Proceedings of the 2024 Winter Simulation Conference
H. Lam, E. Azar, D. Batur, S. Gao, W. Xie, S. R. Hunter, and M. D. Rossetti, eds.

EXTENDING SIMULATION MODELING METHODOLOGY FOR DIGITAL TWIN APPLICATIONS

Bhakti Stephan Onggo[1], Christine S. M. Currie[2]

[1]CORMSIS, Southampton Business School, University of Southampton, Southampton, UK
[2]CORMSIS, School of Mathematical Sciences, University of Southampton, Southampton, UK

ABSTRACT

The number of reported works in Digital Twin (DT) has significantly increased in recent years. A fundamental component of a DT is the digital representation of a real object, process, or system that decision makers wish to evaluate or manage. One of the most utilized digital representations is a simulation model. Traditionally, simulation has been primarily employed for offline applications. However, the introduction of DT technology has enabled an effective online connection between a simulation model and its real counterpart, allowing the use of simulation as part of dynamic control systems for real-time decision making. This paper discusses whether steps in simulation modeling methodology such as conceptual modeling, input modeling, model development, validation, and output analysis, need adjusting for DT applications.

1 INTRODUCTION

As a relatively new concept co-existing in a wide range of application domains, it is inevitable that the Digital Twin (DT) has many definitions; some of which were discussed in Barricelli et al. (2019). In simulation communities, several related terms were introduced from at least as early as the 1990s (e.g., online simulation, real-time simulation, dynamic data-driven adaptive simulation and symbiotic simulation) (Onggo 2019). At the 1993 Winter Simulation Conference, Rogers and Gordon (1993) mentioned the term *online simulation* when they were referring to Rogers and Flanagan's work that used simulation as a tool to control a real system by finding the best decision to respond to a change in the real system. This description is very similar to what we know now as DT. One of the relevant definitions of the term *real-time simulation* is a simulation that can run as fast as the "wall-clock" time. In many applications we are wishing to use a DT to make operational decisions in real-time, where "real-time" is somewhat dependent on the application and the decision being made, e.g., in recent work with a manufacturer, their interpretation of real-time was within 3 minutes. However, it can also mean returning results in seconds. In such cases, a fast simulation is essential. There is a dedicated symposium for real-time simulation applications called The International Symposium on Distributed Simulation and Real-Time Applications (DS-RT) that has been around since 1997. Another relevant term is *symbiotic simulation*. The parallel and distributed simulation working group at the 2002 Dagstuhl seminar on grand challenges for modeling and simulation, introduced the term symbiotic simulation system to refer to a new paradigm for discrete event simulation (DES) that interacts with the physical system in a mutually beneficial way (Fujimoto et al. 2020). Subsequent work by Aydt et al. (2008) and Onggo et al. (2018) have shaped this term to become very similar to what we know as DT today. For example, if we compare the three types of symbiotic system in Onggo et al. (2018) and the three types of DT in Kritzinger et al. (2018), the similarity is remarkable. The term digital model in Kritzinger et al. (2018) is exactly what a conventional simulation application looks like.

In the past five years, the term DT has become the most popular in simulation journals and simulation conference proceedings. For example, in the 2023 Winter Simulation Conference, there were at least 30 papers with the word "digital twin" in their titles compared to one paper each for real-time simulation and symbiotic simulation (none used online simulation). A quick search on Scopus database for papers with the words "digital twin", "online simulation", "real-time simulation" and "symbiotic simulation" in their titles, abstract and keywords published in journals, proceedings and books that have "simulation" in their titles (i.e., `SRCTITLE(simulation)`) reveals the same pattern (see Figure 1). We plot the data from 1993 to mark the time when Rogers and Gordon (1993) first described the term online simulation at the Winter Simulation Conference that is very similar to what we know as DT today. The figure shows that the term DT has become dominant since 2019. Although conceptually similar to DT, online simulation and symbiotic simulation have never received the same growth as DT since their inception. Please note that this plot underestimates the number of DT publications as it does not count those in publication outlets that do not use simulation in their titles. For example, DT is also a popular topic in engineering although a significant number of the papers do not focus on the simulation aspect.

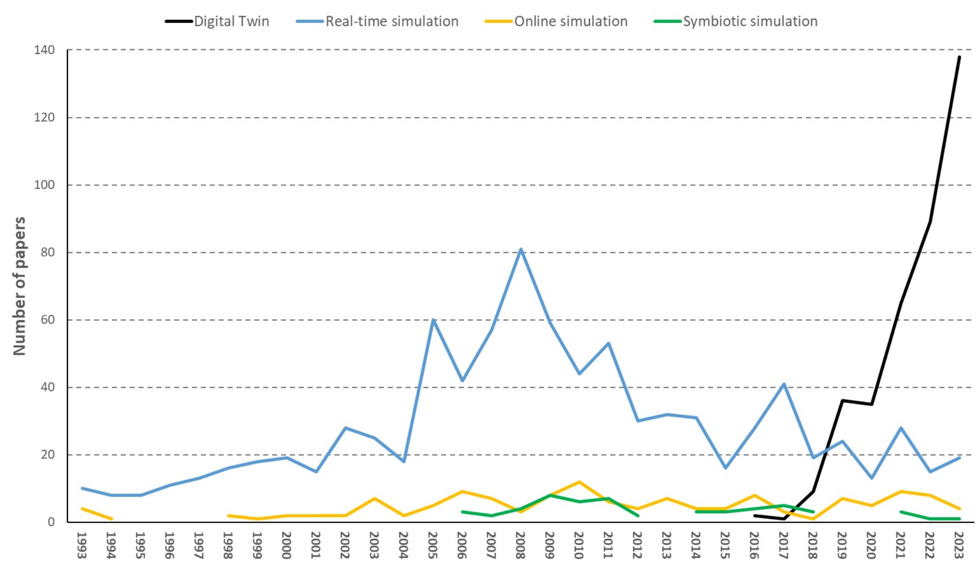

Figure 1: The number of papers on digital twin, symbiotic simulation, online simulation and real-time simulation in simulation publication outlets (1993-2023).

Most (if not all) DT definitions agree that a simulation model is one of the most utilized digital representations of real systems. Conventionally, simulation models have been developed using historical data. Once a model is validated, it is used to make decisions about the real system (primarily for selecting the best design or policy). Hence, there is no live connection between a simulation model and its real system (offline simulation). However, the introduction of DT technology, which enables live connections between a simulation model and its real system; and in particular a data feed from the real system to the simulation model with a return flow of forecasts and recommendations. This opens up new opportunities for simulation modeling to be used in real-time operational decision support. Given the obvious potential of DT, it is important to evaluate if the current simulation modeling methodology needs adjusting for DT applications, which is the objective of this paper. This paper does not discuss the simulation of DT.

Following this introduction, the remainder of this paper is organized as follows. Section 2 will briefly present conventional simulation modeling methodology. This is followed by Section 3 where we present the main discussion on how simulation modeling methodology is affected by its use in DT applications. Finally, we end our paper with conclusions and future research directions.

2 CONVENTIONAL SIMULATION MODELING METHODOLOGY

All good simulation textbooks (e.g., Law and Kelton (2000), Nelson and Pei (2021)) explain how to develop and use conventional simulation models correctly (i.e. simulation modeling methodology). Likewise, each year, the Winter Simulation Conference publishes tutorial papers on the methodology used in different stages of a simulation project. The main stages are briefly explained below.

Problem structuring: Operations Researchers use simulation models to solve real problems, from the well-defined to the more complex (the latter often being described as wicked problems). While it is straightforward for the well-defined problems, understanding the more complex problems may require tools such as soft systems methodology (Checkland and Poulter 2020) or strategic options development and analysis (Ackermann and Eden 2020). The main objective is to avoid solving the wrong problems.

Conceptual modeling: From the problem definition, stakeholders need to communicate about the simulation model requirements. Therefore, there is a need for a method to support the communication between stakeholders which may include people unfamiliar with simulation. Therefore, a method to represent the simulation model conceptually (independent of the simulation software) is useful. Onggo (2010) discussed some methods for conceptual model representation. A conceptual model comprises components such as objectives, inputs, outputs, contents (scope, structure, level of detail, assumptions, and simplifications) and data requirements (Robinson 2008). The objectives component describes the purpose of the model which will guide the remaining components. The inputs and outputs specify the inputs (e.g., parameters, decision variables) to the model and model outputs (e.g., performance indicators), respectively. The contents component captures the structure and the fidelity of the model. This is typically represented using a process flow diagram in DES, stock and flow diagram in system dynamics (SD) or state chart in agent-based simulation (ABS). Recently, Monks et al. (2019) developed guidelines that included how to document conceptual model components for DES, SD and ABS.

The objective of conceptual modeling is to decide "what to model, and what not to model" (Robinson 2015). This paper will not enter the debate on various conceptual modeling definitions. Instead, we direct interested readers to Robinson (2008).

Input modeling: A simulation model needs to represent the underlying uncertainty in the real-word system by applying the appropriate distribution functions. A slightly less discussed (although important) aspect of input modeling is how to address the input data uncertainty. Therefore, the objective of input modeling is to represent and account for the uncertainty in the input data correctly. The Winter Simulation Conference provides tutorials on distribution fitting (Law 2013) and input uncertainty (Song et al. 2014).

Model coding or development: This is the stage where we implement the conceptual model into an executable simulation model using a computer programming language or simulation software.

Verification and validation: The objective of verification is to make sure the simulation model implements the conceptual model correctly. Modern simulation software provides various verification tools such as debugger, animation, and watch variables. A simulation model is a simplified representation of its real system. The simplification is guided by the objective of the simulation study. Hence, any simulation model needs to be validated against its intended use. Validation can be defined as "substantiation that a model within its domain of applicability possesses a satisfactory range of accuracy consistent with the intended application of the model" (Sargent 2013). In other words, verification is concerned with building the model right and validation is concerned with building the right model.

Output analysis: Once validated, a simulation model can be used for decision making. The objective of output analysis is to ensure we analyze the simulation outputs correctly. This includes how to correctly deal with concepts like terminating or non-terminating simulation, warm-up period, number of replications and statistical comparison tests (Currie and Cheng 2016; Law 2023). It can also incorporate experimentation and simulation optimization, both of which are frequently used in decision support systems.

3 SIMULATION MODELING METHODOLOGY FOR DT APPLICATIONS

In simulation textbooks that we are aware of, the modeling methodology is discussed mainly in the context of the conventional simulation applications (not DT applications). In this section we will consider possible adjustments that are needed when moving from a conventional simulation model to a DT for five of the key stages of a simulation project: conceptual modeling, input modeling, model development (or coding), validation and output analysis.

3.1 Conceptual Modeling

The initial ideas for simulation conceptual modeling (Robinson 2008) were produced with conventional simulation applications in mind. Hence, we need to evaluate whether adjustments are needed if the simulation is used in DT applications. DT applications affect conceptual model specification in several ways.

Firstly, unlike conventional applications which use only historical data, DT applications use both historical data and real-time data. Hence, the inputs component needs to include the real-time data specification (e.g., data rate, data format, data stream, data quality, data source, sensor locations), when the real-time data becomes historical data, and the criteria to identify pattern changes in the real-time data (e.g., when the real system changes or when the sensors are faulty).

Secondly, DT applications often use an ensemble of models, sometimes as part of a multi-fidelity approach. For example, Rhodes-Leader et al. (2022) used multi-fidelity modeling to support short-term operational airline disruption management decisions by employing a deterministic integer program as the low fidelity model and simulation as the high fidelity model. Cao et al. (2021) proposed a simulation optimization for DT using a multi-fidelity framework where a neural network was used as the low fidelity model and simulation as the high fidelity model. Hence, for DT applications, we need to specify the objectives, outputs, and contents for each model. The rule to choose which model is used for a given condition as well as the interaction between models also need to be specified. If the models form a hierarchy, then several layers of objectives, outputs, and contents components as well as the interactions between layers need to be specified.

Thirdly, DT applications can be viewed as a hybrid model (Onggo et al. 2018). Therefore, the conceptual model needs to specify the interfaces between simulation models and other DT components such as optimization and machine learning models.

Finally, since the simulation model in a DT application needs to adapt to changes in the real system, the simulation model may alter relevant input models to respond to changes in the input data or alter its model structure to respond to a change in the structure of the real system. Therefore, for DT applications, the conceptual model needs to specify the rules to adapt the simulation models (e.g., when and how). For example, the mechanism to trigger the model change could be condition-based or time-based (Friederich and Lazarova-Molnar 2023). Given the possibility of frequent changes to the model, one important question is how to best represent such a dynamic model.

3.2 Input Modeling

For a simulation in a DT application, input modeling covers both the input distributions used to govern the model and setting the initial simulation state. In a DT application, the input distributions may be updated automatically based on data from the real system, which differentiates it from input modeling for conventional simulations. Setting the starting state tends not to be a big question in conventional simulation models where models are either run in the steady-state for *non-terminating* simulations or from empty for *terminating* simulations. In what follows, we discuss some recent methods and potential research gaps in automating input modeling alongside a more detailed discussion of setting the starting state.

For a DT simulation model to be a useful guide to future behavior, it is typically run from a simulation state that represents the current state of the real system. This can be achieved using a *hot start*, which for a DES means that queues and activities contain the same number of entities as their counterparts in the real system. In some cases the amount of information available in the real system may be less than that included in a full simulation state. For example, the simulation may account for every operation in an assembly line but data may only be collected on the work in progress in different sections of the line. In such cases, translating the real system data into simulation states may involve some random sampling of the possible simulation states that could result in the same set of real system data.

There is relatively little discussion of automating input modeling in the simulation literature. Barlas and Heavey (2016) describe the KE tool that is designed to extract and process data and prepare outputs in a format that is easily readable by a simulation package. This automates the process of moving from data to input models but does not account for automatic updating of input models. Input models for simulations can take two forms: empirical distributions, where values are sampled at random and with replacement from real data to determine quantities of interest; or parametric statistical distributions. Where empirical distributions are used, updating involves validation of new data and may also include a process for removing old data. Where parametric distributions are used, Bayesian updating or other methods from reinforcement learning could be useful for updating distributions based on new data. More work is needed on translating methods from statistics and machine learning to a simulation setting.

3.3 Model Development

In DT applications, the real system may change its structure during the simulation run (e.g., processes can be streamlined, or a new process might be added). Hence, there is a need to adapt the simulation model to respond to the structural changes in the real system. However, if the changes happen regularly, it is not practical to develop the computerized model manually due to the time it takes to develop and validate the model. Therefore, some research has focused on process mining to address this challenge.

Process mining is a substantial research area in which process models are created by mining administrative data in task or event logs (van der Aalst 2016). We consider here work specific to process modeling for simulation. This has also relevance to generalizable models which are designed to work in different settings. Heib et al. (2023) describes how process mining can be used to translate timestamp data for patients in the Emergency Department of a hospital into a Petri-Net model that can be incorporated into a DES. The algorithm used for the process mining is designed to be sufficiently flexible that it can account for complex processes including simultaneous activities (e.g., waiting for test results while being seen by a clinician) and prioritization. A similar work for a manufacturing application is done by Friederich and Lazarova-Molnar (2022). They develop a process mining method to generate a Petri-Net model that can be used to analyze the system's reliability from an event log and a state log. Another similar method is developed in Lugaresi and Matta (2021) who use data logs from sensors on a manufacturing line to identify nodes and arcs in a directed graph, which can then be converted into a digital twin simulation model. They include a tuning algorithm which uses a mathematical program to reduce the number of nodes in the directed graph, hence reducing the complexity of the model, to best meet a set of *model adequacy score functions*. This helps to reduce the potential for generating over-complex *spaghetti models*. Likewise, Sungkono et al. (2023) develop a graph-based algorithm to discover complex process structures such as invisible tasks in non-free choice for stacked branching relationships condition from large event logs. The algorithm was evaluated using several cases including hazardous medical waste management and hospital record management. One of the main challenges in using process mining to discover processes to be used in a simulation model is striking the correct level of detail for the intended purpose of the simulation model; hence, the algorithm should avoid producing an overly detailed spaghetti model and yet be able to capture the important (often complex) process structures.

3.4 Validation

In DT applications, changes in a real system may occur while its simulation model is running. Therefore, a validation method that handles live data or *online validation* is needed. Figure 2 illustrates one possible DT scenario. Initially, the real-time data (black line) does not consistently deviate outside the 95% confidence interval of the simulation model (solid and dashed light brown lines). However, at time 19, the real-time data surges above the upper confidence level. This should trigger the model adaptation algorithm (e.g., automatic distribution fitting, process mining). The revised model will need to be re-validated. Hence, there is a period where the simulation model is not valid (consequently it cannot be used to make decisions). This illustration highlights two questions: when is the best time to trigger the model adaptation and revalidation, and how. The model adaptation and revalidation process takes time; hence, their computational costs need to be minimized. However, this needs to be balanced with the risk of having an invalid simulation model for too long. We call this period where the simulation model is no longer valid the *simulation downtime*. On the trigger for model validation, Friederich and Lazarova-Molnar (2023) suggested mechanisms: on demand (or condition-based), periodic (or time-based) and continuous.

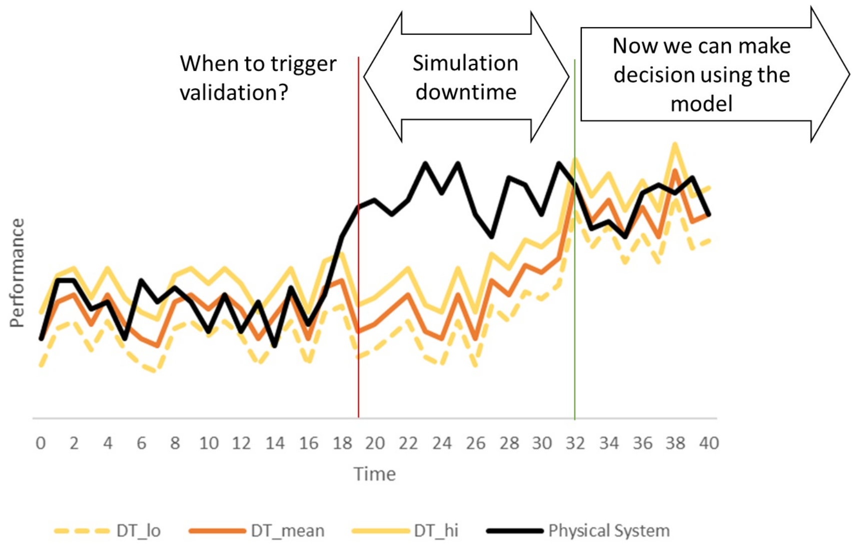

Figure 2: Simulation model validation for DT.

Conventionally, simulation validation is done offline (without any live connection with the real system). Sargent (2013) categorized the offline validation methods into two dimensions: observable versus non observable systems and subjective versus objective. Offline simulation validation methods can still be used in DT applications in certain situations such as when the duration in which we can afford not to have a valid simulation model is long enough, and when we can collect enough data since the real-system changed for model validation. Where the digital twin model is used only at specific decision points, e.g., when a major event occurs, there can be time to apply offline methods in between decision epochs, as discussed in Rhodes-Leader and Nelson (2023). These would still need to be automated but the time constraint on completion is much more relaxed.

If changes in the real system are frequent, there is a greater need for speeding up the validation process. One way to achieve this is through automating the validation process. Researchers have been looking into automating the validation process. Langenbach and Rabe (2023) identified which validation methods can be automated based on their characteristics. One automated validation or *auto-validation* method is by using test-driven simulation development (Onggo and Karatas 2016). They demonstrated how unit testing tools can be utilized for auto-validation in simulation. Friederich and Lazarova-Molnar (2023) expanded the discussion by proposing a validation framework that enables automatic simulation model validation. Auto-validation is especially important when the risk of having an invalid model for too long is high.

Another way to deal with frequent changes in the real system is by using online validation methods. The frequent real system changes may reduce the amount of usable data for conducting an offline validation method. In the following, we will explain several online validation techniques for DT applications that have been proposed.

Recent work has made use of frequency analysis to detect differences between the outputs of a simulation model and the corresponding results from the real system. Such methods typically consider just one output or KPI (key performance indicator) and treat this as a signal or time series. This allows methods from signal processing and time series analysis to be used to compare the signals from the real system and the simulation model. The aim is to detect when the time series output of a simulation no longer matches the behavior of the real system.

Lugaresi et al. (2019) describes *quasi trace driven simulation* (QTDS), which compares the power spectral densities of the signals, essentially the contributions of each frequency to the final signals. Both the simulation and real signals are split into batches and differences between the power spectra of the two signals are estimated and averaged. The result is defined as the *spectral indicator*, a measure between 0 and 1, where 1 corresponds to identical signals and 0 to no match. Results show that the spectral indicator is higher for similar systems and lower for different systems but no suggestion is made as to a sensible threshold for determining whether the traces can be regarded as similar or not. In subsequent work, QTDS is combined with dynamic time-warping, a method used in time series analysis for measuring the similarity between time series that may have different, and potentially varying, speeds (Lugaresi et al. 2022).

Morgan and Barton (2022) develop statistical tests for discrimination of two systems by comparing their Fourier coefficients. Two separate tests are described, the *waFm* test, which compares the weighted mean frequency content of two system trajectories and the *wafmGOF* test, which detects divergence in specific frequency bands. The tests are designed to hold for steady-state simulations and the use of Fourier transforms in this case has definite advantages over other methods such as wavelet transformations because Fourier coefficients are unchanged by timeshifts in the underlying signal. Similar to Lugaresi et al. (2019), they batch their outputs and calculate Fourier coefficients within each batch and some thought is needed as to determine the correct batch size. The benefit of these tests is that they are written as hypothesis tests and experiments suggest that they have good power, particularly for low frequencies, and the size of the test is close to the target. This makes analysis of the results more straightforward.

Other approaches have compared the time series of model outputs using frequency analysis. For example, Rhodes-Leader and Nelson (2023) describes a statistical method based on hypothesis testing that compares the output of the simulation with that of the real system. The method is designed for the case where the system is optimized in *decision epochs*, which correspond to the points at which the simulation model is called on to support a decision. Tests for misalignment of the two series are carried out offline, in between these decision epochs rather than happening continuously.

For DT applications that require regular hot-starts, Oakley et al. (2020) developed the difference (or Δ) method to validate simulation models. At the time of hot-start, the simulation and real system show the same output. As time elapses, there will be some differences in the output values and the variance increases. This method collects data about the difference in the performance since hot-start and generates an empirical distribution at certain points in time. For example, if the hot-start is done every Monday, the method generates six distributions for the difference between Monday and Tuesday, ..., Monday and Sunday, respectively. Then, the method compares each of the six distributions generated by the simulation model and the observed (near-real-time) data to determine if the simulation model is still valid.

Finally, we agree with Hua et al. (2022) that simulation validation for DT applications is challenging. They listed several other challenges such as uncertainty and sensitivity analysis as well as dealing with complex of system-of-systems. Friederich and Lazarova-Molnar (2023) provided a good discussion on the validation framework for models simulating Cyber-Physical Production Systems. Although our paper does not discuss the simulation of DT, their discussion on model validation at runtime is particularly relevant to our discussion on online and automatic simulation model validation. These studies show that research into online and automatic simulation model validation for DT applications is still at an early stage.

3.5 Output Analysis

Digital twin simulations can be used for both forecasting and controlling stochastic systems. Forecasting involves running the simulation models forward in time to estimate likely outcomes with notable examples found in healthcare (Harper and Mustafee 2019; Hoot et al. 2008) and manufacturing (Ghasemi et al. 2023). In these cases, it is important that the simulation model accurately predicts future behavior and provides some measure of the uncertainty in the expected results. In addition to the business or management related KPIs (e.g., patient waiting time, production throughput, profit), we also need to consider the technical performance measures. The reason is that DT is typically used as a dynamic control system, a real-time simulation optimization tool, an automation tool, and/or a tool to increase situational awareness. Therefore, the experimentation needs to assess how good the simulation model is in supporting these DT roles. For example, the following technical measures (among others) can be used to asses how good the simulation model is in supporting DT as part of a dynamic control system.

- *Response time* can be used to measure the simulation downtime or to evaluate how quickly the simulation model can be changed to respond to changes in the real system.
- *Overshoot* measures how much the real system's output that is controlled by the simulation model exceeds the threshold during the simulation model downtime.
- *Settling time* measures how long it takes for the real-word system's output to reach and stay within the desired range.

To measure the contribution of the simulation model to the situational awareness of the people using the DT application, techniques from other fields such as physiology, psychology and behavioral science may be useful. If we use one of the most cited situational awareness framework (Endsley 1995), we need to measure how the simulation model affects the three levels of awareness: the people's awareness of the status of the various components in the system, their comprehension on what is going on in the system, and their projection of what will happen in the near future.

The use of digital twin simulations for real-time optimization and control of stochastic systems is developing at speed. The potential of simulation optimization as a decision-making tool within an Industry 4.0 framework was discussed in Xu et al. (2016) and subsequent work has built on these ideas, with the common theme being the need to develop simulation optimization methods that return results quickly.

Simulation optimization in a traditional setting can often involve a relatively lengthy experiment, which is typically not feasible with a DT. Kleijnen et al. (2005) review methods for optimizing the design of simulation experiments for conventional simulation and suggest three reasons for experimenting with a simulation model: (1) gaining a basic understanding of a model or system; (2) making robust decisions; and (3) comparing different options. The second of these aims becomes less appropriate when using a DT as by making decisions with up-to-date information, we are necessarily making the process more robust to changes in the underlying environment. While DT systems mostly focus on the third aim of comparing different options, synonymous with simulation optimization, it is important that the first of these aims is realized in the background to facilitate fast optimization when new information becomes available. Appropriate designs are covered in Sanchez et al. (2020) and these could be used to determine the most appropriate fast simulation optimization algorithms or in conjunction with metamodel-based simulation optimization, in which a metamodel that describes the output of the simulation model for a set of inputs is fitted offline and then used to provide a recommendation when called on by the digital twin. Goodwin et al. (2024) and Cao et al. (2021) use machine learning models as metamodels within the optimization, both allowing for the possibility that the machine learning model may not be perfectly accurate.

Multi-fidelity simulation optimization provides a more general term that encompasses metamodel-based simulation optimization, but the low-fidelity model needs not be a static statistical or machine learning model. For example, Xu et al. (2016) use a queueing model as their simplified model. This returns results quickly and can be less computationally demanding to fit as it follows a similar structure to the underlying more complex simulation model. Kang et al. (2021) describe how to make use of several low-fidelity models with different biases to improve the estimate of the high-fidelity simulation model. Using a slightly different architecture, the *Generalized Ordinal Learning Framework* or GOLF, builds models for functions of interest offline before the exact scenario is known and then supplements with additional, carefully chosen simulation replications in real-time. This helps to speed up the online optimization (Pedrielli et al. 2019).

4 CONCLUSIONS AND FUTURE RESEARCH DIRECTIONS

Simulation modeling methodology conventionally does not consider the live connection between a real system and its simulation model. Therefore, current simulation modeling methodology has mostly been developed for simulation models that do not need frequent changes (reasonably static). However, DT applications may require the simulation models to be updated more frequently to reflect changes in the real systems. This paper has discussed that this difference requires some adjustments to the simulation modeling methodology when developing and using simulation models for DT applications. This paper has specifically addressed the required adjustments in the conceptual modeling, input modeling, model development (or coding), validation and output analysis stages of a simulation project.

More research in simulation modeling methodology is needed to ensure that simulation is developed and used correctly for DT applications. Based on what we have discussed in this paper, we have identified the following lines of research.

- There is a need for research to identify what to include in a conceptual model specification as a consequence of the use of ensembles of models and a mix of live and historical data, as well as the need to interact with other DT components. There is an open question on the best way to represent a conceptual model when frequent changes are expected.
- With the use of live data and the possibility of frequent changes in the input data patterns, the methods to automate the updating of input distributions as new data become available are needed. However, research into automatic input modeling for DT applications is lacking.
- Research should also focus on how to decide which state variables to use in translating the current state of the system into initial conditions for a hot start simulation, particularly considering cases where the state information available in the real system is less than that in the simulation model. Screening experiments (see Kleijnen et al. (2005)) seem to have an important role to play here.
- More research is needed in the area of process mining for automatic updating of a simulation model. There have been several research projects considering the use of process mining to discover processes from the real system that will be used to adapt the simulation model. However, the current process discovery algorithms have not been able to discover all possible complex processes and their interactions.
- Most of the existing work on online validation is designed for the steady-state case while many DT simulations describe systems with time-dependent output. Hence, more frequency-domain methods for non-stationary output that provide automated statistical tests and thresholds for detecting a difference in behavior are needed to improve online validation.
- Refining methods for detecting changes in real processes will be particularly important for fast-changing systems. For example, process mining may be used to detect the deviation between processes in the real system and processes in its simulation model (*conformance checking*). Time series change point detection techniques (Aminikhanghahi and Cook 2017) may also be useful.
- Although there has been considerable research on speeding up simulation optimization methods to make them suitable for real-time applications, there are additional challenges related to the use of the simulation within a dynamic control system. For example, the reactiveness, meaning the speed at which recommendations are given, needs to be optimized because if it is too high, the real system may become unstable and if it is too low, the real system may not perform optimally.

More experiments that measure technical performance related to the use of DT for dynamic control systems such as responsiveness, overshoot and settling time are needed.

- More research is needed to measure the impact of simulation on the situational awareness of the people in a DT setting, and more generally on the behavior of decision-makers when responding to a DT providing real-time decision support.

These research areas are important because a simulation model is one of the most used digital representations of a real system in DT applications. Hence, we need to make sure that it is developed and used correctly.

AUTHOR BIOGRAPHIES

BHAKTI STEPHAN ONGGO is a Professor of Business Analytics at the University of Southampton. He is a member of the Centre for Operational Research Management Sciences and Information Systems (CORMSIS) and Centre for Healthcare Analytics. His research interests include simulation modeling methodology and its applications in digital twin, health care, disaster management and supply chain. His e-mail address is b.s.s.onggo@soton.ac.uk. His website is https://bsonggo.wordpress.com/.

CHRISTINE S.M. CURRIE is a Professor of Operational Research in Mathematical Sciences at the University of Southampton and a member of the Centre for Operational Research, Management Sciences and Information Systems (CORMSIS). She is Editor-in-Chief for the Journal of Simulation. Her research interests include simulation optimization, applications of simulation in health care, optimal pricing and disaster management. Her email address is christine.currie@soton.ac.uk and her homepage is https://www.southampton.ac.uk/people/5wzzxf/professor-christine-currie.

REFERENCES

Ackermann, F. and C. Eden. 2020. “Strategic Options Development and Analysis”. In *Systems Approaches to Making Change: A Practical Guide*, edited by M. Reynolds and S. Howell, 139–199. London: Springer.

Aminikhanghahi, S. and D. J. Cook. 2017. “A Survey of Methods for Time Series Change Point Detection”. *Knowledge and information systems* 51(2):339–367.

Aydt, H., S. J. Turner, W. Cai, and M. Y. H. Low. 2008. “Symbiotic Simulation Systems: An Extended Definition Motivated by Symbiosis in Biology”. In *Proceedings of the 22nd Workshop on Principles of Advanced and Distributed Simulation (PADS)*, 109–116. July 3rd-6th, Rome, Italy: IEEE.

Barlas, P. and C. Heavey. 2016. “KE Tool: An Open Source Software for Automated Input Data in Discrete Event Simulation Projects”. In *2016 Winter Simulation Conference (WSC)*, 472–483 https://doi.org/10.1109/WSC.2016.7822113.

Barricelli, B. R., E. Casiraghi, and D. Fogli. 2019. “A Survey on Digital Twin: Definitions, Characteristics, Applications, and Design Implications”. *IEEE Access* 7:167653–167671.

Cao, Y., C. Currie, B. S. Onggo, and M. Higgins. 2021. “Simulation Optimization for a Digital Twin Using a Multi-Fidelity Framework”. In *2021 Winter Simulation Conference (WSC)*, 1–12 https://doi.org/10.1109/WSC52266.2021.9715498.

Checkland, P. and J. Poulter. 2020. “Soft Systems Methodology”. In *Systems Approaches to Making Change: A Practical Guide*, edited by M. Reynolds and S. Howell, 201–253. London: Springer.

Currie, C. S. and R. C. Cheng. 2016. “A Practical Introduction to Analysis of Simulation Output Data”. In *2016 Winter Simulation Conference (WSC)*, 118–132 https://doi.org/10.1109/WSC.2016.7822084.

Endsley, M. R. 1995. “Measurement of Situation Awareness in Dynamic Systems”. *Human Factors* 37(1):65–84.

Friederich, J. and S. Lazarova-Molnar. 2022. "Data-Driven Reliability Modeling of Smart Manufacturing Systems Using Process Mining". In *2022 Winter Simulation Conference (WSC)*, 2534–2545 https://doi.org/10.1109/WSC57314.2022.10015301.

Friederich, J. and S. Lazarova-Molnar. 2023. "A Framework for Validating Data-Driven Discrete-Event Simulation Models of Cyber-Physical Production Systems". In *2023 Winter Simulation Conference (WSC)*, 2860–2871 https://doi.org/10.1109/WSC60868.2023.10407382.

Fujimoto, R., W. H. Lunceford Jr, E. Page, and A. M. Uhrmacher. 2020. "Grand Challenges for Modelling and Simulation". In *Dagstuhl Seminar Report 350*, 1–75. Leibniz-Zentrum für Informatik: Schloss Dagstuhl https://doi.org/10.4230/DagSemRep.350.

Ghasemi, A., Y. T. Yeganeh, A. Matta, K. E. Kabak and C. Heavey. 2023. "Deep Learning Enabling Digital Twin Applications in Production Scheduling: Case of Flexible Job Shop Manufacturing Environment". In *2023 Winter Simulation Conference (WSC)*, 2148–2159.

Goodwin, T., J. Xu, N. Celik, and C.-H. Chen. 2024. "Real-Time Digital Twin-Based Optimization with Predictive Simulation Learning". *Journal of Simulation* 18:47–64.

Harper, A. and N. Mustafee. 2019. "A Hybrid Modelling Approach Using Forecasting and Real-Time Simulation to Prevent Emergency Department Overcrowding". In *2019 Winter Simulation Conference (WSC)*, 1208–1219 https://doi.org/10.1109/WSC40007.2019.9004862.

Heib, A. R., C. S. Currie, B. S. Onggo, H. K. Smith and J. Kerr. 2023. "A Generalized Symbiotic Simulation Model of an Emergency Department for Real-Time Operational Decision-Making". In *2023 Winter Simulation Conference (WSC)*, 1042—-1053 https://doi.org/10.1109/WSC60868.2023.10407981.

Hoot, N. R., L. J. LeBlanc, I. Jones, S. R. Levin, C. Zhou, C. S. Gadd *et al*. 2008. "Forecasting Emergency Department Crowding: A Discrete Event Simulation". *Annals of Emergency Medicine* 52(2):116 – 125.

Hua, E. Y., S. Lazarova-Molnar, and D. P. Francis. 2022. "Validation of Digital Twins: Challenges and Opportunities". In *2022 Winter Simulation Conference (WSC)*, 2900–2911 https://doi.org/10.1109/WSC57314.2022.10015420.

Kang, Y., L. Mathesen, G. Pedrielli, F. Ju and L. H. Lee. 2021. "Multifidelity Modeling for Analysis and Optimization of Serial Production Lines". *IEEE Transactions on Automatic Control* 66(8):3460–3474.

Kleijnen, J. P., S. M. Sanchez, T. W. Lucas, and T. M. Cioppa. 2005. "A User's Guide to the Brave New World of Designing Simulation Experiments". *INFORMS Journal on Computing* 17:263–289.

Kritzinger, W., M. Karner, G. Traar, J. Henjes and W. Sihn. 2018. "Digital Twin in Manufacturing: A Categorical Literature Review and Classification". *Ifac-PapersOnline* 51(11):1016–1022.

Langenbach, K. and M. Rabe. 2023. "Approach for Classifying the Automatability of Verification and Validation Techniques". In *2023 Winter Simulation Conference (WSC)*, 1665–1675 https://doi.org/10.1109/WSC60868.2023.10407984.

Law, A. M. 2013. "A Tutorial on How to Select Simulation Input Probability Distributions". In *2013 Winter Simulations Conference (WSC)*, 306–320 https://doi.org/10.1109/WSC.2013.6721429.

Law, A. M. 2023. "Design and Analysis of Simulation Experiments Using Three Simple Statistical Formulas". In *2023 Winter Simulation Conference (WSC)*, 1487–1500 https://doi.org/10.1109/WSC60868.2023.10407750.

Law, A. M. and W. D. Kelton. 2000. *Simulation Modeling & Analysis*. 3rd ed. New York: McGraw-Hill, Inc.

Lugaresi, G., G. Aglio, F. Folgheraiter, and A. Matta. 2019. "Real-Time Validation of Digital Models for Manufacturing Systems: A Novel Signal-Processing-Based Approach". In *Proceedings of the 15th International Conference on Automation Science and Engineering (CASE)*, 450–455. August 22nd-26th, Vancouver, BC, Canada: IEEE.

Lugaresi, G., S. Gangemi, G. Gazzoni, and A. Matta. 2022. "Online Validation of Simulation-Based Digital Twins Exploiting Time Series Analysis". In *2022 Winter Simulation Conference (WSC)*, 2912–2923 https://doi.org/10.1109/WSC57314.2022.10015346.

Lugaresi, G. and A. Matta. 2021. "Automated Manufacturing System Discovery and Digital Twin Generation". *Journal of Manufacturing Systems* 59:51–66.

Monks, T., C. S. Currie, B. S. Onggo, S. Robinson, M. Kunc and S. J. Taylor. 2019. "Strengthening the Reporting of Empirical Simulation Studies: Introducing the STRESS Guidelines". *Journal of Simulation* 13(1):55–67.

Morgan, L. E. and R. R. Barton. 2022. "Fourier Trajectory Analysis for System Discrimination". *European Journal of Operational Research* 296(1):203–217.

Nelson, B. and L. Pei. 2021. *Foundations and Methods of Stochastic Simulation: A First Course*. 2 ed. New York: Springer.

Oakley, D., B. S. Onggo, and D. Worthington. 2020. "Symbiotic Simulation for the Operational Management of Inpatient Beds: Model Development and Validation using Δ-Method". *Health care management science* 23:153–169.

Onggo, B. S. 2019. "Symbiotic Simulation System (S3) for Industry 4.0". In *Simulation for Industry 4.0: Past, Present, and Future*, edited by M. M. Gunal, 153–165. Cham: Springer.

Onggo, B. S. and M. Karatas. 2016. "Test-Driven Simulation Modelling: A Case Study using Agent-Based Maritime Search-Operation Simulation". *European Journal of Operational Research* 254(2):517–531.

Onggo, B. S., N. Mustafee, A. Smart, A. A. Juan and O. Molloy. 2018. "Symbiotic Simulation System: Hybrid Systems Model Meets Big Data Analytics". In *2018 Winter Simulation Conference (WSC)*, 1358–1369 https://doi.org/10.1109/WSC.2018.8632407.

Onggo, S. 2010. "Methods for Conceptual Model Representation". In *Conceptual Modeling for Discrete-Event Simulation*, edited by S. Robinson, R. Brooks, K. Kotiadis, and D.-J. V. D. Zee, 353–370. Boca Raton: CRC Press.

Pedrielli, G., K. Selcuk Candan, X. Chen, L. Mathesen, A. Inanalouganji, J. Xu, *et al*. 2019. "Generalized Ordinal Learning Framework (GOLF) for Decision Making with Future Simulated Data". *Asia-Pacific Journal of Operational Research* 36(6):1–35.

Rhodes-Leader, L., B. L. Nelson, B. S. Onggo, and D. J. Worthington. 2022. "A Multi-Fidelity Modelling Approach for Airline Disruption Management using Simulation". *Journal of the Operational Research Society* 73(10):2228–2241.

Rhodes-Leader, L. A. and B. L. Nelson. 2023. "Tracking and Detecting Systematic Errors in Digital Twins". In *2023 Winter Simulation Conference (WSC)*, 492–503 https://doi.org/10.1109/WSC60868.2023.10408052.

Robinson, S. 2008. "Conceptual Modelling for Simulation Part I: Definition and Requirements". *Journal of the Operational Research Society* 59(3):278–290.

Robinson, S. 2015. "A Tutorial on Conceptual Modeling for Simulation". In *2015 Winter Simulation Conference (WSC)*, 1820–1834 https://doi.org/10.1109/WSC.2015.7408298.

Rogers, P. and R. J. Gordon. 1993. "Simulation for Real-Time Decision Making in Manufacturing Systems". In *1993 Winter Simulation Conference (WSC)*, 866–874 https://doi.org/10.1145/256563.256863.

Sanchez, S. M., P. J. Sanchez, and H. Wan. 2020. "Work smarter, not harder: a tutorial on designing and conducting simulation experiments". In *2020 Winter Simulation Conference (WSC)*, 1128–1142 https://doi.org/10.1109/WSC48552.2020.9384057.

Sargent, R. G. 2013. "Verification and Validation of Simulation Models". *Journal of Simulation* 7:12–24.

Song, E., B. L. Nelson, and C. D. Pegden. 2014. "Advanced Tutorial: Input Uncertainty Quantification". In *2014 Winter Simulation Conference (WSC)*, 162–176 https://doi.org/10.1109/WSC.2014.7019886.

Sungkono, K. R., R. Sarno, B. S. Onggo, and M. F. Haykal. 2023. "Enhancing Model Quality and Scalability for Mining Business Processes with Invisible Tasks in Non-Free Choice". *Journal of King Saud University-Computer and Information Sciences* 35(9):101741.

van der Aalst, W. 2016. *Process Mining: Data Science in Action*. 2 ed. Berlin: Springer.

Xu, J., E. Huang, L. Hsieh, L. H. Lee, Q. S. Jia and C. H. Chen. 2016. "Simulation optimization in the era of Industrial 4.0 and the Industrial Internet". *Journal of Simulation* 10(4):310–320.