.

Following on from identification and classification of uncertainty, researchers can start to generalise analysis and treatment of uncertainties. As in other domains some types of uncertainty could be reduced or eliminated, though the engineering value of reducing uncertainty is an important consideration: for instance, if more certainty in parameterisation or agent interaction reduces the number of runs of a simulator need to achieve statistically valid results, the cost of the increased certainty needs to be weighed against the cost of each simulation run. If it takes significant effort (e.g. undirected laboratory exploration) to increase certainty of parameters or interactions, but each simulation run takes milliseconds, and 10 000 runs is sufficient to achieve statistically reliable results, the cost of certainty is almost certainly too high. Conversely, if a tool-based calibration that takes a few hours to complete reduces the number or speed of execution runs so that experimentation time (*in silico*) reduces from months to hours, then the certainty-increasing intervention is likely to be cost-effective. Cost-effectiveness may also depend on the use of the simulation results. For instance, in a multi-million dollar drug development programme, a few days, or even months, spent improving the precision and reliability of

[4] https://www.kieranalden.info/index.php/spartan/
[5] https://www.kieranalden.info/index.php/aspasia/
[6] toolsoriginatingintheworkoftheCoSMoSteamandYorkComputationalImmunologyLab; https://www.simomics.com/offerings
[7] https://www.york.ac.uk/robot-lab/robospartan/

a simulation of complex interactions with the selected drug(s) is likely to be feasible; the same reductions of precision in a laboratory research simulation where outcomes can be quickly validated *in vivo* or *silico* would likely not be cost-effective. In many cases, we anticipate that greater effectiveness will come through identifying, evaluating and documenting the uncertainties, than through actions taken to reduce simulation uncertainty.

There is ample literature on handling uncertainties that are amenable to quantification. These range from creating statistically valid results sets to determine the statistical significance of results, to using known distributions to determine the values of specific parameters in each agent in a simulation [4, 5, 31]. Two examples, from our existing research simulations, are as follows.

- In research to tie down the specifics of uncertain low-level behaviours known only from limited time-interval observation in the laboratory, simulation experimentation was able to pinpoint the timing and extent of interactions implicated in Peyer's Patch formation, which were subsequently tested and accepted in the laboratory [1, 6].
- In research to isolate values for parameters approximated by the domain expert, an AI search over the set of uncertain parameters yielded sets of candidate values; working with the domain expert, this led to identification of the best real-world match, and specifically to identification of one parameter whose uncertain value was then shown to have caused divergence between the simulated and real observations; the result was again validated in the laboratory [11].

However, we must recognise that a statistical characterisation does not offer understanding: the uncertainty of values or computational behaviour may be reduced, but uncertainty remain in how the real world achieves an observed behaviour. Furthermore, it is a feature of complex system simulation (whether for research or engineering) that the models and implementation actually characterised the combined effect of many unknowns. Where we use quantitative modelling or analysis of uncertainty, there is a critical need to record the rationale and likely side-effects of quantification. Argumentation of fitness for purpose already captures some information about uncertainties, but needs to be adapted to systematically record uncertainty. Since an argument of fitness for purpose is only a structural summary of the rationale for our confidence in simulation, we also need a systematic process and template for documenting what we do know about each type of uncertainty, whether related to the domain, simulation, experimentation or results.

An ultimate goal would be to determine system-level uncertainty: to be able to combine known and unknown uncertainty evaluations in a way that allows a summary of the overall effect of uncertainty, and an evaluation of the consequent reliability of results or observations on simulation.

5 CONCLUSION

Uncertainty pervades any complex system, whether in reality or simulation. There is a wealth of writing on the classification of uncertainty, but rather less on domain-specific uncertainty, its identification, description and evaluation.

Systematic complex systems simulation engineering, as advocated by CoSMoS, provides a basis for exploring how uncertainties in research and engineering domains and simulations can be identified, modelled, evaluated and documented. The state of the art in other areas of uncertainty research offer directions to take in relation to mitigation, quantification and description of uncertainty. However, we recognise that, when working with uncertainty, there are likely to be residual uncertainties (such as systematic but unexplained differences between simulated and observed behaviour) that can only be recorded.

An approach to manage uncertainty in complex systems simulation and experimentation needs to address not just design practices but also the design or experimentation, rationale or fitness for purpose, and analysis of results. In these areas, existing tools can be enhanced to model and address aspects of uncertainty.

We envisage that the management of uncertainty in research and engineering simulation of complex systems would have direct application alongside existing simulation and support tools in reducing development time and costs. For instance, the pharmaceuticals industry now employs simulation in drug discovery, but lacks a way to record and evaluate the unknowns and uncertainties of the domain and simulated system: clear, partly-quantifiable, evaluation of the limitations and caveats of the simulation will help to reduce costs and improve reliability of simulation use in drug discovery.

In robotics, improved understanding and observance of uncertainty should lead to simulation platforms that better represent the context and environment of individual or swarm robots, prior to embodiment. Capturing and analysing cumulative uncertainties in the simulator and domain has the potential to make robotic systems development more effective; we can also support simulation of service or maintenance robotic systems in situ, exploring the evolved behaviour, safety, and reliability of robots in realistic domains with improved (and sometimes quantifiable) likelihood of successful embodiment.

REFERENCES

[1] Kieran Alden. 2012. *Simulation and Statistical Techniques to Explore Lymphoid Tissue Organogenesis.* Ph.D. Dissertation. University of York. http://etheses.whiterose.ac.uk/3220/.

[2] Kieran Alden, Paul S. Andrews, Fiona A. C. Polack, Henrique Veiga-Fernandes, Mark C. Coles, and Jon Timmis. 2015. Using argument notation to engineer biological simulations with increased confidence. *J. R. Soc. Interface* 12, 104 (2015), online. https://doi.org/10.1098/rsif.2014.1059

[3] Kieran Alden, Paul S. Andrews, Jon Timmis, Henrique Veiga-Fernandes, and Mark C. Coles. 2011. Towards Argument-Driven Validation of an in-silico Model of Immune Tissue Organogenesis. In *ICARIS (LNCS)*, Vol. 6825. Springer, 66–70.

[4] Kieran Alden, Mark Read, Paul S Andrews, Jon Timmis, and Mark Coles. 2014. Applying spartan to Understand Parameter Uncertainty in Simulations. *The R Journal* 6, 2 (2014), 63–80. https://doi.org/10.32614/RJ-2014-025

[5] Kieran Alden, Mark Read, Jon Timmis, Paul S. Andrews, Henrique Veiga-Frenandes, and Mark S. Coles. 2013. Spartan: A Comprehensive Tool for Understanding Uncertainty in Simulations of Biological Systems. *PLoS Comp. Bio.* 9, 2 (2013), e1002916. https://doi.org/10.1371/journal.pcbi.1002916

[6] Kieran Alden, Jon Timmis, Paul S. Andrews, Henrique Veiga-Fernandes, and Mark C. Coles. 2012. Pairing experimentation and computational modelling to understand the role of tissue inducer cells in the development of lymphoid organs. *Frontiers in Immunology* 3, 172 (2012), online. https://doi.org/10.3389/fimmu.2012.00172

[7] Paul S Andrews, Fiona Polack, Adam T Sampson, Jon Timmis, Lisa Scott, and Mark Coles. 2008. Simulating biology: towards understanding what the simulation shows. In *Workshop on Complex Systems Modelling and Simulation.* Luniver press, 93–123.

[8] Yuan-Shun Dai, Min Xie, Quan Long, and Szu-Hui Ng. 2007. Uncertainty analysis in software reliability modeling by bayesian analysis with maximum-entropy

principle. *Transactions on Software Engineering* 33, 11 (2007), 781–795.

[9] Naeem Esfahani and Sam Malek. 2013. Uncertainty in self-adaptive software systems. In *Software Engineering for Self-Adaptive Systems II*. Springer, 214–238.

[10] Stephanie Evans, Kieran Alden, Lourdes Cucurull-Sanchez, Christopher Larminie, Mark C. Coles, Marika C. Kullberg, and Jon Timmis. 2017. ASPASIA: A toolkit for evaluating the effects of biological interventions on SBML model behaviour. *PLoS Computational Biology* 13, 2 (2017), e1005351. https://doi.org/10.1371/journal.pcbi.1005351

[11] Philip Garnett, Arno Steinacher, Susan Stepney, Richard Clayton, and Ottoline Leyser. 2010. Computer simulation: the imaginary friend of auxin transport biology. *Bioessays* 32, 9 (2010), 828–835. https://doi.org/10.1002/bies.200900185

[12] 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. De Angelis. 2006. A standard protocol for describing individual-based and agent-based models. *Ecological Modelling* 198, 1–2 (2006), 115–126. https://doi.org/10.1016/j.ecolmodel.2006.04.023

[13] Volker Grimm, Steven F. Railsback, Christian E. Vincenot, Uta Berger, Cara Gallagher, Donald L DeAngelis, Bruce Edmonds, Jiaqi Ge, Jarl Giske, Jürgen Groeneveld, Alice S. A. Johnston, Alexander Milles, Jacob Nabe-Nielsen, J. Gareth Polhill, Viktoriia Radchuk, Marie-Sophie Rohwäder, Richard A. Stillman, Jan C. Thiele, and Daniel Ayllón. 2020. The ODD protocol for describing agent-based and other simulation models: A second update to improve clarity, replication, and structural realism. *Journal of Artificial Societies and Social Simulation* 23, 2 (2020), 1 – 7.

[14] Anne-Laure Jousselme, Patrick Maupin, and Éloi Bossé. 2003. Uncertainty in a situation analysis perspective. In *6th International Conference of Information Fusion*. IEEE, 1207 – 1214. https://doi.org/10.1109/ICIF.2003.177375

[15] Tim Kelly. 2004. A systematic approach to safety case management. *SAE transactions* 113 (2004), 257–266.

[16] Marc C. Kennedy and Anthony O'Hagan. 2001. Bayesian calibration of computer models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 63, 3 (2001), 425–464.

[17] Frank H Knight. 2012. *Risk, uncertainty and profit*. Courier Corporation.

[18] Nancy G. Leveson and Kathryn A. Weiss. 2004. Making Embedded Software Reuse Practical and Safe. In *12th ACM International Symposium on Foundations of Software Engineering*. Association for Computing Machinery, 171 – 178. https://doi.org/10.1145/1029894.1029897

[19] Andrew W. Lo and Mark T. Mueller. 2010. Warning: Physics Envy May be Hazardous to Your Wealth! *Journal of Investment Management* 8, 2 (2010), 13–63. https://doi.org/10.2139/ssrn.1563882

[20] Daniel Moyo. 2014. *Investigating The Dynamics of Hepatic Inflammation Through Simulation*. Ph.D. Dissertation. University of York. http://etheses.whiterose.ac.uk/6967/

[21] William L Oberkampf, Sharon M DeLand, Brian M Rutherford, Kathleen V Diegert, and Kenneth F Alvin. 2002. Error and uncertainty in modeling and simulation. *Reliability Engineering & System Safety* 75, 3 (2002), 333–357.

[22] M. Padulo and Marin D. Guenov. 2012. A methodological perspective on Computational Engineering Design under uncertainty. In *European Congress on Computational Methods in Applied Sciences and Engineering*. T.U. Wien, 7509 – 7528. https://eccomas2012.conf.tuwien.ac.at/

[23] Hyeong-Uk Park. 2014. *Integrated uncertainty techniques in aircraft derivative design optimization*. Ph.D. Dissertation. Ryerson University, Canada.

[24] Edoardo Patelli, H. Murat Panayirci, Matteo Broggi, Barbara Goller, Pierre Beaurepaire, Helmut J. Pradlwarter, and Gerhart I. Schüller. 2012. General purpose software for efficient uncertainty management of large finite element models. *Finite Elements in Analysis and Design* 51 (2012), 31–48. https://doi.org/10.1016/j.finel.2011.11.003

[25] Fiona A. C. Polack. 2012. Choosing and adapting design notations in the principled development of complex systems simulations for research. In *Modelling of the Physical World Workshop*. ACM Digitial Library, 1–6. https://doi.org/10.1145/2491617.2491623

[26] Fiona A. C. Polack. 2015. Filling gaps in Simulation of Complex systems: the Background and Motivation for CoSMoS. *Natural Computing* 14, 1 (2015), 49–62. https://doi.org/10.1007/s11047-014-9462-5

[27] Fiona A. C. Polack, Paul S. Andrews, Theodor Ghetiu, Mark Read, Susan Stepney, Jon Timmis, and Adam T. Sampson. 2010. Reflections on the Simulation of Complex Systems for Science. In *ICECCS*. IEEE Press, 276–285.

[28] Fiona A. C. Polack, Paul S. Andrews, and Adam T. Sampson. 2009. The engineering of concurrent simulations of complex systems. In *CEC*. IEEE Press, 217–224.

[29] Fiona A. C. Polack and Steffen Zschaler. 2020. Report on Workshop: Planning the Future of Agent Simulation. *Artificial Life* (2020).

[30] Mark Read, Paul S. Andrews, Jon Timmis, and Vipin Kumar. 2012. Techniques for Grounding Agent-Based Simulations in the Real Domain: a case study in Experimental Autoimmune Encephalomyelitis. *Mathematical and Computer Modelling of Dynamical Systems* 18, 1 (2012), 67–86.

[31] Mark Read, Paul S. Andrews, Jon Timmis, and Vipin Kumar. 2012. Techniques for grounding agent-based simulations in the real domain: a case study in experimental autoimmune encephalomyelitis. *Mathematical and Computer Modelling of Dynamical Systems* 18, 1 (2012), 67 – 86. https://doi.org/10.1080/13873954.2011.601419

[32] Mark N. Read. 2011. *Statistical and Modelling Techniques to Build Confidence in the Investigation of Immunology through Agent-Based Simulation*. Ph.D. Dissertation. University of York.

[33] Beat Schäffer, Stefan Plüss, and Georg Thomann. 2014. Estimating the model-specific uncertainty of aircraft noise calculations. *Applied Acoustics* 84 (2014), 58–72.

[34] Christian D. Schunn. 2010. From Uncertainly Exact to Certainly Vague: Epistemic Uncertainty and Approximation in Science and Engineering Problem Solving. In *Psychology of Learning and Motivation*. Vol. 53. Academic Press, 227 – 252. https://doi.org/10.1016/S0079-7421(10)53006-8

[35] SCSC-141B 2018. *GSN Community Standard. Version 2*. Standard. SCSC Assurance Case Working Group. http://scsc.uk/SCSC-141B

[36] Susan Stepney. 2013. CoSMoS simulation experiment reproducibility and the ODD protocol. In *CoSMoS workshop*. Luniver Press, 93 – 107.

[37] Susan Stepney and Fiona A.C. Polack. 2018. *Engineering Simulations as Scientific Instruments: A Pattern Language*. springer.

[38] Mark D. Wilkinson, Michel Dumontier, IJsbrand Jan Aalbersberg, Gabrielle Appleton, Myles Axton, Arie Baak, Niklas Blomberg, Jan-Willem Boiten, Luiz Bonino da Silva Santos, Philip E. Bourne, Jildau Bouwman, Anthony J. Brookes, Tim Clark, Mercè Crosas, Ingrid Dillo, Olivier Dumon, Scott Edmunds, Chris T. Evelo, Richard Finkers, Alejandra Gonzalez-Beltran, Alasdair J. G. Gray, Paul Groth, Carole Goble, Jeffrey S. Grethe, Jaap Heringa, Peter A. C. 't Hoen, Rob Hooft, Tobias Kuhn, Ruben Kok, Joost Kok, Scott J. Lusher, Maryann E. Martone, Albert Mons, Abel L. Packer, Bengt Persson, Philippe Rocca-Serra, Marco Roos, Rene van Schaik, Susanna-Assunta Sansone, Erik Schultes, Thierry Sengstag, Ted Slater, George Strawn, Morris A. Swertz, Mark Thompson, Johan van der Lei, Erik van Mulligen, Jan Velterop, Andra Waagmeester, Peter Wittenburg, Katherine Wolstencroft, Jun Zhao, and Barend Mons. 2016. The FAIR Guiding Principles for scientific data management and stewardship. *Scientific Data* 3, 1 (2016), 160018. https://doi.org/10.1038/sdata.2016.18

[39] Richard A. Williams, Richard Greaves, Mark Read, Jon Timmis, Paul S. Andrews, and Vipin Kumar. 2013. In silico investigation into dendritic cell regulation of CD8Treg mediated killing of Th1 cells in murine experimental autoimmune encephalomyelitis. *BMC Bioinformatics* 14 (2013), S6–S9. https://doi.org/10.1186/1471-2105-14-S6-S9

[40] Man Zhang, Bran Selic, Shaukat Ali, Tao Yue, Oscar Okariz, and Roland Norgren. 2016. Understanding uncertainty in cyber-physical systems: a conceptual model. In *European conference on modelling foundations and applications*. Springer, 247–264.

[41] Chen Zhehan, Du Fuzhou, and Tang Xiaoqing. 2013. Research on uncertainty in measurement assisted alignment in aircraft assembly. *Chinese Journal of Aeronautics* 26, 6 (2013), 1568–1576.