36th European Modeling & Simulation Symposium, 024
21th International Multidisciplinary Modeling & Simulation Multiconference

2724-0029 © 2024 The Authors.
doi: 10.46354/i3m.2024.emss.024

The coexistence of commercial and open-source simulation solutions in the Digital Twin era

Denisa Borges[1, 2*], Luís M. S. Dias[1,2], Ricardo J. Machado[1,2] and Marcelo F. N. Henriques[1,2]

[1] CCG/ZGDV Institute, University of Minho, Azurém Campus, Guimarães, 4800-058, Portugal
[2] ALGORITMI Research Centre, University of Minho, Braga, 4710-057, Portugal

[*]Corresponding author. Email address: denisa.borges@ccg.pt

Abstract

In general terms, simulation involves modelling a theoretical or real-world system, or some of its parts. Indeed, it is a discipline of high applicability. Nevertheless, its usage is not ubiquitous. In spite of this, a lack of recent research addressing this issue was notice. For simulation to achieve its full potential, it shall overcome some of its limitations and barriers. However, current (proprietary) commercial simulation solutions are dependent on the organizations which develop them, which can limit the modelers' ability to replicate specific behaviors or integrate the simulation model with other solutions, e.g., MES, Digital Twin. Moreover, the licensing aspect of commercial solutions can be a barrier to the use of simulation, either by a large team, or its adoption by SMEs. Therefore, this article structures and clarifies some simulation types typically used, to promote its usage. This is followed by discussing the limitations and barriers of commercial simulation solutions, suggesting opportunities provided by open-source simulation solutions. Nonetheless, we strongly believe that there is room for both commercial and open-source simulation solutions to coexist, which may yield significant benefits for both solutions and for the simulation discipline as a whole. Moreover, such coexistence may promote the use of simulation as a component of the Digital Twin.

Keywords: Discrete-event simulation; Open-source solutions; Digital twin

1. Introduction

Simulation is a widely used discipline, proven by the sheer number of articles detailing its application capabilities, both of its various types and also application domains, in both real and theoretical contexts. It is indeed a main pillar of Industry 4.0, and an integral part of the Digital Twin solution (Cañas et al., 2021).

Many of the scientific papers do not include the simulation concepts and principles, e.g., in their modeling approach presentation (King and Harrison, 2013). However, these are relevant to simulation newcomers, to clarify these concepts, particularly because many of them were defined decades ago, and are scattered in several books and journals, which makes it more challenging for those newcomers to have a holistic perspective on simulation (Kelton et al., 2015).

Hence, we feel that there is a need to concentrate those simulation concepts and principles in one place, for newcomers to be able to get to know and understand them.

Moreover, current commercial simulation solutions present some challenges. In this way, this article aims at discussing such challenges, as well as presenting the responses provided by open-source simulation solutions to those challenges (Dagkakis et al., 2013). It is important to emphasize that the discussion of this article will solely focus on discrete event simulation (DES). Recent studies related to this research topic were not found. Albeit, some studies do in fact raise awareness of open-source solutions (Vieira et al., 2019). Yet, they do not discuss the limitations and barriers which commercial simulation solutions might have, and how open-source simulation solutions might answer.

This article is structured in four sections. The first section is the current one, in which a brief contextualization is made, presenting the research opportunities found. Furthermore, both the research questions and objectives are presented, as well as the research methodology. The second section contains the literature review, in which some of the fundamental concepts of simulation are outlined. Subsequently, the advantages and disadvantages of commercial simulation solutions are enumerated, followed by the open-source simulation solutions. The third section discusses the results obtained from the literature review. The fourth and final section draws conclusions about the research developed, as well as presenting limitations and future research opportunities.

1.1. Research questions and objectives

To bridge the identified gaps, a set of research questions have been outlined, which contributed to the definition of the research objectives. These are:

RQ1: What are the fundamental concepts inherent to simulation?

RO1: Clarify the fundamental concepts of simulation.

RQ2: What different approaches are there which can improve the simulation process?

RO2: Analyze and propose different approaches to improve the simulation process.

1.2. Methodology

This section defines the methodology upon which the performed research is based. The methodology serves as the research strategy which outlines how this research was conducted. Therefore, presenting the methodology is crucial to demonstrate the process itself and be transparent. To accomplish this, a study plan was established, encompassing all components, including primary, secondary, and tertiary variables. Above all, there must be a commitment to rigorously adhere to the defined methodology throughout the research process (Garg, 2016). The methodology employed in the research performed for this article was a systematic literature review, complemented with the snowball effect.

This review establishes a foundation for the continuous advancement of knowledge, aids in addressing gaps, and highlights areas in need of further research. Moreover, review articles play a crucial role in charting the course for future research directions (Webster and Watson, 2002). Hence, journals and conference articles were scrutinized, while fundamental concepts of simulation were gathered mainly through books, given that many of their constructs were defined several decades ago. Article selection was based on a ten-year period, with some specific exceptions for books and the snowball effect.

Additionally, keywords were defined to focus the research in the particular topic and the aforementioned objectives. It is noteworthy that, on certain cases, some articles were obtained through the snowball effect. This was triggered by the analysis of an article gathered from the systematic literature review, which referenced an article where the fundamental concepts of simulation were presented in a clearer, deeper, or the original manner. In **Figure 1.** it is possible to glimpse the literature review strategy used in the systematic literature review employed in this article.

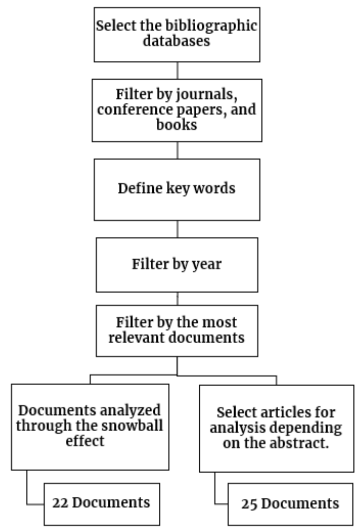

Figure 1. Strategy for selecting the articles analyzed

For full transparency and enhance reader's experience, a table containing all articles used was created. This table details the source, keywords, period and type of document information used to filter each article, following the systematic strategy defined. This table can be accessed here https://n9.cl/4vm41o

In summary, forty-seven documents were selected, of which twenty-two were obtained through the snowball effect. For better understanding, **Table 1** in "Appendix A" was prepared, which synthesizes the defined keywords as well as the filters used for document selection.

2. State of the art

2.1. What is simulation?

The strength of simulation is related to its ability to look into the future and thus make assumptions (Rozinat et al., 2009). Simulation can be considered as a representation of behaviors of a real-world system within a certain time interval. In order for the simulation process to occur, it is necessary to collect historical data from the systems to be simulated and consequently analyze this data so that, in the end, it is possible to predict the results and thus improve the functioning of the systems (Sharma, 2015) .

The authors Wooley et al. (2023) summarized the capability of simulation into four levels: modeling, analysis, prediction, and prescription. The first level concerns the computational modeling of a system, for which historical data must logically be collected to be incorporated into the computational model. The second level [analysis] is reached during experiments, corresponding to the testing phase of the developed model. The prediction level is achieved when the simulation model can predict the outcome that a real-world system will obtain if it adopts a certain behavior. The fourth and final level, prescription, uses the simulation results to improve the system, and thus prescribe a better solution. In the simulation discipline, the terminology has become a 'standard,' meaning that definitions are clear on their own. However, the same does not apply to the Digital Twin. Therefore, Wooley et al. (2023) have adapted the 'capabilities' of simulation to define a language that is more easily understood in the area of the Digital Twin.

In the last decade, the importance of simulation has grown significantly. The interfaces and designs of products make [commercial] simulation software more attractive and user friendly, reducing the learning curve required for its use. With advancements in hardware and software, even beginners can create visually appealing 3D simulations (Brannick and Coghlan, 2005; Joines and Roberts, 2013). Over the past years, the acquisition and maintenance costs of information technology (IT) have decreased, while the performance and capabilities of simulation tools have improved. They are capable of solving increasingly complex problems, in shorter amounts of time (Bélanger and Venne, 2010).

Simulation can model a variety of systems, due to its ability to represent entity movement within the system itself, and analyze the impact of resources on system performance. Through simulation models, it is possible to predict the impact of changes in production and, to some extent, how services will be affected by these same changes (Pegden, 2015). Attached to the simulation model is uncertainty, for the model may or may not describe the system's behavior in an exemplary manner. To address this gap, it will be necessary to conduct a set of validation tests (Bossel, 1994).

Given the multiplicity and high applicability of simulation, it can be categorized into three types (Kelton et al., 2015) , which will be explained below.

2.2. Types of Simulation

2.2.1. Static Simulation Models vs. Dynamic Simulation

In static simulation, time does not play a natural role (Kelton et al., 2015) in the simulation model. The term "simulation model", in this case, refers to the equations, mathematical expressions, and computer algorithms that simulate the behavior and performance of a system in real-world situations (Abar et al., 2017). Examples of a static simulation include generating random numbers to simulate a lottery game or evaluating a financial statement of profits and losses. According to authors Kelton et al. (2015), although there is a notion of time in the model, it is necessary for this time to induce changes in the structure or operations of the model for it to no longer be considered a static model. In dynamic simulation, time plays a natural role (Kelton et al., 2015), meaning that time will influence the results of the simulation and is an essential part of the model's structure and functioning. A practical example involving dynamic simulation is the case of supply chains that include transportation or logistics, as it is necessary to represent the starting point, movement, and arrivals over time.

2.2.2. Continuous Change Dynamic Models vs. Discrete Change Models

Dynamic models are characterized by the presence of state variables that, together, describe the state of a system at any point in the simulation. For example, simulating the water level in a tank, in this case, the state variable would be the water level, which can change continuously with the increase or decrease of water; thus, the system undergoes continuous change as the state variables change over time. In contrast, if the state variables change at instantaneous points, the model would be characterized by discrete changes. In reality, the vast majority of queue simulation models are of discrete change because state variables, such as queue length, only change at moments of discrete events, such as the arrival of customers (Kelton et al., 2015).

2.2.3. Deterministic Models vs. Stochastic Models

A deterministic model is characterized by the fact that the input variables guiding the simulation model are constant and non-random. An example of such models is the case of simple production lines, which are composed of a queueing system with fixed service times for each part, and fixed arrival times between parts. With this type of simulation, it is expected that, unless we change the input values, the simulation results will be the same in multiple model executions. However, in the vast majority of simulation models, input values are not static but rather random (Kelton et al., 2015). Stochastic simulation is an important tool for operations research professionals to evaluate the performance of systems with random behavior and mathematically unattainable performance measures. It can also be an alternative when certain experiments involve significant costs or when there is no analytical way to infer about the real system (Corlu et al., 2020).

As mentioned earlier, the vast majority of simulation models are of stochastic origin, so it makes sense to delve into stochastic simulation. One of the most important steps in simulation is input modeling, which involves selecting probability distributions, i.e., input models that best characterize the system inputs. Next, the generation of sample paths follows, in which random variables are generated to perform the simulation. Lastly, output analysis is conducted, where output simulation data is collected and analyzed to develop performance measures. Naturally, these [output] measures are affected by the [input] distributions (Corlu et al., 2020). However, there is uncertainty associated with simulation outputs, justified by the lack of knowledge about input models. This uncertainty is referred to as "input uncertainty". One way to reduce simulation uncertainties could be to execute the simulation a greater number of times, which could be done computationally, i.e., replications. However, the uncertainty of inputs cannot be addressed in this manner but rather through the collection of more input data. Unfortunately, this is often a very arduous and challenging task, and, in some instances, impossible (Corlu et al., 2020). Nevertheless, despite the existence of methods to represent input uncertainty, many of these have not been integrated into all software packages.

2.3. Discrete event system (DES)

A discrete event system (DES) is a discrete event-driven state system in which state changes are entirely dependent on the occurrences of discrete events over time. Manufacturing systems, transportation systems e.g., urban traffic networks, communication systems e.g., wireless networks; are examples of discrete event systems (Choi, 2013). Discrete event simulation utilizes a logical or mathematical model of a physical system to depict state changes at specific points over the simulation time. It is necessary that, as soon as a state change occurs, a precise description of it be made (Chen, 2015). DES is a powerful tool for accurately modeling complex systems (Dehghanimohammadabadi and Keyser, 2015) and supporting decision-making in production planning and logistics tasks (Lang et al., 2021).

2.4. Computational Simulation

Through simulation, it is possible to draw important conclusions about the behavior of real-world systems. These conclusions are derived from computational modeling, which means using a computer to model behaviors based on logical, statistical, and mathematical relationships (McHaney, 2009). This idea aligns with the theory of authors Kelton et al. (2015), that computational simulation refers to the method that studies a wide variety of real system models through numerical evaluation using software to mimic system operations or characteristics.

Like most analysis methods, it involves systems and models. Computational simulation deals with system models, which can include: a packet distribution network, warehouses and transportation links, a supermarket with inventory control, cashiers, and customer service (Kelton et al., 2015).

2.5. Commercial Solutions

A way to analyze real-world systems with uncertainty could be through simulation. There are numerous commercial simulation solutions available, such as ARENA, SIMIO, and AnyLogic (Garwood et al., 2018; Peyman et al., 2021). Typically, when discussing simulation, the concept of a 'simulation agent' arises. It is simply an entity, notion, or a software abstraction akin to programming constructs like objects, methods, and functions. A simulation agent presents a higher level of abstraction because instead of being expressed in attributes, it is typically defined based on intended actions. Moreover, a simulation agent is reactive, capable of communicating with users, and actively responding to changes in its environment. With this concept understood, it becomes easier to grasp the meaning of Agent Based Modeling and Simulation (ABMS). This is nothing more than a way to categorize computational models that generate dynamic behaviors and enable the creation of communication protocols between agents (Abar et al., 2017).

It is important to understand that agent-based modeling can be done in two ways: through commercial simulation solutions, for example, simulation packages available on the market; or through programming languages that meet specific requirements for simulation agents (Belyaev and Desyatirikov, 2023). Agent-based modeling can be done on desktops or on large-scale computing clusters, starting small and scaling up to larger models using dedicated ABMS toolkits, using familiar programming languages. (Macal and North, 2009).

According to Mohammed Hasan et al. (2019), simulation solutions is used to solve a variety of problems, ranging from layout and movement analysis in production lines, to production capacity. These commercial simulation solutions may employ a structure of simulation modeling based on intelligent objects (Tsaousoglou and Manesis, 2014). Intelligent objects consist of models which can be reused in multiple modeling projects. These objects are easily stored and shared (Houck and Whitehead, 2019). These commercial simulation solutions allow the creation of a visual model of the system under investigation by drawing objects directly on the screen. Objects could represent queues or arrival points.

Commercial simulation solutions can indeed be used to create models of various systems, regardless of complexity; with a modern graphical interface (GUI), which may allow for programming code in a specific or common programming language for model development (Antonova et al., 2023). Some commercial simulation solutions enable both 2D and 3D visualization, simulation of supply chains, manufacturing systems, as well as transportation logistics. Additionally, advanced forecasting, analysis, transportation scheduling, supply chain management, and exporting the model results in a spreadsheet (e.g., in a *.CSV file) or to an external database (e.g., SQL) (Abar et al., 2017).

Currently, there are also commercial simulation solutions available to simulate the movement of a group of robots and even model multi-object systems. Depending on their needs, the modelers can define drives, friction, and other simulation parameters (Siwek et al., 2019). This way, modelers are not as dependent on the software own definitions. However, although commercial simulation solutions integrate a set of realistic activities; the truth is that they make a great effort to incorporate all operational policies into a single model (Dehghanimohammadabadi and Keyser, 2017); hence, changing a policy or creating a new one is a quite challenging task for the modeler.

Commercial simulation solutions have contributed to the dissemination of discrete event simulation in both academic and industrial communities. Furthermore, the strength of these solutions, as previously noted, lies in their highly advanced graphical interfaces (GUIs) and highly effective computational resources (Dagkakis and Heavey, 2016).

The reasons for adopting commercial simulation solutions are numerous, including: more appealing graphic user interface, reduce modeling effort (through intelligent objects), and better efficiency in outputting results (computational resources management).

By contrast, these current and emerging simulation solutions leave their modelers at the mercy of the goals of the organizations that develop and market these same commercial simulation solutions. Such challenges can range from changing an intelligent object, as previously mentioned; to connecting to an external database, or modeling a completely different behavior.

Albeit, the most notable challenge, and the first one to be faced by modelers is price. The price may entail two components: cost of acquisition (one-time payment), and maintenance (recurring payment). Many organizations, especially SMEs, do not have the capacity for an investment of this magnitude (Lang et al., 2021). Those licenses do not grant the right to revisit the source code (Belyaev and Desyatirikov, 2023).

Faced with these issues, simulation, instead of being seen as an opportunity for improvement, is viewed as an obstacle to achieving organizational goals.

2.6. Open-source Solutions

In the real world, commercial simulation solutions may capture most system behaviors. Though, simulation solutions fail to handle certain activities involving complex human decision making, or highly computational support tools developed to enhance the efficiency of processes within an organization. Moreover, in order to achieve long term strategic planning objectives, it is crucial to integrate these decision-making activities into a simulation model.

Limitations of commercial simulation solutions become a barrier when implementing complex human decision making, e.g., bucket brigade; without knowing the logic behind the commercial simulation solution itself. Therefore, the simulation model must be more effective and robust after incorporating both simulation and a highly computational support tool (Dehghanimohammadabadi and Keyser, 2017).

The main difference between commercial and open-source solutions is the availability of the source code. That is, the source code of commercial simulation solutions is proprietary, i.e., it is not available to the public; whereas in the latter case, the source code is available and accessible to everyone; thus, open-source [code] (Belyaev and Desyatirikov, 2023).

This source code is the basis for the commercial simulation solutions, hence, it defines every aspect of it. For example, the software logic, the intelligent objects' behavior, and external connections. Users of commercial simulation solutions (with proprietary source code) are constrained due to the inherent restrictions imposed on them (Banik and Zimmer, 2022).

The fact that the source code is proprietary prevents competitors from copying the original logic, which took development effort to produce. However, by not being open, it also presents a barrier to modelers.

Open-source simulation solutions allow users to add components that other software may not possess; or even, in some cases, create their own palettes of high level objects, not being limited by the resources of proprietary programming or scripting language (King and Harrison, 2013).

Moreover, by having their source code available to everyone, open-source solutions – whether they are focused on simulation or not – offer greater transparency. This transparency offers new opportunities to: know and being able to modify the original logic, and crowdsource support (maintenance and new features) and intelligent objects.

The authors Dagkakis and Heavey (2016) find potential in the use of open-source simulation solutions, as an alternative to commercial simulation solutions; which include: frequent updates (maintenance), user support through 'User' and 'Programming' manuals (crowdsource), and 3D user interfaces – very similar to commercial simulation solutions.

A very important feature of open-source solutions is that they allow users to add new [intelligent] objects, which means that modelers are not restricted to the goals of the organizations which develop the commercial simulation solutions. Moreover, some of these solutions enable the capture of synthetic data in fully customized virtual environments (Mousavi and Estrada, 2021).

According to the researchers Dagkakis et al. (2014), there is a need to develop simulation objects to model production flow control. One way to address this need may involve using open[-source] solutions, which allow for the demonstration and development of transparent and collaborative solutions. This shift enables positive cost-benefit relationships, as these options are less expensive to access, install, and handle; not to mention the fact that they are typically less demanding on computing resources, hence, requiring more affordable computer equipment to run (Ani et al., 2022).

The authors Dagkakis and Heavey (2016) suggest the implementation of an open repository for DES intelligent objects, in which a user can use and adapt those shared intelligent objects according to her needs – and share her new approach. For example, User A develops and shares an intelligent object mimicking the behavior of a gas-powered AGV (Automated guided vehicle). User B can use that original intelligent object shared by User A, and adapt it to be battery-powered, including logic to stop a task and direct itself to the charging station. User B can share that new intelligent object.

The fact that there are no proprietary licenses, potentially makes this repository inclusive and extensible Dagkakis and Heavey (2016) – not limited to the organization developing the commercial simulation solution.

Discrete event simulation presents some constraints which limit its adoption in industries – among other fields. These include the inherent costs of collecting and managing input data, as well as the costs of integrating simulation solutions with the organization's data infrastructure (e.g., databases, ERP, MES) and withing its own organizational processes.

Due to these issues, researchers Barlas and Heavey, (2016) advocate for the importance of automated input data for real-time simulation, as well as the relevance of interoperability between simulation software and data infrastructure.

Interoperability, namely the ability for simulation models to communicate with other solutions – such as databases, ERP, MES – is a critical necessity to be able to reach the Digital Twin concept. I.e., in order for the (digital) simulation model to be a perfect twin of the (real) system, the two must be able to interchange data in both ways.

Moreover, coupled with the digitization that the Industry 4.0 brought, a range of opportunities arose, including the Digital Twin (Mourtzis, 2020). Currently, there is still no clear definition of what a Digital Twin is (Cespedes-Cubides and Jradi, 2024).

Two of the major references related to the digital twin are from Mr. Michael Grieves and NASA (National Aeronautics and Space Administration), respectability:

> "a digital informational construct about a physical system could be created as an entity on its own. This digital information would be a 'twin' of the information that was embedded within the physical system itself" (Grieves, 2016).

> "integrated multi-physics, multi-scale, probabilistic simulation of a vehicle or system that uses the best available physical models, sensor updates, fleet history, etc., to mirror the life of its flying twin" (Glaessgen and Stargel, 2012; Shafto Mike et al., 2010)

The commonalities among the studies on Digital Twins concern the fundamental structure of the Digital Twin, and the flow of information between the physical asset and its digital counterpart. Albeit, the same authors highlight that some studies utilize isolated digital simulations, referring to their solution as a Digital Twin – which those authors disagree (Cespedes-Cubides and Jradi, 2024).

By opting for the Digital Twin concept, along with intelligent algorithms, organizations may monitor data, improve the operations of their systems, and develop innovative products and services; as the Digital Twin may be used to bridge the gap between input data uncertainty and system optimization (Ojstersek et al., 2023).

Certain open-source event-based simulation solutions extend specific libraries, allowing for modeling systems in the cloud. This enables developers to define the best approach for resource allocation among existing data centers. Additionally, they can also generate information about request response times; that is, the time it will take to process the requests between the cyber and physical world (Sitaram et al., 2014).

3. Results and Discussion

There are numerous articles demonstrating the ability of simulation to improve the performance of real-world systems. However, it was evident that the basic concepts associated with the definition of simulation itself are not clearly discussed; they are typically solely applied. This gap created the opportunity to discuss the simulation concepts.

Afterwards, the commercial simulation solutions were discussed. It was possible to verify that, despite simulation being applied in multiple contexts, there are still several barriers associated to its inception and ubiquitous usage. Examples of these are licensing (cost), closed-source code (logic), and limitations on data integration. Hence, users are at the mercy of the organizations which develop and market those commercial simulation solutions. These constraints place simulation as a barrier, rather than a tool for system improvement / optimization.

Some authors point to open-source simulation solutions as a possible answer to the aforementioned barriers and limitations. Open-source simulation solutions offer more permissive licenses (often free of charge), a transparent predefined logic (for model flexibility and adaptability), crowdsource intelligent objects (a future benefit), and extended data integration.

Such examples are DESMO-J (Java), OpenSim (C++), JaamSim (Java), SimPy (Python), and JSIM (Java). SimPy, as an example, is an open-source object-oriented library and a tool for the development of discrete event simulation models (Peyman et al., 2021). It presents a generic approach that simplifies relevant services such as supply chain, logistics, and healthcare (Mohapatra and Roy, 2023). This solution uses Python, which is a very flexible language, in the sense that it integrates easily with other solutions.

As a more practical example, the authors Joglekar et al., (2022) reconfigured a SimPy based Digital Twin of an assembly line, using interfaces that allowed real-time data visualization, as well as the 3D Digital Twin. Python generators have a high applicability in large-scale calculations, but they are also ideal for coding processes that run in parallel in a DES environment (Dagkakis et al., 2016).

Table 2. is presented as a summary of the key aspects which differ between the commercial and open-source simulation solutions.

Table 2. Summary of commercial vs. open-source simulation solutions.

Commercial	Open-source
Proprietary licenses	More permissive license
Closed-source code	Open-source code
Opaque predefined logic	Transparent predefined logic
Closed intelligent objects	Open intelligent objects
Professional support	Crowdsourcing support
Limited data integration	Extended data integration

It is quite important to highlight that at no time do we discredit the features and potential of commercial simulation solutions. We strongly believe that it is possible for both commercial and open-source simulation solutions to coexist, because both solutions answer to different market needs, and the advent of the Digital Twin era will widen the demand for applying simulation.

By way of example, the iOS and Android mobile operating systems have coexisted for some time now. They offer a different experience, which answers to a different set of requirements / needs.

If a user wants to simulate a system, and his time is limited for modeling, then a commercial simulation solution might be the most adequate. By contrast, if another user wants to simulate a system which must integrate with other data infrastructure, then an open-source simulation solution might be the most appropriate.

Moreover, throughout the life cycle of a system, e.g., a production line, the simulation solution used might change. During its conceptual phase, the commercial simulation solution might be suited to mimic that future system, due to the predefined logic, tested intelligent objects, and professional support. Upon start of production, the same organization might opt for an open-source simulation solution, which offers a more permissive license (possibly even free) and extended data integration.

4. Conclusions

With the advent of the Digital Twin era, the demand for applying simulation will increase significantly. Hence, the barriers to use discrete-event simulation (DES) should be reduced, in order for the discipline to remain relevant and became ubiquitous. A possible solution to these is open-source simulation solutions. This article demonstrates the main differences between commercial and open-source simulation solutions, which were uncovered by multiple sources. Commercial simulation solutions present as advantages its predefined logic and tested intelligent objects, which accelerate modeling time; coupled with professional support. By contrast, open-source simulation solutions offer more permissive licensing (often free), a transparent predefined logic, crowdsource intelligent objects, and extended data integration. Regardless of these differences, we strongly believe that it is possible for both commercial and open-source simulation solutions to coexist, because both solutions answer to different market needs. It is worth highlighting that this study does not cover the modelling experience itself. Rather, on this particular study it was opted to provide a simulation solutions overview. Therefore, for future work we aim at evaluating the modelling experience on open-source simulation solutions, and comparing it to commercial simulation solutions.

Funding and Acknowledgements

This paper was developed with funding from Missão Interface. Project co-financed by RRP - Recovery and Resilience Plan by the European Union (operation code 03/C05-i02/2022.P5).

Appendix A

Table 1. Bibliographic analyses

Abstract / Key Word / Title	Bibliographic database	Period	Results
“Agent-based”	Scopus	2008-2018	≅ 30.000
“Agent Based Modelling and Simulation”	Google Scholar	2013-2024	≅ 18.000
“Commercial Solutions" and "Anylogic"	b-on	2013-2024	70
“Discrete event simulation" and “manufac*”	b-on	2013-2024	186
“Digital twin" and "Discrete event simulation”	Scopus	2013-2024	92
“Digital Twin" and "Review" and “simulation”	Google Scholar	2013-2024	≅26.000
“Industry 4.0 principles”	Scopus	2013-2024	166
“Manufac*" and "Software arena”	Google Scholar	2013-2024	46
“Methodology for Research”	Scopus	2013-2024	150
“MATLAB" and "SIMIO”	Scopus	2013-2024	12
“Open-source " and “simulation” and “discrete event simulation”	b-on	2013-2024	≅ 1.400
"Open-source simulation solutions" and "DESMO-J"	b-on	2013-2024	7
“Open-source simulation solutions " and "JAAMSIM”	b-on	2013-2024	13
“Open-source simulation solutions" and "ManPy”	b-on	2013-2024	17
“Open-source simulation solutions" and "OpenSim”	b-on	2013-2024	983
“Open-source simulation solutions" and "Simpy”	b-on	2013-2024	115
“Open-source simulation solutions" and "Simscape”	b-on	2013-2024	≅ 1.100
“Operation of manufac* systems”	Scopus	2013-2024	53
“Simio”	Scopus	2013-2024	198
Snowball effect	Scopus	-	2
Snowball effect	Google Scholar	-	14
Snowball effect	b-on	-	5
“Simulation tools" and "Manufac*”	Google Scholar	2013-2024	154
“Stochastic simulation" and “manufacture”	Google Scholar	2013-2024	≅ 12.000

References

Abar, S., Theodoropoulos, G. K., Lemarinier, P., and O’Hare, G. M. P. (2017). Agent Based Modelling and Simulation tools: A review of the state-of-art software. *Computer Science Review*, 24, 13–33. https://doi.org/10.1016/j.cosrev.2017.03.001

Ani, U. D., Watson, J. M., Carr, M., Cook, A., and Nurse, J. R. (2022). A review of the use and utility of industrial network-based open source simulators: Functionality, security, and policy viewpoints. *The Journal of Defense Modeling and Simulation: Applications, Methodology, Technology*, *19*(3), 263–286. https://doi.org/10.1177/1548512920953499

Antonova, V. M., Malikova, E. E., Bogomolov, V. E., and Malikov, A. Y. (2023). Some Features of AnyLogic Integration into the University’s Education System. *2023 Systems of Signals Generating and Processing in the Field of on Board Communications*, 1–5. https://doi.org/10.1109/IEEECONF56737.2023.1009 2076

Banik, S., and Zimmer, V. (2022). *System Firmware: An Essential Guide to Open Source and Embedded Solutions*. Apress. https://doi.org/10.1007/978-1-4842-7939-7

Barlas, P., and Heavey, C. (2016). Automation of input data to discrete event simulation for manufacturing: A review. *International Journal of Modeling, Simulation, and Scientific Computing*, *07*(01), 1630001. https://doi.org/10.1142/S1793962316300016

Bélanger, J., and Venne, P. (2010). *The What, Where and Why of Real-Time Simulation*.

Belyaev, A. S., and Desyatirikov, F. A. (2023). Open Source Solutions for Software Development Platform. *2023 XXVI International Conference on Soft Computing and Measurements (SCM)*, 260–263. https://doi.org/10.1109/SCM58628.2023.10159041

Bossel, H. (1994). *Modeling and simulation*. Peters.

Brannick, T., and Coghlan, D. (2005). *Doing Action Research in Your Own Organization*. Sage Publications.

Cañas, H., Mula, J., Díaz-Madroñero, M., and Campuzano-Bolarín, F. (2021). Implementing Industry 4.0 principles. *Computers & Industrial Engineering*, *158*, 107379. https://doi.org/10.1016/j.cie.2021.107379

Cespedes-Cubides, A. S., and Jradi, M. (2024). A review of building digital twins to improve energy efficiency in the building operational stage. *Energy Informatics*, 7(1), 11. https://doi.org/10.1186/s42162-024-00313-7

Chen, W. (2015). *Out-of-order Parallel Discrete Event Simulation for Electronic System-level Design*. Springer International Publishing. https://doi.org/10.1007/978-3-319-08753-5

Choi, B. K. (2013). *Modeling and Simulation of Discrete Event Systems*.

Corlu, C. G., Akcay, A., and Xie, W. (2020). Stochastic simulation under input uncertainty: A Review. *Operations Research Perspectives*, *7*, 100162. https://doi.org/10.1016/j.orp.2020.100162

Dagkakis, G., and Heavey, C. (2016). A review of open source discrete event simulation software for operations research. *Journal of Simulation*, *10*(3), 193–206. https://doi.org/10.1057/jos.2015.9

Dagkakis, G., Heavey, C., Robin, S., and Perrin, J. (2013). ManPy: An Open-Source Layer of DES Manufacturing Objects Implemented in SimPy. *2013 8th EUROSIM Congress on Modelling and Simulation*, 357–363. https://doi.org/10.1109/EUROSIM.2013.70

Dagkakis, G., Papagiannopoulos, I., and Heavey, C. (2016). ManPy: An open-source software tool for building discrete event simulation models of manufacturing systems: ManPy: an OS Software Tool for Building DES Models. *Software: Practice and Experience*, 46(7), 955–981. https://doi.org/10.1002/spe.2347

Dagkakis, G., Rotondo, A., Papagiannopoulos, I., Heavy, C., Geraghty, J., Young, P., and Holland, R. (2014). From COTS Simulation Software to an Open-source Platform: A Use Case in the Medical Device Industry. *Procedia CIRP*, *25*, 283–292. https://doi.org/10.1016/j.procir.2014.10.040

Dehghanimohammadabadi, M., and Keyser, T. K. (2015). *Iterative Simulation-Optimization of a Multi-State Manufacturing System*.

Dehghanimohammadabadi, M., and Keyser, T. K. (2017). Intelligent simulation: Integration of SIMIO and MATLAB to deploy decision support systems to simulation environment. *Simulation Modelling Practice and Theory*, *71*, 45–60. https://doi.org/10.1016/j.simpat.2016.08.007

Garg, R. (2016). Methodology for research I. *Indian Journal of Anaesthesia*, *60*(9), 640. https://doi.org/10.4103/0019-5049.190619

Garwood, T. L., Hughes, B. R., Oates, M. R., O'Connor, D., and Hughes, R. (2018). A review of energy simulation tools for the manufacturing sector. *Renewable and Sustainable Energy Reviews*, *81*, 895–911. https://doi.org/10.1016/j.rser.2017.08.063

Glaessgen, E., and Stargel, D. (2012). The Digital Twin Paradigm for Future NASA and U.S. Air Force Vehicles. *53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference
20th AIAA/ASME/AHS Adaptive Structures Conference
14th AIAA*. 53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference
20th AIAA/ASME/AHS Adaptive Structures Conference
14th AIAA, Honolulu, Hawaii. https://doi.org/10.2514/6.2012-1818

Grieves, M. (2016). *Origins of the Digital Twin Concept*. https://doi.org/10.13140/RG.2.2.26367.61609

Houck, D., and Whitehead, C. (2019). Introduction To Simio. *2019 Winter Simulation Conference (WSC)*, 3802–3811. https://doi.org/10.1109/WSC40007.2019.9004692

Joglekar, A., Bhandari, G., and Sundaresan, R. (2022). An Open Simulator framework for 3D Visualization of Digital Twins. *2022 IEEE International Conference on Internet of Things and Intelligence Systems (IoTaIS)*, 278–284. https://doi.org/10.1109/IoTaIS56727.2022.9975980

Joines, J. A., and Roberts, S. D. (2013). *Simulation Modeling with SIMIO: A Workbook*.

Kelton, W. D., Sadowski, R. P., and Zupick, N. B. (2015). *Simulation with Arena* (6. ed). McGraw-Hill Education.

King, D. H., and Harrison, H. S. (2013). Open-source simulation software “JaamSim” *2013 Winter Simulations Conference (WSC)*, 2163–2171. https://doi.org/10.1109/WSC.2013.6721593

Lang, S., Reggelin, T., Müller, M., and Nahhas, A. (2021). Open-source discrete-event simulation software for applications in production and logistics: An alternative to commercial tools? *Procedia Computer Science*, *180*, 978–987. https://doi.org/10.1016/j.procs.2021.01.349

Macal, C. M., and North, M. J. (2009). Agent-based modeling and simulation. *Proceedings of the 2009 Winter Simulation Conference (WSC)*, 86–98. https://doi.org/10.1109/WSC.2009.5429318

McHaney, R. (2009). *Understanding computer simulation*. Ventus Publishing.

Mohammed Hasan, K. A., Kadhum, A. H., and Morad, A. H. (2019). Evaluation and Improvement of Manufacturing System Using Computer Software Arena. *Al-Khwarizmi Engineering Journal*, *15*(4), 71–78. https://doi.org/10.22153/kej.2019.10.003

Mourtzis, D. (2020). Simulation in the design and operation of manufacturing systems: State of the art and new trends. *International Journal of Production Research*, *58*(7), 1927–1949. https://doi.org/10.1080/00207543.2019.1636321

Mousavi, M., and Estrada, R. (2021). SuperCaustics: Real-time, open-source simulation of transparent objects for deep learning applications. *2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 649–655. https://doi.org/10.1109/ICMLA52953.2021.00108

Ojstersek, R., Javernik, A., and Buchmeister, B. (2023). Optimizing smart manufacturing systems using digital twin. *Advances in Production Engineering & Management*, *18*(4), 475–485. https://doi.org/10.14743/apem2023.4.486

Pegden, D. (2015). *Process Improvement A Concise Guide for Manager*.

Peyman, M., Copado, P., Panadero, J., Juan, A. A., and Dehghanimohammadabadi, M. (2021). A Tutorial on how to Connect Python with Different Simulation Software to Develop Rich Simheuristics. *2021 Winter Simulation Conference (WSC)*, 1–12. https://doi.org/10.1109/WSC52266.2021.9715511

Rozinat, A., Mans, R. S., Song, M., and Van Der Aalst, W. M. P. (2009). Discovering simulation models. *Information Systems*, *34*(3), 305–327. https://doi.org/10.1016/j.is.2008.09.002

Shafto Mike, Conroy, M., Glaessgen, E., Kemp, C., LeMoigne, J., and Wang, L. (2010). *NASA - DRAFT MoDeling, SiMulATion, inFoRMATion Technology & PRoceSSing RoADMAP Technology Area 11*.

Sharma, P. (2015). *Discrete-Event Simulation*. 4(04).

Sitaram, D., Hallymysore, P., Harwalkar, S., Kathare, S., Srinivasan, A., Reddy, A., Kumar, K., and Sekhar, S. (2014). OpenSim: A Simulator of OpenStack Services. *2014 8th Asia Modelling Symposium*, 90–96. https://doi.org/10.1109/AMS.2014.28

Siwek, M., Baranowski, L., Panasiuk, J., and Kaczmarek, W. (2019). *Modeling and simulation of movement of dispersed group of mobile robots using Simscape multibody software*. 020045. https://doi.org/10.1063/1.5092048

Tsaousoglou, G., and Manesis, S. (2014). 3D Simulation of Industrial Large-scale Ceramics Furnace in SIMIO Platform Environment: *Proceedings of the 11th International Conference on Informatics in Control, Automation and Robotics*, 716–723. https://doi.org/10.5220/0005046207160723

Vieira, A. A. C., Dias, L. M. S., Santos, M. Y., Pereira, G. A. B., and Oliveira, J. A. (2019). A ranking of the most known freeware and open source discrete-event simulation tools. *THE EUROPEAN MODELING AND SIMULATION SYMPOSIUM*, 200–2019. https://doi.org/10.46354/i3m.2019.emss.029

Webster, J., and Watson, R. T. (2002). *ANALYZING THE PAST TO PREPARE FOR THE FUTURE: WRITING A LITERATURE REVIEW*.

Wooley, A., Silva, D. F., and Bitencourt, J. (2023). When is a simulation a digital twin? A systematic literature review. *Manufacturing Letters*, *35*, 940–951. https://doi.org/10.1016/j.mfglet.2023.08.014