A community-sourced view on engineering digital twins: A Report from the EDT.Community

Loek Cleophas, Thomas Godfrey, Djamel Eddine Khelladi, Daniel Lehner, Benoit Combemale, Mark van den Brand, Michael Vierhauser, Manuel Wimmer, Steffen Zschaler

A Community-Sourced View on Engineering Digital Twins: A Report from the EDT.Community

Loek Cleophas
Software Engineering Technology
Eindhoven University of Technology
Eindhoven, the Netherlands
l.g.w.a.cleophas@tue.nl

Thomas Godfrey
Department of Informatics, King's College London
London, UK
thomas.1.godfrey@kcl.ac.uk

Djamel Eddine Khelladi
CNRS, IRISA, INRIA, Univ. Rennes
Rennes, France
djamel-eddine.khelladi@irisa.fr

Daniel Lehner
Institute of Business Informatics - Software Engineering
Johannes Kepler University
Linz, Austria
daniel.lehner@jku.at

Benoit Combemale
IRISA, INRIA, Univ. Rennes
Rennes, France
benoit.combemale@irisa.fr

Mark van den Brand
Software Engineering Technology
Eindhoven University of Technology
Eindhoven, the Netherlands
m.g.j.v.d.brand@tue.nl

Michael Vierhauser
LIT Secure and Correct Systems Lab
Johannes Kepler University
Linz, Austria
michael.vierhauser@jku.at

Manuel Wimmer
Institute of Business Informatics - Software Engineering, CDL-MINTu
Johannes Kepler University
Linz, Austria
manuel.wimmer@jku.at

Steffen Zschaler
Department of Informatics, King's College London
London, UK
szschaler@acm.org

ABSTRACT

Digital Twins are an important concept, enabling what-if scenario exploration, predictive maintenance, and other approaches. They help in saving time and physical resources when developing and evolving systems, whether natural or engineered. However, constructing and maintaining digital twins is a challenging engineering task – and, to date, there is a lack of understanding of the engineering techniques and methodologies required. To address these challenges, we created EDT.Community, a programme of seminars on the engineering of digital twins hosting digital twins experts from academia and industry. In this paper, we report on the main topics of discussion from the first year of the programme. We contribute by providing (1) a common understanding of open challenges in research and practice of the engineering of digital twins, and (2) an entry point to researchers who aim to close gaps in the current state of the art.

CCS CONCEPTS

• **Computer systems organization → Embedded and cyber-physical systems**.

KEYWORDS

digital twin, digital engineering, systems engineering

MODELS '22 Companion, October 23–28, 2022, Montreal, QC, Canada

ACM ISBN 978-1-4503-9467-3/22/10.
https://doi.org/10.1145/3550356.3561549

ACM Reference Format:
Loek Cleophas, Thomas Godfrey, Djamel Eddine Khelladi, Daniel Lehner, Benoit Combemale, Mark van den Brand, Michael Vierhauser, Manuel Wimmer, and Steffen Zschaler. 2022. A Community-Sourced View on Engineering Digital Twins: A Report from the EDT.Community . In *ACM/IEEE 25th International Conference on Model Driven Engineering Languages and Systems (MODELS '22 Companion), October 23–28, 2022, Montreal, QC, Canada.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3550356.3561549

1 INTRODUCTION

Digital Twins (DTs) are an important concept, originating in the area of operations management and Industry 4.0 [8], and increasingly seeing applications in many areas of engineering, science, and even business management and organisational decision-making. A DT is a computational representation of a (potentially still to be created) real-world entity (an object, system, process, organisation, or organism). It is continuously kept up-to-date with the state of the entity, and has the ability to influence the state and behaviour of the entity based on changes to the digital representation [10]. DTs often provide the opportunity for simulation-based analysis of what-if scenarios, predictive maintenance, etc.

While DTs thus have many potential benefits, developing and maintaining them is a significant and complex engineering task. To date, there is a lack of understanding of the engineering processes and techniques needed to successfully and systematically develop, validate, maintain, and use DTs.

To help to address these challenges, we have founded the community and programme, EDT.Community[1], for seminars on the engineering of DTs. We structured the seminar programme in three seminar series per year, each one ending with a separate panel

[1]https://edt.community

discussion with the speakers of that series.[2] While the seminars provide in-depth details on a broad scope of applications for DTs (cf. the EDT.Community YouTube channel[3]), the panel discussions bring the speakers together to confront their points of views, and identify commonalities and differences.

In this paper, we report a summary of the main topics and challenges that have been discussed during the first year of the EDT.Community seminar programme. Therefore, we are focusing on presenting the challenges that the community has agreed on during the panel discussions. For particular solutions, we refer to the recordings of the talks available on our youtube channel referenced above.

Thus, this paper aims to contribute to a common understanding of open challenges in research and practice of the engineering of DTs. They also provide an entry point to researchers who aim to close gaps in the current state of the art.

2 EDT.COMMUNITY

EDT.Community is a community and a programme of online seminars on the topic of digital twins. It is organised by the authors of this paper, with Cleophas, Godfrey, Khelladi, Lehner, and Vierhauser serving as organising committee and Combemale, van den Brand, Wimmer, and Zschaler as steering committee. The main intention and purpose of EDT.Community is to provide a central space for researchers and practitioners to discuss the foundations and engineering of digital twins and to further share knowledge and ambitions for the field from different points of view. With the wide scope of the digital twin research space, and the many interpretations of what constitutes a "digital twin", an initial objective of our seminars was to establish a common definition and a shared understanding and vocabulary for the engineering of digital twins across different domains.

The programme has been run as a sequence of series, with each series consisting of four to five talks, run bi-weekly, and followed by a joint panel discussion. The panels are designed to allow speakers from the talks of a series to debate and discuss topics on the theory, engineering, and application of digital twins.

We provide a brief overview of all speakers and topics below. Some of the talks focused on foundational topics, such as how to define digital twins; how to combine formal methods and digital twins; and a reference architecture for digital twin construction. Others focused on applied topics across a vast application range, ranging from high-tech systems (such as aircraft design and production printing systems) to living environments and social settings (including tomato agriculture, human-focused health, try-on technology in fashion, social networks, and parcel delivery). Further talks addressed models and digital twins from a DevOps perspective, and conceptual modelling from a digitalization perspective. The range of talks included presenters giving kaleidoscopic overviews of the research around digital twins with organisations from industry and academia. We list all the talks below:

- Julien Schmaltz, ICT Group, *Digital Twins for Digital Engineering.*
- Sarah Wise, UCL, *An Agent-based Approach to Last-Mile Parcel Delivery in Urban Contexts.*
- Guy de Spiegeleer, Safran, *The benefits of an extended digital twin to support the design of complex systems.*
- Willem Jan Knibbe, Wageningen University & Research, *Digital Twins in the Living Environment.*
- Eugen Schindler, Canon Production Printing, *Holistically Composable and Sustainable Digital Twins and Threads.*
- Andreas Wortmann, University of Stuttgart, *Ceci n'est pas un jumeau numérique.*
- Peter Gorm Larssen, Aarhus University, *Possibilities with Digital Models, Digital Shadows and Digital Twins for Cyber-Physical Systems.*
- Patrick Debois, Snyk, *Will models make a comeback in the DevOps SDLC?*
- Bayu Jayawardhana, University of Groningen, *Integration of data-driven and model-based engineering in future industrial technology.*
- David McKee, Slingshot Simulations and Digital Twin Consortium, *A Reference Architecture for Building Digital Twins.*
- Einar Broch Johnsen, University of Oslo, *Digital Twins as Evolving Model-Centric Systems.*
- Jan Recker, University of Hamburg, *From Representation to Mediation: A New Agenda for Conceptual Modeling Research in a Digital World.*
- Mark Harman, Meta, *cyber-cyber digital twins: Facebook's Web-enabled simulation.*
- Marcelino Rodriguez Cancio, Couture Technologies, *Le Twin, C'est Chic: Digital Twins for Fashion Application.*

The content presented in these talks can be regarded as a representative of digital twin research and practice across various domains and regions. They outline current state of practices, ongoing efforts, and open challenges regarding DTs in our community. Thus, in this paper, we discuss common themes across these talks and panels, to provide a community-source view on the engineering of Digital Twins.

The rest of this paper is structured based on topics that have been discussed during the first year of the EDT.Community seminar and comprise our community-sourced view. We first address the discussions on the definitions of digital twins, their foundations and engineering principles, before we report on the discussions around the adoption of digital twins in practice.

3 DEFINITION OF A DIGITAL TWIN

Using a common definition of what a DT is seems central to any discussion or common research effort towards engineering DTs. Andreas Wortmann, however, outlined in his talk that there is still no such common definition that is widely accepted in research. He backed this statement with insights from a recent mapping study of 356 papers [6]. He also proposed an alternative definition that covers all of the DT aspects found in this mapping study, defining a DT of a system as consisting of *"a set of models of the system, a set of digital shadows, and provides a set of services to use the data and models purposefully with respect to the original system"*, as described in [4]. Although this proposed definition covers aspects found in most surveyed papers, he mentioned in the following panel discussion that this definition still needs to be adopted by academia and industry in order to have actual value.

In contrast to this theoretical definition, David McKee in his talk gave insights into the current industrial understanding of DTs, presenting the outcome of a standardization effort by the OMG Digital Twin Consortium (DTC). In this effort, more than 300 companies agreed on the following definition of a DT: *"a virtual representation*

[2] The third panel was postponed and will be merged with the envisioned panel at the end of a fourth seminar series.

[3] https://www.youtube.com/channel/UC-Qd0zGA4LFdSMZTPHJ69kw

of real-world entities and processes, synchronized at a specified frequency and fidelity". This very well correlates with the well-cited definition by Kritzinger et al. in [10]. In addition to the presented definition, David McKee also mentions the purpose of a DT, and the models used to represent the DT, as essential aspects. This definition can be seen as representative of current industrial adoption of DTs, as the 300 members of the DTC agreed that it represents their current state of practice. There are also many overlaps with the definition proposed by Andreas Wortmann.

However, in both the first and second series' panel discussions, the missing agreement on a single definition of DTs was still raised as a central issue on the research roadmap of DTs.

In addition to this variation in used DT definitions, several aspects that are still missing in most of these definitions have also been discussed in panel discussions. Foremost, boundaries of DTs—i.e. which parts of a cyber-physical system can be named as DT, and which one can not—are still undefined. These unclear boundaries lead to uncertainty on whether particular elements can be regarded as part of the DT (and thus should be tackled by DT researchers) or not. One particular open point regarding boundaries is the pluriformity of DTs. This means that there are different entities that can be represented by a DT (e.g., a device, system, or process—whether engineered or natural), leading to different requirements on the DT. A common definition requires to cover all of these requirements, but the breadth of entities may also indicate that different definitions may be needed to cover different domains.

It seems that there are still several roadblocks on the way to fully understanding and defining DTs. However, such an effort can be regarded as a prerequisite to further work on DTs, including the aspects presented in the following sections of this paper. Based on such a common definition, the role of DTs, particularly in digitalization, can be discussed. Without such a definition, it is difficult to focus research efforts on DTs. Even more importantly though, without the required industry adoption of such a definition, DT research outcomes will hardly be adopted by industry in the future. On the way to a widely accepted DT definition, main challenges will include the integration of the different viewpoints of the various domains talking about DTs, and the separation of core functionalities of a DT (such as synchronization with a physical asset) with techniques that might make use of a DT, such as simulation. The latter aspect was also mentioned by Mark Harman in his talk on the cyber-cyber DTs.

4 FOUNDATIONS AND ENGINEERING PRINCIPLES FOR DIGITAL TWINS

This section elaborates on the open challenges concerning foundations and engineering principles for digital twins which have been collected from the discussions at the EDT.Community seminars and panels. During our seminars and panels, a recurring issue was a lack of life-cycles for DTs, although they should support the life-cycles of systems. In particular, the issue has been raised that the life-cycle of DT development that can be different from standard software development. For example, testing and validation of DTs would be different from "regular" software. Whereas in the previous section we highlighted the lack of standardized and widely adopted definitions of DTs, the same holds true for their life-cycles. Inspiration can be taken from the life-cycle models in software engineering [11] (e.g. from DevOps, as pointed out by Patrick Debois in his talk), but an in-depth analysis is necessary to investigate the commonalities and particularities of a DT life-cycle w.r.t. the different domains. A general suggestion from our panel was for the lifespan of the DT to exist before, during, and after the life-cycle of the physical asset. A DT can be used for experimentation before construction of the physical asset, and can be used after the life-cycle of the asset to promote circular economy by contributing to the construction of subsequent products. One also needs to consider how to maintain and handle the evolution of DTs and their co-evolution with the physical product. This could also help in easing adoption of DTs at a larger scale in industry.

Moreover, one of the points of discussion was centred around the reuse and composition of several digital twins for a single physical system. While this challenge has been studied in an MDE context for model composition [1], it still remains fairly unexplored for DTs. Behind the challenge of composability of DTs, the panellists noted a first preliminary task of how to leverage existing models and their structure, behaviour, and constraints for extraction to a DT.

Another open question asked during the panel focused on how a physical system can remain safe as it gets reconfigured by the DT? Indeed, companies who use DT solutions still struggle with this aspect of safety and security and it remains an open research challenge [2, 7].

Finally, our panellists discussed the state of practice of *"hacking/-ad-hoc"* development of DTs since there currently exists little guidance on abstracting or making generic aspects of a DT, such as cosmotech[4].

We see this as a potential consequence of the lack of standardised definitions of DTs as discussed above. Addressing this challenge could foster greater reuse among DTs and leverage of existing technologies/techniques (such as MDE for a top-down generation approach [5, 9, 12]). Beyond this particular challenge, the MDE and SE community can contribute and transfer its strong experience and knowledge to DTs.

Through our range of talks and panels, we found that a consensus on a definition of digital twin foundations has still not been met, and this has potential impact on the ability to share and re-use DTs as well as the composition of multiple DTs for single physical products.

Safety and security remains a concern, and DT developers need to ensure appropriate systems are in place to prevent the digital twin compromising security or safety properties of the physical asset.

5 ADOPTION OF DIGITAL TWINS

Beyond establishing a formal definition of what constitutes a DT, and then tackling its technical implementation, it is equally important to consider the barriers to adoption of the technology in practise. DTs are an exciting solution, but for the technology to be adopted in different domains, stakeholders need to first be sold on the problem(s) a DT can solve.

[4]https://cosmotech.com/

During our seminars and panels, the following unique selling propositions (USPs) of DTs came up:

- **Faster time-to-market**: Physical products can be developed more quickly when based on a digital counterpart.
- **Higher likelihood of first-time-right implementations**: For production systems, digital representations of a physical product can help developers identify potential flaws before first-time production.
- **Enabling DT-based selection of 'best-choice' organism production**: For example, in greenhouses, DTs using sensors and models can be used to improve the choice between plant types (genotypes) to grow in future production cycles.
- **DTs for process improvement**: Similarly, DTs can be used to optimize processes used in organisations or systems.
- **Enabling predictive maintenance of systems**: DTs allow real-time detection and diagnosis of faults in a physical system.
- **Lower demand on limited resources**: Testing with DTs uses less energy, minerals, biological samples, etc. in comparison to testing with physical products.

However, for many industries, these USPs have not been realised due to unfamiliarity with DTs as a concept. During our panel, we asked the question what domains have not yet received the adoption of DTs, even though they would make good candidates for the technology. Transportation planning was raised as a prime example. In many modern cities, transportation is a sophisticated system of large amounts of data collection, modelling and simulation. While we have seen examples of DTs for urban delivery vehicles (see the talk by Sarah Wise), to the best of our knowledge, there has not yet been an integration of public transport simulations with runtime using a DT. In domains like this, where the technical challenges of data collection and modelling have already been addressed, we have to ask why DTs have not seen greater success.

From our discussions, we first highlight that cost can be a significant limiting factor. It can often be difficult for companies to establish the costs of investing in a DT, and to quantify the benefits of introducing the technology. DTs can be complex and time-consuming to develop, requiring specific expertise which may necessitate new employment positions in a company. Beyond system design time, it will be necessary for staff to maintain a DT during its lifetime. The money and time investment needed to facilitate this can be difficult to manage, especially for new or smaller companies. As discussed in the talk by Patrick Debois, and covered in our second panel, DTs can lead to a significant cultural change in companies. DTs can promote a continuous engineering approach to product development, which can allow for rapid response to product change requests, but as a negative consequence, can result in the product never being 'finished' and always striving for improvements.

Companies must also consider their existing solutions, and how these legacy systems will be integrated with, or replaced by, a DT. A company with existing modelling infrastructure will need to establish the development cost/benefit of migrating their existing solution to be compatible with a DT approach, or developing a new system from scratch. To help address this, we pose the open research question: 'What is an effective incremental path to DTs?'. Rather than companies implementing large interventions to introduce DTs in one shot, can we follow an iterative, constructive process in which companies increase their technological readiness and "build-up" to a full-scale DT, without compromising the DT's effectiveness and without incurring large up-front costs.

It should be noted that, as with many new technologies, DTs present an ideal solution which may or may not ever actually be fully realised. DTs serve as a dot on the horizon to move towards. How and how far to go in developing a DT should be fit into the enterprise model connecting business and IT models, and in the end, be driven by both business needs as well as technological advances.

6 CONCLUSION AND PERSPECTIVES

In this paper, we present a community-sourced view based on the first year of our EDT.Community seminar series, comprising fourteen talks and two panel discussions. First of all, the lack of a common, widely accepted definition of DTs is still an open issue. Besides developing such a common definition, there are also several other lines of foundational work to be done regarding DTs. The current state of practice of DTs is thereby regarded as rather "ad-hoc development" instead of actual engineering. Thus, more cross-domain methods for engineering DTs are required in the future. More specifically, in this paper, we discuss the standardization of the DT life-cycle, the reuse and composition of several DTs for one physical system, and methods for DT maintenance and evolution of DTs, as well as safety and security of DTs.

Along with this foundational work, another central aspect of our community-sourced view is the adoption of DTs in practice. Unfamiliarity with the DT concept, as well as the costs for creating DTs in terms of money, effort, and required expertise seem to currently limit their rapid adoption. To improve acceptance of DTs in practice, one recommendation is communicating the USPs of DTs, which is also discussed in this paper.

Of course, these adoption challenges are not new or limited to DTs. Thus, in the future, existing models for adopting technology (e.g., Assmann & Engels [3] for service engineering) could be leveraged in the context of DTs. In order to aid this adoption from a technical perspective, further research as well as industry-ready solutions covering the foundations of DTs are required. This foundational work however should be based on a common understanding and definition of *what* a DT is. Therefore, as a first step, different viewpoints from different domains must be integrated into one, or potentially several, definitions that can be agreed on by both academia and industry.

ACKNOWLEDGMENTS

The authors want to use this opportunity to appreciate the contributions of all presenters of the EDT.Community to enable the discussion in our community, and deliver the results that are presented in this paper. This work is partially funded by the Austrian Science Fund (P 30525-N31) and by the Austrian Federal Ministry for Digital and Economic Affairs and the National Foundation for Research, Technology and Development (CDG). It is also partially funded by the *RENNES METROPOLE* under grant *AIS no. 19C0330* and by *ANR* agency under grant *ANR JCJC MC-EVO^2 204687*. Thomas Godfrey is supported by the KCL funded Centre for Doctoral Training (CDT) in Data-Driven Health.

REFERENCES

[1] Anas Abouzahra, Ayoub Sabraoui, and Karim Afdel. 2020. Model composition in Model Driven Engineering: A systematic literature review. *Information and Software Technology* 125 (2020), 106316. https://doi.org/10.1016/j.infsof.2020.106316

[2] Kaznah Alshammari, Thomas Beach, and Yacine Rezgui. 2021. Cybersecurity for digital twins in the built environment: current research and future directions. *Journal of Information Technology in Construction* 26 (2021), 159–173.

[3] Martin Assmann and Gregor Engels. 2008. Transition to service-oriented enterprise architecture. In *European Conference on Software Architecture*. Springer, 346–349. https://doi.org/10.1007/978-3-540-88030-1_34

[4] Pascal Bibow, Manuela Dalibor, Christian Hopmann, Ben Mainz, Bernhard Rumpe, David Schmalzing, Mauritius Schmitz, and Andreas Wortmann. 2020. Model-driven development of a digital twin for injection molding. In *International Conference on Advanced Information Systems Engineering*. Springer, 85–100. https://doi.org/10.1007/978-3-030-49435-3_6

[5] Marco Brambilla, Jordi Cabot, and Manuel Wimmer. 2017. *Model-Driven Software Engineering in Practice, Second Edition*. Morgan & Claypool Publishers. https://doi.org/10.2200/S00751ED2V01Y201701SWE004

[6] Manuela Dalibor, Nico Jansen, Bernhard Rumpe, David Schmalzing, Louis Wachtmeister, Manuel Wimmer, and Andreas Wortmann. 2022. A Cross-Domain Systematic Mapping Study on Software Engineering for Digital Twins. *Journal of Systems and Software* 193 (2022), 111361. https://doi.org/10.1016/j.jss.2022.111361

[7] Matthias Eckhart and Andreas Ekelhart. 2019. Digital twins for cyber-physical systems security: State of the art and outlook. In *Security and quality in cyber-physical systems engineering*. Springer, Cham, 383–412. https://doi.org/10.1007/978-3-030-25312-7_14

[8] Michael Grieves. 2014. Digital twin: manufacturing excellence through virtual factory replication. *White paper* 1, 2014 (2014), 1–7.

[9] Stuart Kent. 2002. Model Driven Engineering. In *Integrated Formal Methods*, Michael Butler, Luigia Petre, and Kaisa Sere (Eds.). 286–298.

[10] Werner Kritzinger, Matthias Karner, Georg Traar, Jan Henjes, and Wilfried Sihn. 2018. Digital Twin in manufacturing: A categorical literature review and classification. *IFAC-PapersOnLine* 51, 11 (2018), 1016–1022.

[11] Nayan B Ruparelia. 2010. Software development lifecycle models. *ACM SIGSOFT Software Engineering Notes* 35, 3 (2010), 8–13.

[12] Douglas C Schmidt. 2006. Model-driven engineering. *IEEE Computer* 39, 2 (2006), 25.