Manufacturing Letters 28 (2021) 54–58

Contents lists available at ScienceDirect

Manufacturing Letters

journal homepage: www.elsevier.com/locate/mfglet

Letters

Modelling cycle for simulation digital twins

Sean Reed [a], Magnus Löfstrand [a,*], John Andrews [b]

[a] School of Science and Technology, Örebro University, Örebro, Sweden
[b] Department of Mechanical, Materials and Manufacturing Engineering, University Park, University of Nottingham, Nottingham, United Kingdom

ARTICLE INFO

Article history:
Received 13 March 2020
Received in revised form 10 February 2021
Accepted 15 April 2021
Available online 24 April 2021

Keywords:
Digital twin
Industry 4.0
Discrete event simulation
Modelling cycle

ABSTRACT

Digital twins (DT) form part of the Industry 4.0 revolution within manufacturing and related industries. A DT is a digital model (DM) of a real system that features continuous and automated synchronisation and feedback of optimisations between the real and digital domains. A core technology for predictive capabilities from DT is discrete event simulation (DES). The modelling cycle for developing and analysing DES models is significantly different compared to DM. A DT specific DES modelling cycle is introduced that is evolved from that of DM. The availability of specialised software tools for DT tailored to these differences would benefit industry.

1. Introduction

Industry 4.0 [1] refers to the fourth industrial revolution, characterised by the application to industrial production and manufacturing of cyber-physical systems that are composed from deeply intertwined networked physical and software components. This includes digital twins (DT) [2–5] as a core component which are computer models that simulate the behaviour of a real world physical system in the deployed environment, mirroring its current condition using online data from sensors and information systems, to predict and optimise its performance. Similar to a digital model (DM), a DT is a digital representation of a physical object, but features bi-directional and automatic data flow of operational data and feedback between them [6] (see Fig. 1). Facilitated by growth in internet of things (IoT) connectivity, computing power, and advanced analytics, DT have the potential for enhancing a manufacturing system by predicting future events, such as machine faults or production bottlenecks, based on its current status and anticipated behaviour and then implementing the appropriate responses to optimise performance in near real-time [7,8].

Simulation has been defined as the imitation of the operation of a real-world system over time, through the generation and observation of artificial histories, from which inferences are drawn concerning the operation of the real system [9]. Within manufacturing, discrete event simulation (DES) is the most popular simulation technique [10]. A DES model comprises of interacting entities representing the tangible (e.g. manufacturing station) and intangible (e.g. work-in-progress queue) elements of the modelled system [11] that are updated at discrete, but possibly random, event times [12]. The totality of entities and their attributes represents the state of the real system [11]. Simulation has been used extensively for analysing and optimising the design and operation of manufacturing systems [13] and DT has been described as the latest wave in simulation technology [14]. A simulation DT can synchronise with the status (e.g. degradation status) and future operating and maintenance schedules of the assets in the real manufacturing system, simulate the probable future outcomes (e.g. predict a likely machine failure), determine an optimised solution (e.g. scheduling a preventive maintenance action) and update the real system accordingly (e.g. update the computerised maintenance management system (CMMS)) [15,16]. Simulation has been shown as an effective tool for real-time control of manufacturing systems [17–20]. The activities involved in the development, implementation and analysis of a DES model is described by a modelling cycle. For a DES DM, the modelling cycle has been addressed in many publications such as [21–23]. Direct autonomous feedback control from the DT to the physical world has been identified as one of the seven key research issues for advancing Digital Twin-driven smart manufacturing [24]. The emerging ISO 23,247 standard is a generic framework for manufacturing DT [8], comprising four layers: the observable manufacturing system entity, the data collection and device control entity, the digital twin model entity and the user interface entity. A reference DT architecture has also been proposed comprising of a physical layer, a digital layer, a cyber layer, and communication for data exchange amongst them [25]. This

* Corresponding author at: School of Science and Technology, Örebro University, Örebro SE-701 82, Sweden.

E-mail address: magnus.lofstrand@oru.se (M. Löfstrand).

https://doi.org/10.1016/j.mfglet.2021.04.004
2213-8463/

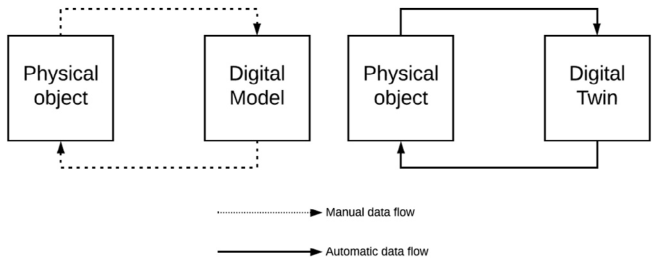

Fig. 1. Difference between a digital model (DM) and digital twin (DT) [6].

paper proposes a DES DT modelling cycle that is intended to fit into these generic frameworks and architectures, comparing it to a DES DM modelling cycle to highlight the major differences.

2. Simulation modelling cycles for DM and DT

A typical modelling cycle for developing and analysing a DES DM is shown in Fig. 2, exemplified by Figure 1.7 in Page and Kreutzer [23] and Figure 1.1 in Banks [22]. The main activities during the model development phase are: define the problem and modelling goals; collect and analyse data from the real system; develop and validate the conceptual model representing the real system; implement and validate the conceptual model to create the DM. Once implemented, the analysis phase begins where the main activities are: perform the simulations to obtain results; analyse the results to obtain solutions to the modelling goals; and implement the solutions in the real system. Each of these activities generally requires input and oversight from the modeller to perform and are therefore defined as manual activities. The analysis phase for a DM is often intended to be performed once, to obtain the solution to the analysis goals before the model is discarded.

The authors researched how the corresponding model development cycle would be for DES DT, resulting in the model development cycle shown in Fig. 3. In the development phase, which is performed offline, system data sources and analytical queries are collated instead of collecting actual data on the real system. These include database management systems (DBMS), where static data is stored (e.g. future production plans), and data stream management systems (DSMS) that process continuous data streams (e.g. produced by sensors monitoring the real system). This difference stems from the fact that a DT must continuously represent the real system as it evolves over time, therefore data describing the real system at a particular time (e.g. model design time) is insufficient. The modeller must simultaneously develop and validate the queries that will be used to obtain the data from these sources that will be later used in generating the DT model. These queries may range from simple lookups to complex machine learning techniques and include those for reading, updating and inserting data to enable data flow between the DT as shown in Fig. 1.

Instead of developing a static DES model representing the real system at development time, a generator is developed that can output a DES model that mirrors the real system at generation time using data it obtains from the queries. First a conceptual model generator is developed that describes the assumptions, abstractions, and logical structure for generating the model from the data pertaining to the real system. It includes details such as which data sources and queries will be used to obtain data for the individual entities in the model, how that data will be transformed into the formation of those model entities and how the logical interrelations between the entities will be composed. The model generator may encompass complex domain specific logic to transform dynamic inputs into models that must be developed on a case-by-case basis. Following on from this, the model generator is

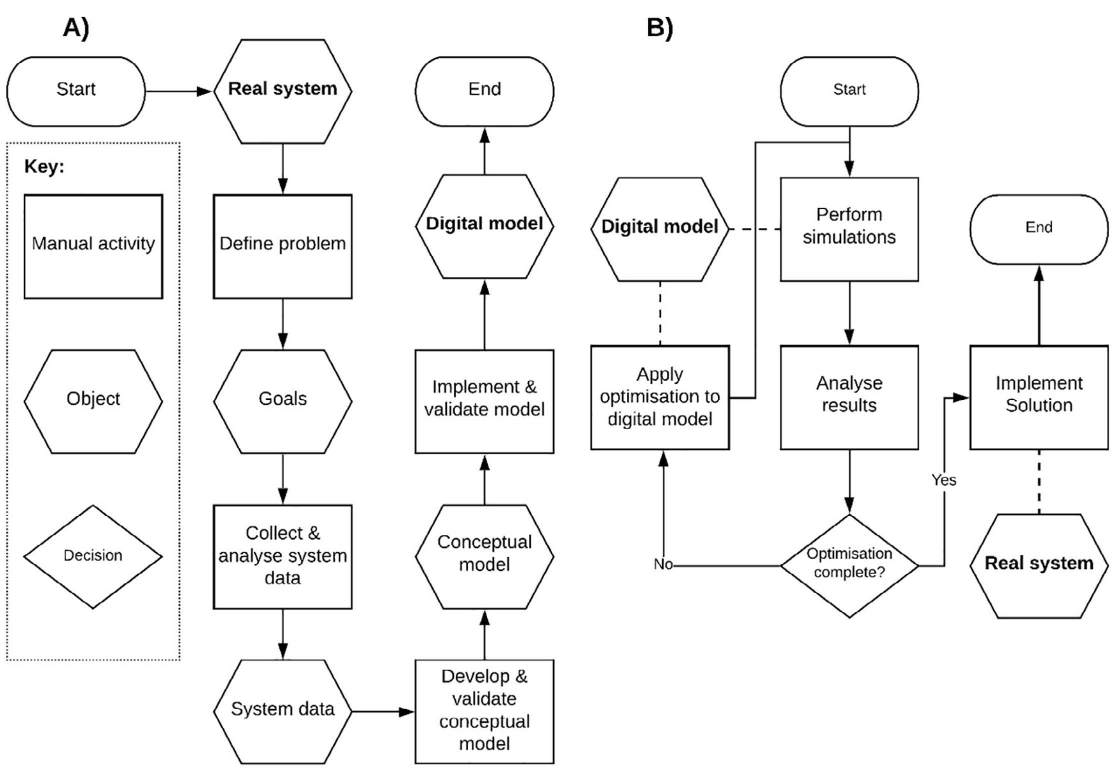

Fig. 2. A) A typical cycle for developing a DES digital model. B) Typical analysis cycle for a DES digital model.

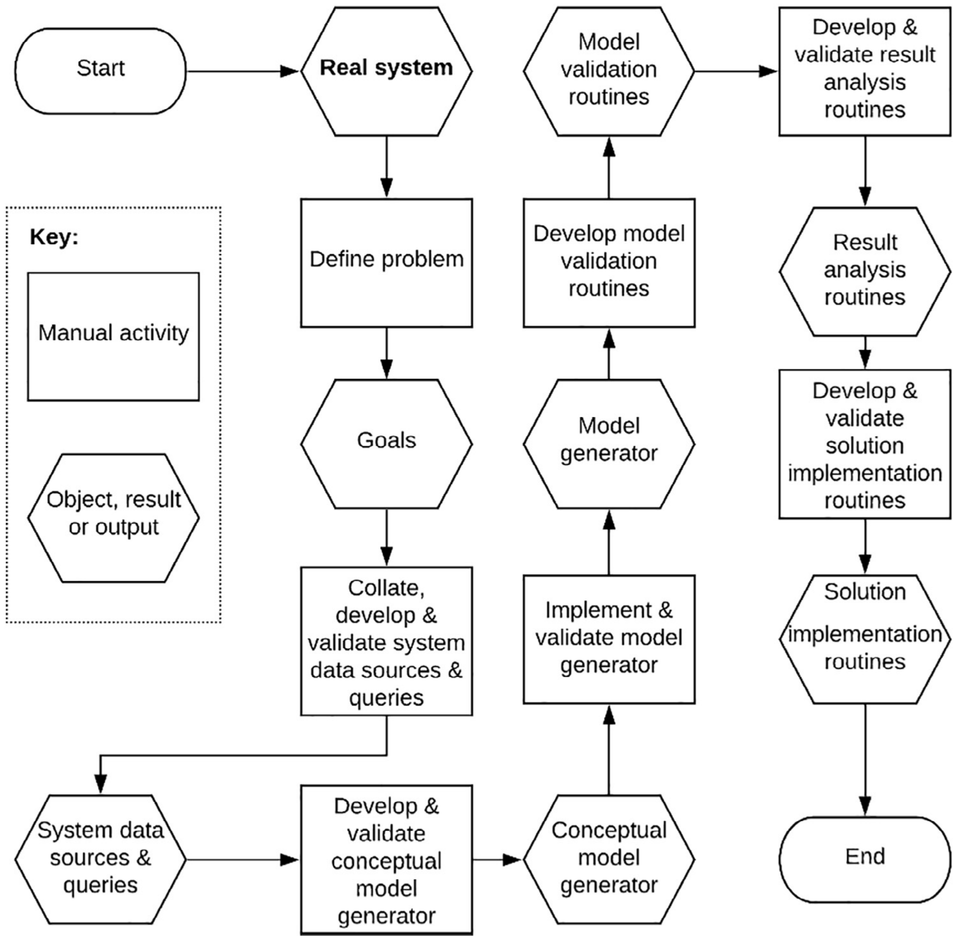

Fig. 3. Proposed cycle for developing a simulation digital twin.

implemented and validated, which includes testing for bugs and correspondence with the conceptual model. The next activities are to develop and validate routines (i.e. algorithms) for validating the DT whenever it is generated, analysing the simulation results to determine optimisations (constituting the 'intelligence' of the DT) and implementing them in the real system. These activities can then be performed each time the model is generated during the analysis cycle, in contrast to the DM modelling cycle where they are performed once manually. The validation routines must ensure that errors in the implementation or data, e.g. from faulty sensors, do not result in the generation of an inappropriate model.

The analysis cycle for a DT, shown in Fig. 4, is performed online continuously and autonomously, using the objects resulting from the development cycle. At the start of each simulation cycle, the digital twin is synchronised to mirror the state of the real system. Usually, only the model entity attributes need updating. Occasionally, when more significant changes occur in the real system, the model entities also need rebuilding by the generator – a more computationally expensive operation. The model is then verified, simulated, and optimised before any optimisations are implemented in the real system and the cycle repeats. The exact time for each cycle will depend on design choices for number of simulation trials (replications), optimisation criterion and implementation, set to balance the need for computational efficiency with avoidance of the model state becoming stale. The distributed execution of this continuous analysis on a cluster of computing nodes is beneficial to obtain faster analysis, increased fault tolerance and improved scalability. This has potential for analysis cycles that update the DT of a manufacturing system every few seconds to reflect its current state in near real-time.

3. Summary and conclusions

A proposed model development and analysis cycle for a DES DT was presented and compared to the equivalent cycle for a DES DM. A major difference is that it uses data queries, a model generator, validation routines, results analysis routines and optimisation implementation routines instead of a static model. This allows analysis to be performed in a repeated cycle that provides automated optimisation of the real system based on its current state in near real-time. To ease the adoption of DT within the manufacturing industry and maximise the productivity benefits from the Industry 4.0 revolution, specialised software tools designed for DES DT modelling that cater to these differences are needed, along with example use cases of the modelling cycle in manufacturing industry applications. The mapping of the DES based predictive DT modelling cycle into the layers and elements of the emerging ISO 23,247 standard [8] and reference architectures [25] for generic DT is another area for further work. The authors hope to contribute to these areas through future research.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

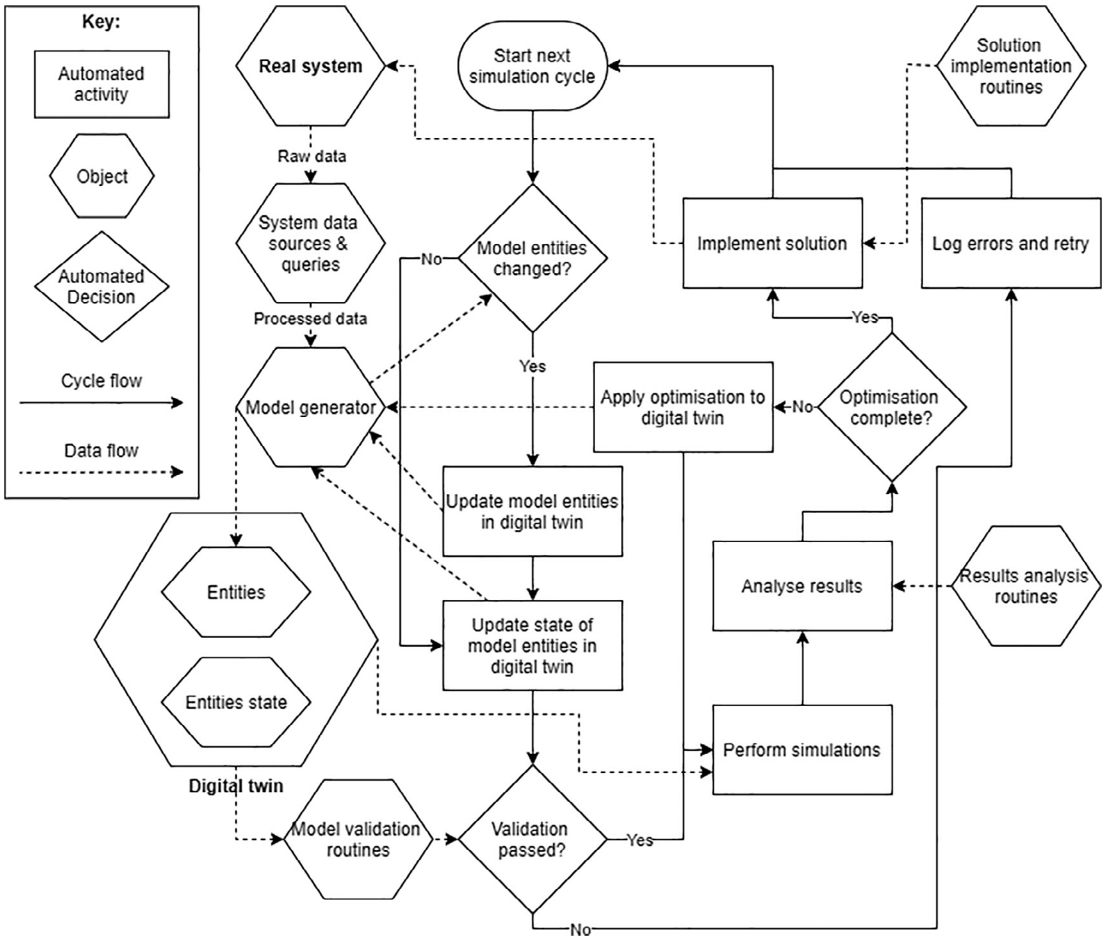

Fig. 4. Proposed continuous analysis cycle for a DES digital twin.

Acknowledgements

This work is with regards to Prof. M. Löfstrand and Dr. S. Reed carried out within the project A digital twin to support sustainable and available production as a service, funded by Produktion2030, the Strategic innovation programme for sustainable production in Sweden, and is supported by the project Production Centred Maintenance for real-time predictive maintenance decision support to maximise production efficiency, funded by the Swedish Knowledge Foundation. We gratefully acknowledge the support and funding.

References

[1] Roblek V, Meško M, Krapež A. A Complex View of Industry 4.0. SAGE Open 2016;6(2). https://doi.org/10.1177/2158244016653987.
[2] Negri E, Fumagalli L, Macchi M. A review of the roles of digital twin in CPS-based production systems. Procedia Manuf 2017;11:939–48. https://doi.org/10.1016/j.promfg.2017.07.198.
[3] Haag S, Anderl R. Digital twin – Proof of concept. Manuf. Lett. 2018;15(Part B):64–6. https://doi.org/10.1016/j.mfglet.2018.02.006.
[4] Tao F, Cheng J, Qi Q, Zhang M, Zhang H, Sui F. Digital twin-driven product design, manufacturing and service with big data. Int J Adv Manuf Technol 2018;94(9–12):3563–76. https://doi.org/10.1007/s00170-017-0233-1.
[5] Grieves M. "Digital Twin: Manufacturing Excellence Through Virtual Factory Replication," 2014.
[6] Kritzinger W, Karner M, Traar G, Henjes J, Sihn W. Digital Twin in manufacturing: A categorical literature review and classification. IFAC-PapersOnLine 2018;51(11):1016–22. https://doi.org/10.1016/j.ifacol.2018.08.474.
[7] Zhuang C, Liu J, Xiong H. Digital twin-based smart production management and control framework for the complex product assembly shop-floor. Int J Adv Manuf Technol 2018;96(1–4):1149–63. https://doi.org/10.1007/s00170-018-1617-6.
[8] Shao G, Helu M. Framework for a digital twin in manufacturing: Scope and requirements. Manuf Lett 2020;24:105–7. https://doi.org/10.1016/j.mfglet.2020.04.004.
[9] Banks J, Carson II JS, Nelson BL, Nicol DM. Discrete Event System Simulation. 5th ed. Pearson Education; 2009.
[10] Jahangirian M, Eldabi T, Naseer A, Stergioulas LK, Young T. Simulation in manufacturing and business: A review. Eur J Oper Res 2010;203(1):1–13. https://doi.org/10.1016/j.ejor.2009.06.004.
[11] Sánchez PJ. Fundamentals of simulation modeling. Winter Simulation Conference Proceedings 2007:54–62. https://doi.org/10.1017/CBO9781107415324.004.
[12] Schriber TJ, Brunner DT. "Inside discrete-event simulation software: How it works and why it matters," Proc. 2013 Winter Simul. Conf., pp. 80–94, 2011, doi: 10.1109/WSC.2013.6721439.
[13] Negahban A, Smith JS. Simulation for manufacturing system design and operation: Literature review and analysis. J Manuf Syst 2014;33(2):241–61. https://doi.org/10.1016/j.jmsy.2013.12.007.
[14] Boschert S, Rosen R. Digital Twin - The Simulation Aspect. In: Hehenberger P, Bradley D, editors. Mechatronic Futures: Challenges and Solutions for Mechatronic Systems and Their Designers. Springer; 2016. p. 59–74.
[15] Vatn J. Industry 4. 0 and real-time synchronization of operation and maintenance. In: in *Proceedings of the 28th International European Safety and Reliability Conference*. p. 681–5.
[16] Errandonea I, Beltrán S, Arrizabalaga S. Digital Twin for maintenance: A literature review. Comput Ind 2020;123:103316. https://doi.org/10.1016/j.compind.2020.103316.
[17] Wu SYD, Wysk RA. Multi-pass expert control system - a control/scheduling structure for flexible manufacturing cells. J Manuf Syst 1988;7(2):107–20. https://doi.org/10.1016/0278-6125(88)90018-0.
[18] Smith JS, Wysk RA, Sturrock DT, Ramaswamy SE, Smith GD, Joshi SB. Discrete event simulation for shop floor control. Winter Simul Conf Proc 1994:962–9. https://doi.org/10.1109/wsc.1994.717475.
[19] Iassinovski S, Artiba A, Fagnart C. A generic production rules-based system for on-line simulation, decision making and discrete process control. Int J Prod Econ 2008;112(1):62–76. https://doi.org/10.1016/j.ijpe.2006.08.028.
[20] Nandi A, Rogers P. Using Simulation to Make Order Acceptance/Rejection Decisions. Simulation 2004;80(3):131–42. https://doi.org/10.1177/0037549704045046.

[21] Law AM. How to build valid and credible simulation models. Proc - Winter Simul Conf 2005;2005:24–32. https://doi.org/10.1109/WSC.2005.1574236.
[22] Banks J. Handbook of Simulation: Modelling, Estimation and Control. John Wiley & Sons; 1998.
[23] Page B, Kreutzer W. The Java Simulation Handbook: Simulating Discrete Event Systems with UML and Java. Shaker Verlag GmbH; 2005.
[24] Lu Y, Liu C, Wang KIK, Huang H, Xu X. Digital Twin-driven smart manufacturing: Connotation, reference model, applications and research issues. Robot Comput Integr Manuf 2020;61(April). https://doi.org/10.1016/j.rcim.2019.101837.
[25] Aheleroff S, Xu X, Zhong RY, Lu Y. "Digital Twin as a Service (DTaaS) in Industry 4.0: An Architecture Reference Model," Adv. Eng. Informatics, vol. 47, 2021, doi: 10.1016/j.aei.2020.101225.