Methodology

Simulation

Simulation: Transactions of the Society for Modeling and Simulation International
1–17

DOI: 10.1177/00375497251383912
journals.sagepub.com/home/sim
Sage SCS

On simulation reuse in healthcare applications

Steffen Zschaler[1], Navonil Mustafee[2], Alison Harper[2], Thomas Monks[3], Bhakti Stephan Onggo[4], Christine S. M. Currie[4] and Fiona Polack[5]

Abstract

Simulation remains a promising technology in healthcare operations research and process optimisation. However, while there have been many research projects applying simulation in this context, the level of sustained uptake in healthcare practice has been lower. We conjecture that an important reason for this is the time, cost and complexity of developing simulation models. Therefore, being able to reuse models would be key to improving uptake of simulation in healthcare. Conventional practice is that simulation models are developed *from scratch* for every new problem. In this paper, we review current strategies for model reuse in the healthcare context, aiming to identify complementary techniques for model reuse available to healthcare modellers and managers. Specifically, we identify three different modes of model reuse – forming a triadic framework – each prioritising a different aspect: FAIR and open-science aspects of model reuse; reusable conceptual simulation domains through modelling languages and transformations; and black-box model(-component) reuse including distributed simulation. We show how these three perspectives complement and enhance each other. We believe that developing concrete mechanisms and tools for leveraging the relationships between the three different modes of model reuse will be key to increasing the uptake of simulation modelling in healthcare practice.

Keywords

Simulation, healthcare, reuse, open science, model transformation, distributed simulation

1. Introduction

Computer-based modelling and simulation (M&S) is an important tool for supporting healthcare operations, strategic planning and policy analysis.[1–3]

Despite its demonstrated potential, simulation remains underused in healthcare decision-making.[2,4] One of the long-standing challenges is the high cost, in both time and expertise, of developing new models for each application. Model reuse is the practice of reapplying existing models, components, or design knowledge.[5] It has been proposed as a way to reduce duplication and promote uptake,[6–9] yet in practice, reuse remains rare, largely due to issues around model compatibility, credibility, discoverability and accessibility.[6,7,9,10] The availability of reusable simulation models in healthcare could significantly improve the use *in practice* of simulation for healthcare process improvements. Centralised healthcare systems, such as the UK National Health Service (NHS), can benefit from economies of modelling and simulation[11] that are brought about by increasing the use of modelling and simulation in such systems through adopting model reusability principles.[4,12]

Previous research has represented simulation reuse as hierarchical, delineating various degrees of reuse ranging from complete model reuse down to minimal, function-level reuse. Robinson et al.[8] described a simple reuse spectrum: (1) full model reuse; (2) component reuse; (3) function reuse; and (4) code scavenging, emphasising the trade-off between ease of use and complexity. Similarly, Hussain et al.[6] propose a six-level reuse hierarchy, illustrating that while full model reuse offers maximum efficiency, it is also the most challenging to achieve due to constraints such as model compatibility and credibility. Balci et al.[10] challenged this thinking, describing

[1] King's College London, UK
[2] University of Exeter Business School, UK
[3] University of Exeter Medical School, UK
[4] University of Southampton, UK
[5] University of Hull, UK

Corresponding author:
Steffen Zschaler, Department of Informatics, King's College London, London WC2B 4BG, UK.
Email: szschaler@acm.org

achievability increasing in the opposite direction, with reuse at the programming/class/function levels made difficult by the many options in programming languages and operating systems, while at the opposite end of the hierarchy, network-centric applications which use M&S components distributed over a network are supported by standards, such that a model built in compliance with a distributed simulation standard, such as the IEEE 1516 High-Level Architecture (HLA),[13] can be reused by other models interconnected through the HLA protocol.

We extend this argument by rethinking earlier conceptualisations. The rise of open-source simulation libraries, modular and hybrid modelling techniques, and modern software engineering practices has changed the way models are built and shared. More broadly, the adoption of open science principles and digital research infrastructures has improved the findability, accessibility and reproducibility of simulation artefacts. These developments open up new possibilities for reuse that go beyond the scenarios described by earlier frameworks. We argue that the field now needs a conceptual update that recognises not only *what* is reused (e.g. models, code, ideas) but also *how* reuse is made possible in practice. There is an evident gap in understanding the practical methods to effectively operationalise simulation reuse, moving beyond theoretical concepts toward implementable solutions. This paper aims to address this gap by reviewing existing literature and proposing a practical, integrated framework to promote simulation reuse in healthcare contexts. We introduce a triadic framework encompassing open science approaches, modelling languages and translations, and black-box components. We show how these provide complementary answers to the challenges in simulation reuse and discuss potential synergies between these approaches. Our aim is to ignite discussion on the best practices and future directions for enhancing reusability in healthcare simulation modelling.

The remainder of this paper is structured as follows: In Section 2, we briefly summarise existing conceptualisations of simulation reuse, and review the simulation reuse research. Drawing on the insights from the literature, in Section 3, we give an overview of the proposed triadic framework that synthesises different approaches for how to reuse simulations in healthcare, before discussing their synergies in Section 4. We conclude the paper in Section 5 with some proposals for a future research roadmap.

2. Review of simulation reuse

Reusable simulation models or simulation model elements are often conceptualised as either generic or generalisable. Boyle et al.[14] differentiated between these two types, defining generic models as those developed from theoretical principles without reliance on specific empirical data,[15] intended primarily for conceptual exploration and general problem-solving. In contrast, generalisable models can be adapted using site-specific data, allowing for broader applicability and more practical reuse across different contexts. This flexibility aims to balance the initial development effort with subsequent reuse benefits.

While this differentiation between generic and generalisable models offers a useful way to characterise what may be reused, it does not fully explain how such reuse occurs in practice or what conditions support it. To better understand the current landscape of reuse-enabling practices and tools, we conducted a search of the recent simulation literature to identify empirical examples of model reuse. We conducted a structured literature search using Web of Science (search terms: 'model reuse' AND (DES OR 'discrete event' OR 'discrete-event' OR ABM OR 'agent-based' OR ABS OR 'agent based')), which returned 43 results. Following full paper review, 18 papers were excluded due to lack of relevance to simulation reuse, being outside of applied contexts, or using simulation methods that are outside of Operational Research, such as CAD or 3D geometric modelling. Of the 25 remaining papers, we examined:

- Techniques for simulation/model reuse;
- The mode of simulation reuse described;
- Challenges for simulation reuse;
- Specific application domains.

The review identified a variety of reuse strategies that go beyond standard typologies. These include generalisable model frameworks, experimental reuse of pre-built models and modular cognitive architectures. Two studies proposed distributed simulation frameworks that support model reuse by enabling federation of existing models.[16,17] Several studies emphasised the role of modelling languages and translations in enabling reuse. These include XML-based description files for structured model generation,[18] model transformation between DEVS and SMP2 for interoperability,[19] and translation across formalisms (e.g. SDL to DEVS) for cross-verification.[20] Others propose reusable frameworks for PDES,[21] abstract simulators[22] and executable enterprise architectures using standardised representations.[23] Together, these show the growing emphasis on formalism-bridging, modularisation and abstraction as enablers of simulation reuse.

Two healthcare-focused studies support the role of open science in enabling reuse. Penn et al.[24] described the reuse and redevelopment of a generic hospital ward model, highlighting trade-offs between flexibility and simplicity. Harper et al.[25] explored how simulation models developed in Python could be made more accessible to non-technical users through deployment strategies. Both studies identified barriers such as complexity, lack of standardisation and limited accessibility, reinforcing the need for reusable models to be openly shared, documented and usable without specialist skills.

Across diverse domains, several studies offered practical insights into simulation reuse, for example Monks et al.[26] experimentally explored trade-offs between model reuse and client learning, while Petersen et al.[27] introduced modular components for biomedical simulations. Technical strategies include component-based modelling,[28] formal pattern reuse via Petri Nets,[29] and interoperable DEVS components.[30] Some of the challenges of reuse of agent simulations of complex immunological behaviours, were examined with a demonstrably fit-for-purpose scientific simulation.[31,32] Other work focused on cognitive architecture standardisation,[33] integrated modelling for nuclear policy,[34] or hybrid frameworks that blend simulation with contextual organisational insights.[9] Challenges such as model familiarity, validation, documentation and stakeholder trust recurred frequently.[35,36]

Collectively, these studies highlight reuse as technically feasible but often limited by practical, organisational and epistemological barriers. Despite an established conceptual understanding of reuse, practical mechanisms for operationalising simulation reuse remain underdeveloped. There is a clear need for methodologies that effectively address these identified challenges, providing frameworks that support not only theoretical conceptualisation but also practical implementation. Drawing on these insights, we propose a triadic framework that synthesises open science principles, modelling languages and black-box components methods. These have been identified across the literature as recurring enablers and complementary strategies to support practical and effective simulation reuse in healthcare.

3. Three different approaches to simulation reuse

In this section, we briefly describe three specific approaches to reusing simulation models. In the next section, we will discuss how these approaches inter-relate.

3.1. Open science tools to enhance the reusability of computer simulation models

The general aim of open science is to make research widely accessible, promoting the transparency, credibility and reuse of computational artefacts.[37,38] In healthcare, code sharing and other open practices are seen as a way of providing a transparent, unambiguous record of analysis,[39] but also maximising limited resources and increasing the potential for impact.[40,41] Barriers to sharing healthcare simulation models stem primarily from time, skills and infrastructure required to prepare code for reuse.[42] Broader surveys of healthcare research practice echo this, indicating that although researchers are willing to share code, actual reuse is constrained by documentation quality, lack of persistent repositories and limited incentives.[43,44] While code reuse is most frequent among colleagues and within trusted networks, wider adoption is inhibited by these structural challenges. It follows that open science practice enables model reuse in healthcare.[45] In this context, we discuss four sets of resources available to the simulation modeller to enhance the open science credentials and reusability of their models. These approaches are mainly, but not exclusively, applicable to models built using free and open-source software (FOSS) such as Python or R.

3.1.1. Discoverability and preservation. Making a model available, and – as importantly – discoverable, makes it more likely to be used in new applications. Across healthcare organisations, we see similar processes, problems and data structures.[9,14] Model reuse helps to raise the profile of the work and increase both its academic and practical impact. A contemporary way to do this is archiving through a digital open science repository. This approach also preserves the model, as archives such as Figshare, Zenodo and the Open Science Framework provide guarantees on persistence. Benefits for modellers include citable models using a Digital Object Identifier (DOI),[46] and meta-data that increases the likelihood that a model is discovered.[40]

3.1.2. Frameworks. Guidance, software and templates exist to support the structuring of simulation models and accompanying artefacts to enable open modelling in healthcare. For healthcare discrete-event simulation modelling, STARS (Sharing Tools and Artefacts for Reusable Simulations)[45] aims to overcome barriers to sharing models in healthcare by providing essential (such as open licensing) and optional (such as enhanced documentation) components to strengthen the reusability of computer simulation models written in free and open source software. The authors provide three applied simulation examples in Python. In an area related to simulation, Smith et al.[47] provide packaging guidance for Health Technology Assessment models in R. More generally, van Lissa et al.[48] developed WORCS (Workflow for Open Reproducible Code in Science), a structured framework to support an open science workflow in R. Each of these support the necessary activities required to develop models that are findable, available, executable, modifiable and reusable.

3.1.3. Model reporting guidelines. Reusing a simulation model will often include some adaptation of the current version to allow for differences in the system, making clear documentation vital. Various guidelines exist for describing simulation models such as STRESS (Strengthening the Reporting of Empirical Simulation

Studies)[49] which has checklists for discrete event simulation, agent based simulation and system dynamics; the popular ODD protocol[50] designed specifically for agent based simulation; and RAT-RS[51] which focuses on documenting how data is used in agent based simulation. Reporting guidelines document the essential steps, inputs and outputs of an M&S lifecycle, supporting reproducibility of results, interpretability of the model, and model reuse. They ensure that a manuscript can be understood, and a model and its results can be replicated and trusted. In healthcare research more widely, this is seen as a priority; for example, the Equator Network[52] is a repository of healthcare research reporting guidelines, aiming to improve the reliability and value of published health research literature through transparent and accurate reporting. The US Food and Drug Administration (FDA) and other medical certification authorities are moving towards defining clear standards for simulation validation and documentation.[53,54]

3.1.4. Evaluation. To confirm if a model is reusable, an option for authors is to submit the computer model for review. Such a review may award badges; for example, ACM journals operate a replicating computational results initiative, where a peer-reviewer will assess the reusability of the model, write a report, and then has the option to award an ACM RCR badge: 'Artefacts Evaluated – Reusable'. This provides confidence to potential re-users that the model is in a state where it could be adopted for their project. Models are only validated for their intended purposes. Other application domains such as defence have explored accreditation of models by an independent third party,[55] but to our knowledge, no such scheme exists within healthcare.

3.1.5. Open science examples. While not specific to simulation, HDR UK[40] provide a set of development principles and a GitHub repository of open-source code in healthcare research. They aim to work with academic partners to enable reuse in healthcare, where new development is seen as an exception, rather than a default. With a focus on discrete-event simulation, Monks et al.[45] provide several example applications implementing their STARS framework in healthcare. A tutorial demonstration model[56] implements an urgent care treatment centre simulation using all of the STARS optional components including enhanced documentation that is hosted online, and a web app to support usability by NHS users. An applied example of a generalisable, reusable model with a web app interface was developed for orthopaedic resource planning,[57,58] and for operating theatre efficiency,[59] both for NHS users. While still in the minority in M&S, more examples of M&S open science are starting to be seen in the academic literature.[42]

3.2. Reusability using modelling languages and translations

Various authors have proposed enabling simulation-model reuse by encoding knowledge about a particular domain in domain-specific modelling languages (DSMLs)[32] or by extending existing general modelling languages (e.g. BPMN)[60,61] from which simulations can be automatically generated,[32,62–64] or where simulations can interpret the models as they run.[65,66] Together, the DSML, extended general modelling language (EGML) and the program for generating the actual simulation encode a reusable simulation domain – new simulation models can easily be created by capturing their specifics in the DSML or EGML, and then generating a simulation. In other words, a simulation model can be represented using a modelling language (DSML or EGML) which is independent of the software that runs the simulation model. One main benefit of this approach is that we do not have to re-code the simulation model every time we need to change the simulation software. Thus, it will help the healthcare organisations to avoid vendor lock-in. Consequently, the DSML or EGML representation of the model can be viewed as an organisation long-term asset. Models captured in this way enable a form of *grey-box reuse*: we do not reuse the full model 'off the shelf', but instead there is a way for explicitly capturing what is specific about the present healthcare setting, while aspects that remain the same across settings can be reused as part of the modelling language and the simulation generators.

Below, we will discuss two approaches: (1) by extending an existing general modelling language, and (2) by creating a DSML.

3.2.1. Reusability using EGML: BPMN and UML. Several authors propose to extend existing general modelling languages for various applications including for healthcare.[60,61] Some general modelling languages such as the Unified Modelling Language (UML; https://www.uml.org/) and Business Process Model and Notation (BPMN; https://www.bpmn.org/) have been widely used in industry and provide specification and conformance standards. Therefore, it is not surprising that authors have proposed to extend UML and BPMN to support applications in healthcare.[63,67,68]

Although process modelling has increasingly been integrated into healthcare management process, BPMN and UML have not been widely used in healthcare applications,[61,69] especially for executable simulation models. This may be attributed to healthcare being a highly regulated sector as well as the complexity of healthcare processes. Pufahl et al.[61] identified nine challenges when modelling complex healthcare processes: BPMN cannot be used directly to model some of the complex healthcare processes. To address this, authors have proposed various

BPMN extensions that can be generalised into three groups: patient-related extensions, medical practice-related extensions and resource-related extensions.[61] Until these challenges are addressed, the work on executable simulation models will be limited to modelling simpler processes. Software tools such as Bizagi, Visual Paradigm and Simul8 can simulate simple models represented in BPMN with the help of standards such as BPSim (bpsim.org). Some examples of BPMN simulation in healthcare include emergency department[70] and orthopaedic outpatient clinic, surgery and neurological theatres, ageing and complex medicine (inpatients).[62]

Research into making UML executable is more mature than BPMN[71] partly because UML was invented earlier than BPMN. The latest UML specification was followed by the introduction of the Foundational UML Subset (fUML; https://www.omg.org/spec/FUML) and Action Language for Foundational UML (AlfUML; https://www.omg.org/spec/ALF) have enabled us to make a model represented using UML executable. Hence, it is possible to represent a simulation model using UML with fUML or AlfUML. However, there is little research that applies this to simulation in healthcare. Some work is using UML to represent healthcare models but only for communication purpose; the translation to software is manual. Examples include fractured neck of femur care process[72] and Emergency Department.[73]

The above examples show the benefits of using an existing general modelling language that has become a standard to enable simulation model reuse in healthcare. Using a standard language means that the model represented using the language will be understood by stakeholders from various backgrounds who are familiar with the standard. If some stakeholders are not familiar with the standard, it is arguably easier and cheaper to provide training over a period of time for a standard than for multiple different simulation software systems. As more people in an organisation learn the same standard modelling language, knowledge accumulates over time, making it easier to share knowledge within the organisation. This is especially true for a healthcare organisation whose main business is not developing simulation models. Furthermore, a widely used standard modelling language (e.g. UML and BPMN) has good resources, such as tutorials, documentation and discussion fora, available for public access. Further benefits include that any software tool that conforms to the standard will provide the same notations for the same simulation components, making communication and training on the simulation software easier; and that a standard typically has a longer shelf-life than a specific software simulation. Note that simulation software is typically updated repeatedly, and at some point backward compatibility is lost, forcing organisations to deal with legacy models that can no longer be run; using a standard modelling language however, new executables can be generated by providing a new or updated translator (or model transformation) to the required simulation software version. Hence, a model written in a standard modelling language can become a long-term asset for an organisation.

The main disadvantage of using an existing standard modelling language is that it is not specifically designed for simulation modelling. Therefore, the standard needs to be extended before it can be used for simulation modelling. Zarour et al.[60] provide a comprehensive review on how BPMN has been extended in various domains including healthcare. Another alternative is to develop a domain specific modelling language (DSML).

3.2.2. Reusability using DSML: ED modelling language. Godfrey et al.[74] describe a DSML particularly catering for the simulation of workflows in emergency departments. Their language is based on notations observed to be used by ED staff to make decisions about alternative workflows and communicate agreed workflows to all members of the ED – specifically, PowerPoint slides with a flow-chart–like notation called an Action Card. The language provided by Godfrey et al.[74] closely mimics the structure and visual notation of these action cards and provides additional concepts to capture information about patients, tests used and their properties, physical department layout, etc. From models expressed in this DSML, a simulation implementation in Repast is automatically generated with the ability to address questions such as the balance between cost, patient risk and length of stay for different action cards.

Godfrey et al.[74] report that developing new simulations in the ED space was quite straightforward once the domain specific language was created and that it was easy to adapt models for new contexts – as long as the new context could be captured within the vocabulary provided by the DSML. More generally, a DSML may not cover all aspects of a new context, but it is straightforward to extend the coverage of the DSML, such that significant parts of the existing models can be reused.[75]

DSMLs are not limited to capturing the vocabulary of the healthcare domain. Zschaler and Polack[32] illustrate other aspects of simulation models that can be expressed – for example, arguments of validity (fitness for purpose), or descriptions of the simulation experiments to be performed. This makes these aspects amenable to formal scrutiny and (semi-)automated execution and evaluation.

Validation is a key consideration in simulation reuse. If validity arguments are explicitly modelled (e.g. using a form of structured argument),[32,76] it becomes possible, in principle, to explicitly trace the impact of changes to the model to parts of the validity argument that need to be reconsidered or that may be reused – though the details of this remain to be studied in depth.

Reproducible execution of experiments is essential for simulation reuse. DSMLs for the description and automated execution of simulation experiments[77,78] make it easier for experiments to be performed consistently irrespective of who runs the experiment. Once an experiment-specification language has been established, it becomes possible to consider generating simulation experiments[79] from queries formulated in a suitable query-modelling language.[32,80]

The DSML approach differs from the previously presented ones in that it uses the language of the domain (e.g. vocabulary familiar to ED staff) rather than the generic concepts of the modelling language (UML, BPMN). This has clear benefits in that clinical staff are directly able to interact with the models without requiring training in BPMN or similar standard notations, which are typically not used directly by clinical staff. The DSML is also directly matched to the needs of the healthcare domain and does not need further adaptation as is the case when using general modelling languages. As such, using a DSML can increase reusability of a simulation model as it can be maintained and adapted to changes in the real-world processes directly by clinical staff. Note a DSML is often created on top of a standard modelling language, giving access to the tooling and practices where appropriate. This means, for instance, that existing BPMN models of relevant processes can be imported into a DSML model.

3.3. Reusing simulation models as black-box components

Black box reuse is common in software engineering,[81] and we have also found several papers in the literature that employ black-box reuse in simulation modelling.[16,17,26,28–30] These approaches typically treat entire simulation models as a component, which is reused without inspection or adaptation of its implementation details. Black-box model reuse often involves the composition of multiple model components.

The multiple model components may be owned by different departments or organisations. Therefore, more often than not, the model components are distributed over a network. Since the 1970s, the Parallel and Distributed Simulation (PADS) community has developed novel approaches to *distributed simulation*, which is the distributed execution of simulation over multiple computers linked over a geographical network, and *parallel simulation*, the use of parallel computing for faster execution of a single simulation over multiple processors.[82,83] These distributed simulation standards, techniques and best practices, thus, offer excellent tools for black-box reuse and linking of healthcare models. However, specialised software must exist that coordinates the execution of the simulation time between existing models.[82] In PADS terminology, the software is referred to as distributed simulation middleware. The IEEE 1516 Standard for M&S High-Level Architecture (HLA), which has become the de-facto standard for distributed simulation, refers to individual simulation systems as federates, which, together with the Run-time Infrastructure (RTI, the middleware that enables message exchange and simulation time synchronisation among federates) and the Federated Object Model (the model defines object and interaction classes), form an HLA federation.[13] The IEEE 1516 standard has been used for example to implement a blood supply chain network consisting of one central blood processing, testing and issuing (PTI) centre and multiple hospitals, with each submodel developed in a commercially available simulation package (CSP) – Simul8 – and implemented as an individual HLA federate, which together, through use of the HLA-RTI 1.5NG middleware, act as a distributed simulation federation.[84] The study by Anagnostou and Taylor[16] developed a distributed simulation of an emergency medical service based in London. The model was a hybrid agent-based and discrete-event simulation; the HLA RTI was the Portico v2.0 RTI implementation of IEEE 1516-2000.[16]

Healthcare simulation models have traditionally been developed using CSPs like Simul8, Flexim and Arena. Implementation of a distributed simulation using CSPs that do not readily support distributed execution of models can be a challenge.[84] Thus, the question is why distributed simulation is considered a feasible approach for model reusability. To answer this, we first refer to some general literature on cost and time for model development, model validation, and so on, and then argue the opportunities that distributed simulation presents in healthcare M&S.

First, following the life cycle of a simulation study, we can identify various cost elements that a project is likely to incur. For example, an M&S expert/consultant may need to be hired; in the conceptual modelling phase, the stakeholders' time has a cost element; model implementation may require the purchase of commercial off-the-shelf software or support; there are likely to be costs towards collecting data, especially if this is primary data. Large and complex models may need several years to reach optimal point in terms of artefact development and would have benefited from several rounds of verification and validation, and step-wise introduction of organisational knowledge, all of which enhances different aspects of trust between the stakeholder, modeller and the model.[85] Distributed simulation enables the reuse of such black-box models; these are the trusted artefacts, developed over several years in individual healthcare organisations, and the likelihood of trust being eroded due to the model being a part of a larger distributed model is low. However, it is acknowledged that the distributed simulation as a whole (simulation federation), which is composed of sub-models, will need additional interventions for the broader group of stakeholders who are now associated with the larger modelling initiative; this is especially the case for modelling artefacts that are from other organisations. To elaborate on this, we take the example of neonatal treatment units (NTUs) in hospitals that are served by one human milk bank (HMB); the former models the processes around ordering and inventory management of perishable milk stocks in NTUs, while the HMB models splitting strategies for the batches of donated milk and allocation of stocks to NTUs.[86] A distributed simulation model of such a milk banking network can thus consist of several NTU models that are trusted by the hospitals and linked to the HMB model. However, for NTUs' stakeholders to trust the HMB model, and vice versa, may take additional time and efforts.

Next, we frame the utility of the distributed simulation approach of whole model (black-box) reuse with reference to the literature on model validation. Sargent[87] discussed the relationship between model confidence and the cost of validation, the amount of time needed for validation and, finally, the value the stakeholder would derive from the model. To clarify this relationship the author used a figure that comprised three axes: cost/time, model confidence and value of the model to the stakeholder. It illustrated that higher model confidence would incur higher costs towards model validation, but the marginal gain of value to costs would be steadily increasing; however, after a certain threshold of confidence is reached, the marginal gain in stakeholders' value attribution of the model compared to validation costs will experience a steady decline.[87] Thus, it is arguable that for a model that has incurred significant investments in time and costs towards model validation, whole-model reuse using distributed simulation standards and middleware presents an additional layer of opportunity. This is shown by extending Sargent's conceptualisation, adding an additional axis to depict the increasing relevance of model reuse using distributed simulation (Figure 1). We take the intersection point between value and cost/time curve as an example. At this point, the value of the model to the stakeholders is high since there has been an investment of both time and costs towards model validation, which has increased model confidence. The relevance of distributed simulation for model reuse is also high at this point.

We take the example of healthcare supply chain simulation to discuss opportunities for DES model reuse. A healthcare supply chain consists of several tiers, such as manufacturers of pharmaceutical products, Tier-1 suppliers such as Active Pharmaceutical Ingredients (API) suppliers, drug distributors with warehouse and logistics functions, and retailers like hospital pharmacies and community pharmacies. A drug manufacturer *(M)* may routinely employ a DES model for planning the production of pharmaceutical products; similarly, a distributor *(D)* may use a DES model to simulate the flow of various drugs through its network of warehouses. *M* may be considering the services of a new distributor through a tendering process. *D* may be a contender for the tender. Since both *M* and *D* employ DES models for the planning of their day-to-day operations, the use of simulation in the tendering process to model the future pharmaceutical production planning, inventory and distribution functions can be considered. This may allow opportunities to experiment with supply chain shocks and determine the resilience of the existing systems, and how the processes in the two organisations may react if they are joined through a successful tendering outcome.

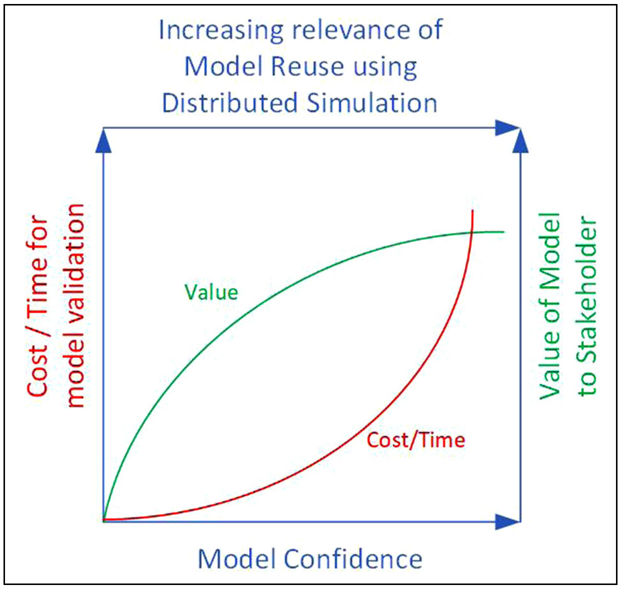
Increasing relevance of Model Reuse using Distributed Simulation
Cost / Time for model validation
Value
Cost/Time
Value of Model to Stakeholder
Model Confidence

Figure 1. Increase of stakeholder trust in models, frequently brought about through the investment of money and time towards model validation – adapted from Sargent[87]– increases the opportunity of reusing trusted models using distributed simulation approaches.

A DES distributed simulation federation composed of the individual DES models in use at *M* and *D*, respectively, could enable experimentation of such future *M–D* supply chain operations. Reusing such healthcare DES models through distributed simulation would arguably be the preferred option compared to developing a new supply chain simulation that integrates the processes for both M and D. The distributed simulation approach treats each model as a black box; when compared to conventional supply chain simulation, its distributed counterpart enables model reuse, data privacy/information hiding, data integrity and opportunities for speeding up the execution of large models.[88] A new model will also take time and money for model validation and to gain the trust of the stakeholders.[87] Finally, distributed simulation enables model reuse, and thus, trusted artefacts are used.

4. Discussion: the triad of reusability

Existing studies on model reuse[6,8] have focused on the costs and benefits of reuse in relation to *what* part of a model is being reused, ranging from code scavenging (reusing individual pieces of model code in a copy-and-paste fashion) to full model reuse.

In our choice of types of model reuse, we have focused on *how* models or model components are being reused. The spectrum here ranges from white-box reuse in the spirit of FOSS to black-box reuse of model components, with an intermediary form of grey-box reuse with modelling languages and transformations. We also found some examples of generic models – models that aim to cover many possible healthcare contexts with minimal parametrisation need. These approaches conceptually sit between grey-box reuse with modelling languages and transformations, and black-box reuse with model components: whether they are closer to one or the other depends on the complexity of the parametrisation available (simple numeric parameters in black-box reuse and potentially complex model structures in grey-box reuse). This leads to a triadic model of model reuse *modes*, which are summarised in Figure 2:

1. The first element of our triad of model reuse focuses on *model reuse using open science methods*. The popularity of FOSS packages for simulation like SimPy, Salabim, AgentPy and Mesa (all in Python) has also meant that the current generation of modellers is more code savvy. This has meant that developers of models and simulations have become less dependent on CSPs. Although CSPs continue to flourish, there is an increasing recognition that an extension of the current state-of-the-art in M&S, through, for example, the development of hybrid simulation models[89] that integrate approaches such as using discrete-event simulation (DES) and machine learning, necessitates a certain degree of flexibility in the use of software. This flexibility is more readily provided using FOSS software and libraries. Developing complex DES models using CSPs requires expertise in either the CSP-specific programming language – for example, Visual Logic for Simul8 – or a general-purpose programming language (e.g. C + + for FlexSim). However, such coding generally tends to be accomplished within the parameters of the overarching CSP, to the extent that interrogating the next event in the event list (an event list is a data structure that consists of the time-stamped ordering of future events which is maintained by a core element of the simulation software which is referred to as the simulation executive or the simulation engine) from an external program could not generally be accomplished without vendor support. In contrast, FOSS DES software and libraries provide access to simulation executives through pre-classes and function calls, which could be more readily integrated into the code base. Thus, with the advent of FOSS and its increasing use in both academia and practice, we recognise that the simulation modeller of today is arguably more adept at developing complete simulations in, for example, Python and R with DES or ABS libraries, which, we argue, significantly extends the piecemeal use of coding that is necessary when introducing complex logic in standard models developed using CSPs. Thus, full-model reuse, as opposed to code scavenging, may no longer be the most challenging approach, different from the observations of Robinson et al.[8]
2. The second element of our triad of model reusability is *black-box reuse of model components*. Component-based simulation is a modular approach to building computer models, where individual components are developed separately, with the intention of model reuse, and interact through well-defined interfaces. When the internal workings of these components are either unknown or irrelevant to the simulation study, this approach is known as black-box component model reuse. A common example is found in simulation federations, where multiple self-contained sub-models operate together within a larger system. In such cases, distributed simulation enables black-box integration by managing data exchange between components through exposed interfaces, without requiring access to their internal logic.

In the literature, we find examples of standards on the use of distributed simulation methods and approaches with CSPs, in particular, the *SISO CSPI PDG Standard for COTS Simulation Package Interoperability Reference Models*[90,91] and the use of the IEEE 1516 HLA standard.[13] However, the use of distributed simulation in the industry has only had limited success. Our discussion on CSPs in relation to FOSS and open science is relevant here. Integration of CSPs with external applications may require bridging software for message passing. For example, Mustafee and Taylor[92] grid-enabled CSP Witness with their *WinGrid* desktop grid computing middleware through the use of software adapters that relayed messages (and callbacks). Similarly, developing a distributed simulation with CSPs usually necessitates the development of ancillary artefacts that couple the CSP implementation of a

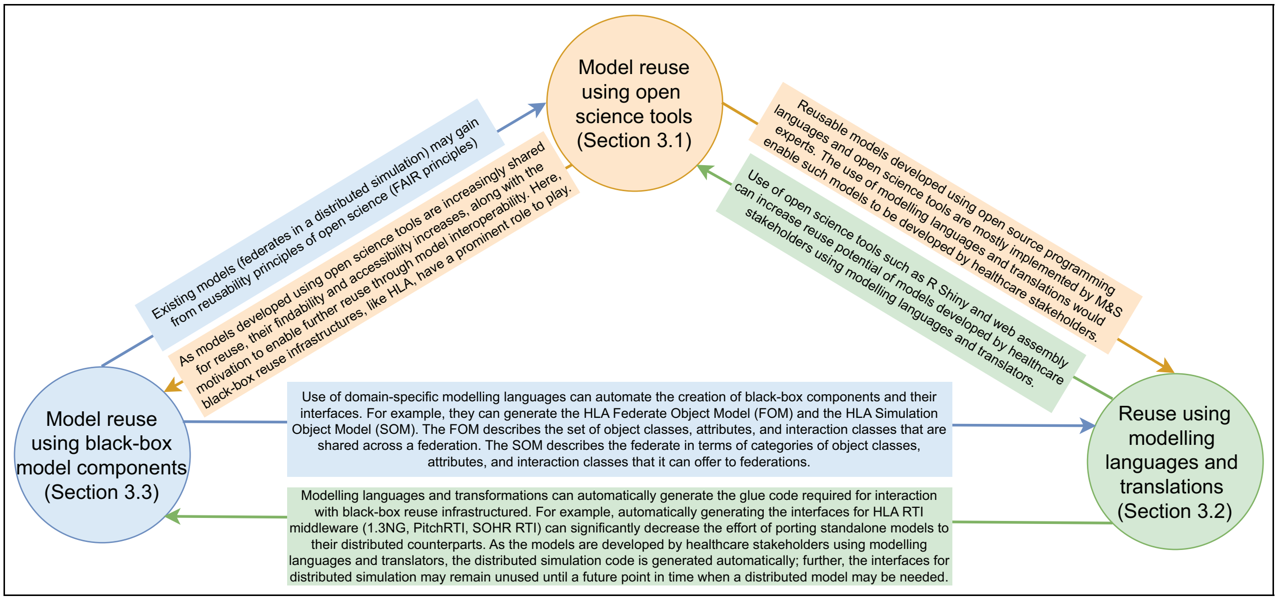
Model reuse using open science tools (Section 3.1)
Model reuse using black-box model components (Section 3.3)
Reuse using modelling languages and translations (Section 3.2)
Existing models (federates in a distributed simulation) may gain from reusability principles of open science (FAIR principles)
As models developed using open science tools are increasingly shared for reuse, their findability and accessibility increases, along with the motivation to enable further reuse through model interoperability. Here, black-box reuse infrastructures, like HLA, have a prominent role to play.
Reusable models developed using open source programming languages and open science tools are mostly implemented by M&S experts. The use of modelling languages and translations would enable such models to be developed by healthcare stakeholders.
Use of open science tools such as R Shiny and web assembly can increase reuse potential of models developed by healthcare stakeholders using modelling languages and translators.
Use of domain-specific modelling languages can automate the creation of black-box components and their interfaces. For example, they can generate the HLA Federate Object Model (FOM) and the HLA Simulation Object Model (SOM). The FOM describes the set of object classes, attributes, and interaction classes that are shared across a federation. The SOM describes the federate in terms of categories of object classes, attributes, and interaction classes that it can offer to federations.
Modelling languages and transformations can automatically generate the glue code required for interaction with black-box reuse infrastructured. For example, automatically generating the interfaces for HLA RTI middleware (1.3NG, PitchRTI, SOHR RTI) can significantly decrease the effort of porting standalone models to their distributed counterparts. As the models are developed by healthcare stakeholders using modelling languages and translators, the distributed simulation code is generated automatically; further, the interfaces for distributed simulation may remain unused until a future point in time when a distributed model may be needed.

Figure 2. The triadic relationship between three modes of simulation re-use.

DES model with the distributed simulation middleware. A middleware based on a standard such as the IEEE 1516 will have open interfaces for communication which are well documented. However, CSPs that are generally used to develop healthcare models are not architecturally designed to support distributed simulation or other forms of black-box model reuse; further, relevant documentation and APIs may not exist that readily allow such reuse. For example, the ancillary software that serves as a bridge between the CSP and the distributed simulation middleware needs to be manually coded. The success of distributed simulation in space and defence, for example, could be partly attributed to the programming of the simulation model (rather than the use of CSPs), making the integration with the distributed simulation software easier. This is not yet prevalent in healthcare. The use of FOSS presents us this opportunity!

3. The final element of our triad supports *model reuse through modelling languages and translations*. Here, we need to differentiate the roles of *software developers* (programmers, CSP vendors and researchers), *model developers* (M&S experts, who may be practitioners or researchers) and *model users* (stakeholders). Importantly, MLTs enable healthcare stakeholders to be directly involved in model building and parametrisation. To enable this, the developers of the MLTs may organise workshops and guided learning sessions for stakeholders, who then learn the modelling language and gain the skill of developing models based on taxonomies they associate with (which is again dependent on domain) rather than M&S constructs. This is typically not possible if either CSPs or FOSS tools are used for model development. For more complex problems, stakeholders and researchers can co-develop the models in a participatory modelling setting.[62] MLTs enable better communication between stakeholders and researchers which is critical for model co-development or participatory modelling. Stakeholders may be able to input parameters for models developed using CSP or FOSS if, for example, the frontend provides relevant functionality. Importantly, MLTs shift implementation effort to the core software artefacts, which need to be developed for particular healthcare domains. This is similar to the skill set in programming necessary for the deployment of distributed simulation techniques and FOSS for furthering the reuse of simulation models.

Table 1 provides a high-level comparison of the three reuse modes according to several categories. The first three columns focus on different stages of model development and reuse, capturing who undertakes the work and what they are developing. Here, it is noticeable that black-box reuse is primarily the work of M&S experts, including the parametrisation stage. In contrast, open-science-based model reuse, and even more so MLT-based reuse allows a shift away from M&S experts and towards healthcare stakeholders for model building and parametrisation. The key differentiator here is the complexity of what healthcare stakeholders can contribute directly: in MLT, they can potentially contribute full models, whereas in open-science

Table 1. Properties of the three reuse modes.

Reuse mode	Core software artefact (what and who)	Model building (what and who)	Model parametrisation (what and who)	Applicable study settings	Real-world uptake	Barriers for uptake	Tool support	IPR considerations
Open Science Tools	Typically a model built by M&S experts using FOSS with accompanying documentation	Data scientists/analysts: direct adaptations to the model code; Other healthcare (re-)users: parametrisation through a web-based 'dashboard' interface	Details of model parameterisation should be provided as part of the documentation to enable reproduction of experiments	All study settings. Where data are confidential then synthetic data may be provided for testing.	Growing uptake in healthcare, examples include reusable models with web-browser interface, and documented open models with dependencies locked down	Confidentiality constraints, lack of incentives to share code, limited technical capacity among stakeholders, and usability challenges for non-technical users	Frameworks (e.g. STARS), infrastructure (e.g. GitHub, Zenodo, Docker) and tools (e.g. R Shiny, Streamlit, WebAssembly)	Requires clear licensing and open-access documentation. Institutional IP policies and lack of licensing clarity may restrict reuse
Black-box model components	M&S expert developers of model components	M&S experts by composing existing models	Mostly M&S experts by providing parameters to model components	Where internal model structure is confidential	Limited in healthcare, but examples in domains such as space and defence	The use of black-box approaches, such as distributed simulation, often requires expertise in applied computing and software engineering; this extends beyond the traditional OR skill-set needed to develop the simulation model	The integration of black-box approaches with commercial simulation products is an area of research. Arguably, commercial vendors have not introduced proprietary tool-level support due to a lack of demand and vendor lock-in	Standards such as IEEE 1516 HLA exist. The implementation of the standards can be protected by copyright, even though the standard itself is public
Modelling Languages and Translations	Domain-specific modelling tool developed by M&S expert	By healthcare stakeholder (using modelling knowledge encoded in domain-specific modelling tool or using participatory modelling with M&S expert)	Healthcare stakeholder – support for parameters more complex than single values; Domain-specific modelling tools are partly self-documenting	Settings with well-established understanding of a general and recurring model structure; participatory modelling settings with stakeholders and M&S expert	Limited in healthcare, but examples in other domains (standard in engineering, but also emerging in airplane fleet management and supply-chain management), BPMN extensions for healthcare, DSML for healthcare	Initial cost of developing domain-specific modelling tool, Higher skill level required for tool creation, cost to maintain translation software	E.g., Eclipse Modelling Framework and related ecosystem, Bizagi, Visual Paradigm, Simul8	Vendor lock-in through reliance on domain-specific modelling tools, which may impose proprietary file formats

based reuse this quickly requires a degree of programming expertise (e.g. available to data scientists/analysts). Another differentiator between open-science-tool based reuse and MLT-based reuse is that well-designed domain-specific modelling tools (in the MLT case) provide a degree of integrated self-documentation, whereas open-science tools often require substantial separate documentation to be made available.

The fourth column briefly discusses characteristics of study settings that make the different reuse modes more or less applicable. Part of the differentiation here is related to confidentiality of model structure – a topic that is further discussed in the final column on IPR considerations. Here, naturally, black-box reuse enables tighter protections of confidentiality and IP than open-science tools would. However, generation of synthetic data can go some way to addressing this issue. MLT-based reuse requires a significant degree of understanding of the study setting, ideally in situations where the likelihood that models will be reused in different contexts is high. As mentioned before, MLT-based reuse also naturally enables participatory approaches to model development.

The next three columns describe issues related to real-world uptake. We give some brief notes on the real-world uptake of each reuse mode and list some tools available to those wishing to create models based on a particular reuse mode. We also describe some of the key barriers to uptake. For open-science tools, these include issues of confidentiality, as noted above, but also a systemic lack of incentives to share code openly. Open-science reuse, being white-box, requires significant programming ability of reusing stakeholders. This is also often true of black-box reuse approaches, not least distributed simulation, which requires the use of complicated technical infrastructures for reuse. This is not always available and this is where MLT-based reuse comes in by providing ways of capturing models closer to the stakeholders' expertise. However, MLT-based reuse comes with a higher initial cost for model development, as there is a need to develop the initial domain-specific modelling tools and not just an individual model.[75]

Finally, we briefly discuss IPR considerations. Clear management of licensing is critical to all three reuse modes, but most importantly in the context of open-science-based reuse, where open-access documentation is also central. Issues with IP management can limit the ability to reuse individual models. For black-box reuse, this is somewhat mitigated in that the individual component models are more easily protected. Even specific implementations of the reuse infrastructure can be copyright protected, even though the relevant standard is public. MLT-based reuse methods, like black-box reuse, can create a degree of vendor lock-in, as they depend on specific technologies and tools (possibly including proprietary file formats) for model development and reuse.

4.1. Hybrid reuse modes

The three modes of model reuse are complementary. Next, we discuss the synergy opportunities created by combining two modes of reuse (cf. Figure 2).

A synergy for model reuse can be identified between black-box model components and FOSS, especially since the development of models in both approaches requires programming knowledge and, thereby, accords the programmer the flexibility of incorporating some of the Findability, Accessibility, Interoperability, and Reusability (FAIR) principles of Open Science tools and making distributed models findable and accessible.[93] This could be achieved, for example, by making distributed simulation federates (each a complete simulation model) available and documented through code repositories and open science archives.

Open Science methods for DES model development can similarly benefit from the interoperability and reusability afforded by black-box model component infrastructures. For example, as model reuse becomes more widespread and complex models start to be routinely developed using FOSS tools, there is likely to be the motivation to use distributed simulation techniques to weave complex simulation federations consisting of available and accessible sub-models, an extension of model reuse which is readily achieved using FOSS programming languages and open science tools at one end, and implementation of distributed simulation standards and APIs at the other.

Next, we consider how MLTs and FOSS tools leverage reusability. A unique aspect of MLTs is the stakeholder taking the role of model builder (Table 1). As modelling languages are domain-dependent, we can consider (say) a modelling language for healthcare developed for a specific regional authority; however, it is arguable that the language will also be largely understood by professionals working in the same domain and across geographies. This increases the potential of model reuse manifold through FOSS implementation of MLTs and the underlying (target) simulators, making them available through open science FAIR principles and related technologies – for example, packages to develop model interfaces, web deployment and web assemblies. Another benefit of this combination is that those who plan to reuse the MLTs do not need to be restricted by the target simulators.

Conversely, MLTs can easily generate the artefacts required to support FAIR principles, making it easier to package simulation models ready to be reused.[94,95]

Finally, with respect to furthering model reusability by combining the approaches on black-box reuse and MLTs, the use of domain-specific MLTs offers the opportunity for automation the generation of interface specifications,[96] such as, for example, the HLA Federate Object Model and the Simulation Object Model. For UML-based MLTs, open source tools such as Papyrus provide an execution environment that can communicate with HLA-compatible applications.[97,98] On the other hand, the developers of MLTs can implement translators using the principle 'distributed simulation artefacts by design', whereby the software generates the interface code for time advancement and message passing; the latter is implemented when such MLT-designed models may need to be linked, making the model reusable as a black box in the HLA model component infrastructure.

Most of the triadic relationships discussed above are technologically feasible with the current state-of-the-art in software engineering, FOSS model development and distributed simulation, while some may need further research, yet others may not be considered realistic or efficacious given the current technology uptake. Irrespective, we believe that the combined application of multiple model reuse modes is a pertinent area of enquiry; an area which has received a shot in the arm with burgeoning use of FOSS and the rapidly advancing open science movement, and the accompanying increase of programming skills in the present generation of simulation modellers.

4.2. A hypothetical case study exemplifying the combined use of the three simulation reuse modes

The example is based on a published study by a co-author on modelling the UK National Blood Service (NBS) supply chain in the Southampton area.[84] The objective of the modelling exercise was to investigate blood ordering policies across multiple hospitals served by a single NBS Processing, Testing, and Issuance (PTI) centre. This PTI centre was responsible for collecting donor blood, testing and processing it into products such as red blood cells (RBCs) and platelets, maintaining inventory and issuing blood units based on hospital demand.

Two models were developed using the commercial simulation package *Simul8*. The first was a conventional model that simulated the operations of one PTI and four hospitals as a single, integrated system. Since blood products have a limited shelf life and must be transfused before expiration, the model incorporated logic to decrement the shelf life of each unit (RBC or platelet) by one minute for every simulated minute. Given the high volume of blood units circulating in the system, this led to substantial computational demands and long experimentation times.

To reduce execution time in the original study, the authors implemented an approach aligned with *black-box model reuse*. Specifically, distributed simulation using HLA-RTI served as the black-box mechanism in the second implementation of the case study. This version comprised one Simul8 model for the PTI centre and four separate models for the individual hospitals, each reflecting a distinct demand profile. Message exchange between these distributed models, along with time synchronisation, was facilitated through a custom black-box HLA/RTI infrastructure.[84]

Thus, the existing study already exemplifies one mode of model reuse: black-box integration. For the purposes of the hypothetical scenario, the remaining part of this section will extend the case study to explore the two other modes of simulation model reuse: reuse enabled by open science tools and reuse through modelling languages and translators.

Model reuse through open science tools: At the time of the original study, Southampton had approximately 16 hospitals, all served by a single NBS PTI centre.[84] Key processes such as physicians requesting blood based on planned clinical activities, checking local hospital inventory, placing orders with the PTI, performing cross-matching, assigning inventory (blood assigned to patients for a clinical procedure), transfusing blood units and expiration of inventory, were broadly similar across the four hospitals that were modelled. However, notable differences existed in terms of hospital characteristics, including size (e.g. large, medium, small), number of physicians, maximum inventory capacity, ordering strategies and inventory policies. Given that multiple hospitals had largely similar operational processes, model reuse using open-source tools such as SimPy could have significantly accelerated the development of individual hospital sub-models. For example, if the FOSS models were developed adhering to frameworks such as Sharing Tools and Artefacts for Reusable Simulations (STARS),[45] it would enable the hospitals, which are part of the same UK NHS ecosystem, to preview the models, validate the conceptual model with user-centric requirements, and download and reuse the model code. This would substantially decrease the time needed to develop a FOSS model from scratch. This illustrates the relevance and value of reuse through open science tools.

Reuse through modelling languages and translators: The NHS delivers healthcare through a network of NHS Trusts. One category of these is Acute Trusts, which provide secondary care in hospitals and specialised medical facilities, including Emergency Departments (EDs). These Trusts typically have a high degree of autonomy in procurement and decision-making. Returning to our hypothetical example, let us assume that the 16 hospitals in Southampton are managed by three Acute NHS Trusts (Trusts A, B and C), all of which are interested in developing simulation models to represent the processes of their respective local blood banks, intending to link these models to the existing NBS PTI model. Based on factors such as available finances, technical expertise within the Trusts' workforce and levels of confidentiality protocols in place, Trust A opts to invest in Simul8, whereas Trusts B and C choose to leverage FOSS by adopting SimPy, recognising the benefits of model reuse. In this example, the Simul8 model for Trust A is akin to a black-box since neither the data, the model logic, nor the model itself is shared. This is in contrast to white-box sharing and reuse of SimPy models by Trusts B and C. Interestingly, the mixing and matching of black versus white-box reuse helps accommodate the varying levels of confidentiality mentioned earlier.

For both the Simul8 and SimPy models to integrate with the NBS PTI model, appropriate interfaces for HLA-RTI are required to enable message exchange and simulation time synchronisation within the overarching simulation federation. Here, the third mode of model reuse, namely, reuse through modelling languages and transformations, becomes critical. This approach can automate the generation of the interface (or 'glue') code needed to facilitate interaction between both Simul8 and SimPy models and the HLA-RTI black-box. By doing so, it significantly reduces the time and complexity that would otherwise be involved in manually implementing such interoperability. This underscores the importance of our third mode of simulation model reuse, which centres on modelling languages and code transformations.

5. Conclusion

Model reuse has the potential to reduce development time in simulation modelling of healthcare and avoid reinventing the wheel in each new project. Reuse does not only have benefits in reducing development time but can also allow the user to benefit from validation carried out by the original developer and aid with experimentation. We believe that increased model reuse will help increase the day-to-day use of simulation to support decision-making for healthcare process improvements.

As discussed, while there is a well-established hierarchy of model reuse, from partial reuse to complete model replication, our work extends this framing by emphasising practical enablers. The triad of open science principles, modelling languages and black-box components method offers concrete means to support operational reuse in contemporary simulation practice. Distributed simulation provides a framework for black-box model reuse with trusted federate models being reused and applied within a new simulation project. The standards that have been developed within the distributed simulation community also make reuse easier. An important driver of model reuse is the accessibility of research, and open science principles have clear links to model reuse. Improving the availability and discoverability of models, as well as ensuring licensing is clear and documentation is available, comprehensive and easily understood will all improve the chances of a model being reused. A DSML, which is designed to enable reuse by encoding knowledge about the domain within the modelling language, can be a useful tool. DSMLs allow for grey-box reuse, reusing generic elements of a healthcare model and including elements that are specific to the particular project. Writing models in standard modelling language or DSML may also give them a longer life than writing them in off-the-shelf packages, which can become obsolete.

While these three modes of model reuse have seen significant research both within and outside the context of simulation development for healthcare in the past, the concrete mechanisms for synergies between the modes are less well understood. There are open research challenges around the development of concrete mechanisms and tools for leveraging the relationships between the three different modes, including for the potential offered by recent developments in artificial intelligence – for example, for generating computer code from natural-language descriptions[99,100] or for managing arguments about simulation studies.[101] Recent moves towards standardised data schemata and data exchange mechanisms (e.g. data dictionaries (https://www.datadictionary.nhs.uk/) or protocols like FHIR) will also be beneficial for simulation reuse by standardising the data-management pipelines required to connect simulations and real-world data. We invite the community to contribute to addressing these challenges.

ORCID iDs

Steffen Zschaler https://orcid.org/0000-0001-9062-6637
Navonil Mustafee https://orcid.org/0000-0002-2204-8924

Funding

The authors disclosed receipt of the following financial support for the research, authorship and/or publication of this article: TM, NM and AH are supported by the Medical Research Council (MR/Z503915/1).

Declaration of conflicting interests

The authors declared no potential conflicts of interest with respect to the research, authorship and/or publication of this article.

References

1. Liu S, Li Y, Triantis KP, et al. The diffusion of discrete event simulation approaches in health care management in the past four decades: a comprehensive review. *MDM Policy Pract* 2020; 5: 2381468320915242.
2. Brailsford SC, Harper PR, Patel B, et al. An analysis of the academic literature on simulation and modelling in health care. *J Simul* 2009; 3: 130–140.
3. Katsaliaki K and Mustafee N. Applications of simulation within the healthcare context. *J Oper Res Soc* 2011; 62: 1431–1451.
4. Erdemir A, Mulugeta L, Ku JP, et al. Credible practice of modeling and simulation in healthcare: ten rules from a multidisciplinary perspective. *J Trans Med* 2020; 18. DOI: 10.1186/s12967-020-02540-4.

5. OMF. OMF: the modeling foundation, 2023, https://www.openmodelingfoundation.org/standards/reusability/ (accessed 18 October 2023)
6. Hussain M, Masoudi N, Mocko G, et al. Approaches for simulation model reuse in systems design – a review. *SAE Int J Adv Curr Pract Mobil* 2022; 4: 1457–1471.
7. Boyle L and Mackay M. A reusable discrete event simulation model for improving orthopedic waiting lists. In: *2022 Winter simulation conference (WSC): Proceedings. Winter simulation conference (WSC)*. Institute of Electrical and Electronics Engineers, 2023, pp. 973–984.
8. Robinson S, Nance RE, Paul RJ, et al. Simulation model reuse: definitions, benefits and obstacles. *Simul Model Pract Theory* 2004; 12: 479–494.
9. Crowe S, Grieco L, Monks T, et al. Here's something we prepared earlier: development, use and reuse of a configurable, inter-disciplinary approach for tackling overcrowding in NHS hospitals. *J Oper Res Soc* 2024; 75: 689–704.
10. Balci O, Arthur JD and Ormsby WF. Achieving reusability and composability with a simulation conceptual model. *J Simul* 2011; 5: 157–165.
11. Jahangirian M, Taylor SJ and Young T. Economics of modeling and simulation: reflections and implications for healthcare. In: *Proceedings of the 2010 Winter Simulation Conference*, 2010, pp. 2283–2292.
12. Mustafee N, Williams MD, Hughes F, et al. Simulation-based study of hematology outpatient clinics with focus on model reusability. In: *Proceedings of the 2011 Winter Simulation Conference (WSC)*, 2011, pp. 1178–1189.
13. IEEE. IEEE 1516-2010: IEEE standard for modeling and simulation (M&S) high level architecture (HLA) – framework and rules, 2010, https://standards.ieee.org/ieee/ 1516/3744/
14. Boyle L, Marshall A and Mackay M. A framework for developing generalisable discrete event simulation models of hospital emergency departments. *Europ J Oper Res* 2021. DOI: 10.1016/j.ejor.2021.12.033
15. Fletcher A and Worthington D. What is a 'generic' hospital model? – A comparison of 'generic' and 'specific' hospital models of emergency patient flows. *Health Care Manag Sci* 2009; 12: 374–391.
16. Anagnostou A and Taylor SJ. A distributed simulation methodological framework for OR/MS applications. *Simul Model Pract Theory* 2017; 70: 101–119. https://doi.org/10.1016/j.simpat.2016.10.007
17. Zhang C, Mao H, Peng G, et al. A novel BOM based multi-resolution model for federated simulation. In: *Proceedings of the 2013 IEEE 17th international conference on computer supported cooperative work in design (CSCWD)*. IEEE, 2013, pp. 178–183. DOI: 10.1109/cscwd. 2013.6580959
18. Li J, Yao Y, Tang W, et al. Research on the development approach for reusable model in parallel discrete event simulation. *Discrete Dynam Nature Society* 2015; 2015: 1–11. DOI: 10.1155/2015/580328.
19. Yonglin L, Weiping W, Qun L, et al. A transformation model from DEVS to SMP2 based on MDA. *Simul Model Pract Theory* 2009; 17: 1690–1709.
20. Fonseca i, Casas P and Ruiz Martin C. Building models in pairs for cross-verification using SDL and DEVS. *Adv Theory Simul* 2024; 7.
21. Ma H, Yao Y and Tang W. An event-thinking development framework for reusable model of parallel and discrete event simulation. In: *Theory, methodology, tools and applications for modeling and simulation of complex systems*. Singapore: Springer, 2016, pp. 37–46. DOI: 10.1007/978-981-10-2663-8_4.
22. Barros F. Abstract simulators for the DSDE formalism. In: *Proceedings 1998 Winter simulation conference*. IEEE, 1998, pp. 407–412. DOI: 10.1109/wsc.1998.745015.
23. Fernandes R, Li B, Benjamin P, et al. Collaboration support for executable enterprise architectures. In: *2009 International Symposium on Collaborative Technologies and Systems*, IEEE, 2009, pp. 520–527. DOI: 10.1109/cts.2009.5067522.
24. Penn ML, Monks T, Kazmierska AA, et al. Towards generic modelling of hospital wards: reuse and redevelopment of simple models. *J Simul* 2020; 14: 107–118.
25. Harper A, Monks T and Mustafee N. Deploying reusable healthcare simulation models in python. In: *2024 Winter simulation conference (WSC)*, 2024, pp. 1173–1184.
26. Monks T, Robinson S and Kotiadis K. Learning from discrete-event simulation: exploring the high involvement hypothesis. *Europ J Oper Res* 2014; 235: 195–205. DOI: 10.1016/j.ejor.2013.10.003
27. Petersen BK, Ropella GE and Hunt CA. Toward modular biological models: defining analog modules based on referent physiological mechanisms. *BMC SystBiol* 2014; 8. DOI: 10.1186/s12918-014-0095-1
28. Teo YM and Szabo C. CODES: an integrated approach to composable modeling and simulation. In: *41st annual simulation symposium (ANSS-41 2008)*. IEEE, 2008, pp. 103–110. DOI: 10.1109/anss-41.2008.24
29. Askari SH, Khan SA, Haris M, et al. Pattern based model reuse using colored Petri nets. In: *19th International conference on computational science and its applications (ICCSA)*, IEEE, 2019; pp. 32–38. DOI: 10.1109/iccsa.2019.000-7.
30. Moallemi M, Wainer G, Bergero F, et al. Component-oriented interoperation of real-time DEVS engines. In: *Proceedings of the 44th Annual Simulation Symposium*. ANSS '11, 2011, pp. 127–134. San Diego, CA: Society for Computer Simulation International.
31. Polack F and Alden K. On developing and validating dynamic systems: simulation engineering. *J Object Tech* 2020; 19.
32. Zschaler S and Polack F. Trustworthy agent-based simulation: the case for domain-specific modelling languages. *Softw Syst Model* 2023: 455–470. DOI: https://doi.org/10.1007/s10270-023-01082-9
33. Ye P, Wang S and Wang FY. A general cognitive architecture for agent-based modeling in artificial societies. *IEEE Trans Comput Soc Syst* 2018; 5: 176–185. DOI: 10.1109/tcss.2017.2777602
34. Carley KM, Morgan GP and Lanham MJ. Deterring the development and use of nuclear weapons: a multi-level modeling approach. *J Defense Model Simul: Appl Methodol Tech* 2017; 15: 483–493. DOI: 10.1177/1548512917706768
35. Edwards KL. Exploring the reusability of discrete-event simulation models: a case study of project challenges and issues of feasibility. In: *2013 IEEE systems and information*

engineering design symposium, IEEE 2013, pp. 7–12. DOI: 10.1109/sieds.2013.6549485

36. Monks T, Robinson S and Kotiadis K. Can involving clients in simulation studies help them solve their future problems? a transfer of learning experiment. *European Journal of Operational Research* 2016; 249: 919–930. DOI: 10.1016/j.ejor.2015.08.037
37. Pouwels XG and Koffijberg H. Introducing open science in teaching health economic modelling. *Pharmacoeconomics-open* 2024, 1–11. DOI: https://doi.org/10.1007/s41669-023-00469-0.
38. Munafò MR, Nosek BA, Bishop DV, et al. A manifesto for reproducible science. *Nature Human Behaviour* 2017; 1: 1–9.
39. Goldacre B, Morton CE and DeVito NJ. Why researchers should share their analytic code. *BMJ: British Medical Journal (Online)* 2019; 367.
40. Health Data Research UK. Open science and open code, 2024, https://www.hdruk.ac.Uk/access-to-health-data/open-science-open-code/ (accessed 2 September 2024).
41. NHS Digital. Statistical publications, open data and data products, 2024, https://digital.nhs.Uk/data-and-information/-statistical-publications-open-data-and-\data-products (accessed 2 September 2024).
42. Monks T and Harper A. Computer model and code sharing practices in healthcare discrete-event simulation: a systematic scoping review. *J Simul 2023*, 1–16. DOI: 10.1080/17477778.2023.2260772.
43. Klebel T, Traag V, Grypari I, et al. The academic impact of open science: a scoping review. *Royal Soc Open Sci* 2025; 12: 241248.
44. Van den Eynden V, Knight G, Vlad A, et al. Towards open research: practices, experiences, barriers and opportunities. *Technical Report, Wellcome Trust* 2016, https://doi.org/10.6084/m9.figshare.4055448 (accessed July 2025).
45. Monks T, Harper A and Mustafee N. Towards sharing tools and artefacts for reusable simulations in healthcare. *J Simul* 2024: 1–20.
46. Lin D, Crabtree J, Dillo I, et al. The TRUST Principles for digital repositories. *Scientific Data* 2020; 7: 144. DOI: 10.1038/s41597-020-0486-7.
47. Smith R, Mohammed W and Schneider P. Packaging cost-effectiveness models in R: a tutorial. *Wellcome Open Res* 2023; 8: 419.
48. Van Lissa CJ, Brandmaier AM, Brinkman L, et al. WORCS: a workflow for open reproducible code in science. *Data Science* 2021; 4: 29–49.
49. Monks T, Currie CS, Onggo BS, et al. Strengthening the reporting of empirical simulation studies: introducing the stress guidelines. *J Simul* 2019; 13: 55–67.
50. Grimm V, Railsback SF, Vincenot CE, et al. The odd protocol for describing agent-based and other simulation models: a second update to improve clarity, replication, and structural realism. *J Artif Soc Soc Simul* 2020; 23.
51. Achter S, Borit M, Chattoe-Brown E, et al. RAT-RS: a reporting standard for improving the documentation of data use in agent-based modelling. *Int J Soc Res Methodol* 2022; 25: 517–540.
52. EQUATOR Network. Enhancing the quality and transparency of health research, 2024, https://www.equator-network.org/reporting-guidelines/ (accessed 2 September 2024).
53. Center for Devices and Radiological Health. Assessing the credibility of computational modeling and simulation in medical device submissions: guidance for industry and food and drug administration staff. Technical Report, U.S. Department of Health and Human Services; Food and Drug Administration, 2023, https://www.regulations.gov/document/ FDA-2021-D-0980-0024
54. Viceconti M and Emili L (eds). *Toward Good Simulation Practice: Best Practices for the Use of Computational Modelling and Simulation in the Regulatory Process of Biomedical Products*. Springer Nature Switzerland, 2024. DOI: 10.1007/978-3-031-48284-7.
55. Sargent RG. Verification, validation and accreditation of simulation models. In: *2000 Winter simulation conference proceedings (Cat. No.00ch37165), Volume 1 2000*, pp. 50–59. DOI: 10.1109/WSC.2000.899697.
56. Monks T and Harper A. Towards sharing tools and artifacts for reusable simulation: deploying a SimPy model as a web app. *Software on Zenodo*, 2024, https://doi.org/10.5281/zenodo.11102678. DOI: 10.5281/zenodo.11102678.
57. Harper A, Monks T, Wilson R, et al. Development and application of simulation modelling for orthopaedic elective resource planning in England. *BMJ Open* 2023; 13.
58. Harper A, Pitt M and Monks T. Open-source modeling for orthopedic elective capacity planning using discrete-event simulation. In: *2023 Winter simulation conference (WSC)*, 2023, pp. 1077–1087. DOI: 10.1109/WSC60868.2023.10408227.
59. Hassanzadeh H, Boyle J, Khanna S, et al. A discrete event simulation for improving operating theatre efficiency. *Int J Health Plan Manag* 2023; 38: 360–379.
60. Zarour K, Benmerzoug D, Guermouche N, et al. A systematic literature review on BPMN extensions. *Bus Process Manag J* 2020; 26: 1473–1503.
61. Pufahl L, Zerbato F, Weber B, et al. BPMN in healthcare: challenges and best practices. *Inform Syst* 2022; 107: 102013.
62. Proudlove NC, Bisogno S, Onggo BS, et al. Towards fully-facilitated discrete event simulation modelling: addressing the model coding stage. *Europ J Oper Res* 2017; 263: 583–595.
63. Antonacci G, Calabrese A, D'Ambrogio A, et al. A BPMN-based automated approach for the analysis of healthcare processes. In: *2016 IEEE 25th international conference on enabling technologies: infrastructure for collaborative enterprises (WETICE)*, 2016, pp. 124–129.
64. Onggo BS and Karpat O. Agent-based conceptual model representation using BPMN. In: *Proceedings of the 2011 Winter Simulation Conference (WSC)*, 2011, pp. 671–682.
65. Heib AR, Currie CSM, Onggo BS, et al. A generalized symbiotic simulation model of an emergency department for real-time operational decision-making. In: *Proceedings of the Winter Simulation Conference 2023*. WSC'23, 2023, pp. 1042–1053. IEEE Press. DOI: 10.5555/3643142.3643228.

66. Gjerloev A, Grieco L, Pagel C, et al. Cancer pathway simulator, https://xip.uclb.com/product/cancer-pathway-simulator (accessed 29 July 2024).
67. Onggo BS, Proudlove NC, D'Ambrogio S, et al. A BPMN extension to support discrete-event simulation for healthcare applications: an explicit representation of queues, attributes and data-driven decision points. *J Oper Res Soc* 2018; 69: 788–802.
68. Spyrou S, Bamidis P, Pappas K, et al. Extending uml activity diagrams for workflow modelling with clinical documents in regional health information systems. In: *Connecting Medical Informatics and Bioinformatics: Proceedings of the 19Th Medical Informatics Europe Conference (MIE2005)*, Geneva, 2005, pp. 1160–1165.
69. Vasilakis C, Lecnzarowicz D and Lee C. Application of unified modelling language (uml) to the modelling of health care systems: an introduction and literature survey. *Int J Healthcare Inform Syst Inform* 2008; 3: 39–52.
70. Neuner C, Bocciarelli P and D'Ambrogio A. BPMN-based simulation analysis of the covid-19 impact on emergency departments: a case study in Italy. In: *2021 Winter Simulation Conference (WSC)*, 2021, pp. 1–12.
71. Ciccozzi F, Malavolta I and Selic B. Execution of UML models: a systematic review of research and practice. *Softw Syst Model* 2019; 18: 2313–2360.
72. Vasilakis C, Lecznarowicz D and Lee C. Developing model requirements for patient flow simulation studies using the unified modelling language (UML). *J Simul* 2009; 3: 141–149.
73. Martin M, Champion R, Kinsman L, et al. Mapping patient flow in a regional Australian emergency department: a model driven approach. *Int Emerg Nurs* 2011; 19: 75–85.
74. Godfrey T, Zschaler S, Batra R, et al. Supporting emergency department risk mitigation with a modular and reusable agent-based simulation infrastructure. In: *2023 Winter Simulation Conference*, 2023, pp. 162–173. IEEE. DOI: 10.1109/wsc60868.2023.10407894.
75. Godfrey T, Batra R, Douthwaite S, et al. A methodology for DSML-assisted participatory agent-based enterprise modelling. In: *15th IFIP WG 8.1 Working Conference on the Practice of Enterprise Modelling (Poem'22)*. New York: Springer, 2022, pp. 201–215. DOI: 10.1007/978-3-031-21488-2_13.
76. Wilsdorf P, Zschaler S, Haack F, et al. Potential and challenges of assurance cases for simulation validation. In: *2024 Winter Simulation Conference (WSC)*, 2024, pp. 2166–2177. DOI: 10.1109/WSC63780.2024.10838818.
77. Ewald R and Uhrmacher AM. SESSL: a domain-specific language for simulation experiments. *ACM Trans Model Comput Simul* 2014; 24: 1–25. DOI: 10.1145/2567895.
78. Mertens J and Denil J. ESS: EMF-based simulation specification, a domain-specific language for model validation experiments. In: *2022 Annual Modeling and Simulation Conference (ANNSIM)*. IEEE, 2022. DOI: 10.23919/annsim55834.2022.9859307.
79. Ruscheinski A, Budde K, Warnke T, et al. Generating simulation experiments based on model documentations and templates. In: *2018 Winter Simulation Conference (WSC)*, 2018. IEEE. DOI: 10.1109/wsc.2018.8632515.
80. Wilsdorf P, Zuska M, Andelfinger P, et al. Validation without data – Formalizing stylized facts of time series. In: *Winter Simulation Conference (WSC 2023)*, 2023. IEEE Press, https://ieeexplore.ieee.org/document/10408388
81. Crnkovic I, Larsson S and Stafford J. Component-based software engineering: building systems from components. *Softw Eng Notes* 2002; 27(3). *Summary of a workshop at 9th IEEE Conf. and Workshops on Engineering of Computer-Based Systems*.
82. Fujimoto RM. *Parallel and Distributed Simulation Systems*. Wiley, 2000.
83. Fujimoto RM. Parallel and distributed simulation. In: *Proceedings of the 2015 Winter Simulation Conference, WSC'15*, 2015, pp. 45–59. IEEE Press. DOI: 10.5555/2888619.2888624.
84. Mustafee N, Taylor SJ, Katsaliaki K, et al. Facilitating the analysis of a UK national blood service supply chain using distributed simulation. *Simulation* 2009; 85: 113–128. https://doi.org/10.1177/0037549708100530
85. Harper A, Mustafee N and Yearworth M. Facets of trust in simulation studies. *Europ J Oper Res* 2021; 289: 197–213.
86. Staff M, Mustafee N and Shenker N. Conceptual modeling for perishable inventory: a case study in human milk banking. In: *2023 Winter Simulation Conference (WSC)*, 2023, pp. 1208–1219.
87. Sargent RG. Verification and validation of simulation models. In: *Proceedings of the 2010 Winter Simulation Conference*, 2010, pp. 166–183.
88. Mustafee N, Katsaliaki K and Taylor SJE. Distributed approaches to supply chain simulation: a review. *ACM Trans Model Comput Simul* 2021; 31: 1–31. DOI: 10.1145/3466170.
89. Mustafee N and Fakhimi M. Towards an integrative taxonomical framework for hybrid simulation and hybrid modelling. In: *Hybrid modeling and simulation: conceptualizations, methods and applications*. Springer, 2024, pp. 3–22.
90. Taylor SJE, Mustafee N, Strassburger S, et al. The SISO CSPI PDG standard for commercial off-the-shelf simulation package interoperability reference models. In: *2007 Winter simulation conference*, 2007. IEEE. DOI: 10.1109/wsc.2007.4419652.
91. Taylor SJE, Turner SJ, Strassburger S, et al. Bridging the gap: a standards-based approach to OR/MS distributed simulation. *ACM Trans Model Comput Simul* 2012; 22: 1–23. DOI: 10.1145/2379810.2379811.
92. Mustafee N and Taylor SJ. Speeding up simulation applications using wingrid. *Concur Comput Pract Exp* 2009; 21: 1504–1523.
93. Wilkinson MD, Dumontier M, Aalbersberg IJ, et al. The fair guiding principles for scientific data management and stewardship. *Scientific Data* 2016; 3: 1–9.
94. Bocciarelli P, D'Ambrogio A, Mastromattei A, et al. Automated development of web-based modeling services for MSAAS platforms. In: *Proceedings of the symposium on model-driven approaches for simulation engineering*. Mod4sim '17, 2017. San Diego, CA: Society for Computer Simulation International.
95. Wallis K, Polack F and Zschaler S. MDE for open science. *Technical report, MDENet, the expert network in model-*

driven engineering, 2023, https://mde-network.com/wp-content/uploads/2023/01/11539_MDENet_OpenScience-edit-26012023.pdf

96. Gomes C, Meyers B, Denil J, et al. Semantic adaptation for FMI co-simulation with hierarchical simulators. *SIMULATION* 2018; 95: 241–269. DOI: 10.1177/0037549718759775
97. Possik J, Asgary A, Solis AO, et al. An agent-based modeling and virtual reality application using distributed simulation: case of a covid-19 intensive care unit. *IEEE Trans Eng Manag* 2022; 70: 2931–2943.
98. Gorecki S, Ribault J, Zacharewicz G, et al. Risk management and distributed simulation in papyrus tool for decision making in industrial context. *Comput Ind Eng* 2019; 137: 106039.
99. Jackson I, Jesus Saenz M and Ivanov D. From natural language to simulations: applying AI to automate simulation modelling of logistics systems. *Int J Prod Res* 2023; 62: 1434–1457. DOI: 10.1080/00207543.2023.2276811.
100. Monks T, Harper A and Heather A. Unlocking the potential of past research: using generative AI to reconstruct healthcare simulation models, 2025. DOI: 10.48550/ARXIV.2503.21646.
101. Zschaler S, Wilsdorf P, Godfrey T, et al. Dialectic models for documenting and conducting simulation studies: exploring feasibility. In: *Winter simulation conference*, 2025.

Author biographies

Steffen Zschaler is a Reader in Software Engineering at King's College London. His research focuses on model-driven engineering (MDE) with a particular focus on its applications to the engineering of simulations and the specification of optimisation problems. You can find out more at https://steffen-zschaler.de/.

Navonil Mustafee is Professor of Analytics and Operations Management at the University of Exeter Business School. His research focuses on simulation methodologies, including hybrid modeling and simulation, and their application in healthcare, supply chain management, circular economy and climate change adaptation and resilience. He is a Joint Editor-in-Chief of the Journal of Simulation.

Alison Harper is a Lecturer in Operations and Analytics at University of Exeter Business School. Her research interests include applied health and care research, reusable simulation models, real-time discrete-event simulation and hybrid modeling.

Thomas Monks is an Associate Professor of Health Data Science at University of Exeter Medical School. His research interests include open science for computer simulation, Generative AI, urgent and emergency care, and real-time discrete-event simulation.

Bhakti Stephan Onggo is a Professor of Business Analytics at the University of Southampton Business School. His research focuses on simulation modeling methodology and its applications in digital twin, health care, disaster management and supply chain. His website is https://bsonggo.wordpress.com/.

Christine Currie is a Professor of Operational Research in Mathematical Sciences at the University of Southampton. She is the former Editor-in-Chief for the Journal of Simulation. Her research interests include simulation optimisation, applications of simulation in health care, optimal pricing and disaster management.

Fiona Polack is Professor at the University of Hull, UK. Her research interests are in model-driven engineering and the engineering of complex systems simulation, with specific contributions in relation to the expression and maintenance of fitness-for-purpose of scientific simulations, using argumentation approaches.